



**UNIWERSYTET RZESZOWSKI
WYDZIAŁ MATEMATYCZNO-PRZYRODNICZY**

Kierunek: Informatyka

Studia stacjonarne

**Jakub Kuśnierz
92075**

**Implementacja aplikacji Chatbot
reagującej na emocje rozmówcy**

Praca inżynierska

wykonana pod kierunkiem

dr. inż. Marcina Ochaba

Rzeszów 2019



UNIVERSITY OF RZESZÓW
FACULTY OF MATHEMATICS AND NATURAL SCIENCES

The field of study: Informatics

Types of studies full-time

Jakub Kuśnierz
92075

**A chat bot application that reacts to
the emotions of the interlocutor -
design and implementation**

Type of the thesis: Engineer

**The thesis written under the supervision of
dr inż. Marcin Ochab**

Rzeszów 2019

Promotor

.....

(IMIĘ I NAZWISKO PROMOTORA)

Przyjmuję pracę pt.:

.....

.....

.....

.....

(PODPIS PROMOTORA)

OŚWIADCZENIE

.....
Imię (imiona) i nazwisko studenta/doktoranta

.....
Nazwa jednostki

.....
Nazwa kierunku/ nazwa dyscypliny

.....
Numer albumu:

Oświadczam, że moja praca dyplomowa pt.:

„Implementacja aplikacji Chatbot reagującej na emocje rozmówcy”

1. została przygotowana przeze mnie samodzielnie*,
2. nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2018 r., poz. 1191 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
3. nie zawiera danych i informacji, które uzyskałem w sposób niedozwolony,
4. nie była podstawą nadania dyplomu uczelni wyższej lub tytułu zawodowego ani mnie ani innej osobie.

Ponadto oświadczam, że treść pracy przedstawionej przeze mnie do obrony zawarta na przekazywanym nośniku elektronicznym jest identyczna z wersją drukowaną.

.....
(miejscowość, data)

.....
(czytelny podpis autora pracy)

** uwzględniając merytoryczny wkład promotora pracy*

..... roku

.....

(IMIĘ I NAZWISKO STUDENTA)

.....

(NR ALBUMU)

.....

(WYDZIAŁ)

.....

(KIERUNEK STUDIÓW)

.....

(RODZAJ I FORMA STUDIÓW)

OŚWIADCZENIE

Oświadczam, że część badawcza niniejszej pracy została wykonana w pracowni Laboratorium w ramach Projektu „Uniwersyteckie Centrum Innowacji i Transferu Wiedzy Techniczno-Przyrodniczej”, Nr UDA-RPPK.01.03.00-18-001/10-00, współfinansowanego przez Unię Europejską z Europejskiego Funduszu Rozwoju Regionalnego w ramach Regionalnego Programu Operacyjnego Województwa Podkarpackiego na lata 2007-2013.

.....

(PODPIS PROMOTORA)

.....

(PODPIS STUDENTA)

Spis treści

1. Wstęp.....	13
2. Chatboty.....	15
2.1. Historia chatbotów	15
2.2. Wykorzystanie chatbotów.....	15
2.3. Platforma Wit.ai	16
2.4. Pozostałe platformy tworzenia chatbotów	17
3. Analiza emocji w tekście.....	19
3.1. Przetwarzanie języka naturalnego	19
3.2. Klasyfikacja natężenia emocji	19
3.3. Stosowane podejścia analizy wydźwięku	20
3.4. Zastosowany algorytm oceny.....	21
4. Technologie, biblioteki, narzędzia	22
4.1. Java.....	22
4.1.1. Java 10	22
4.1.2. Spring Boot.....	22
4.1.3. Maven.....	22
4.1.4. Thymeleaf.....	23
4.2. MySQL.....	23
4.3. Platforma Wit.ai	23
4.4. Biblioteka Morfologik	23
4.5. Słownik anotacji emocjonalnej	24
5. Projekt aplikacji.....	25
5.1. Działanie aplikacji.....	25
5.1.1. Wybór filmu.....	25
5.1.2. Historia recenzji	26
5.1.3. Przeprowadzanie dialogu.....	27
5.1.4. Podsumowanie oceny.....	27
5.2. Algorytm przeprowadzania konwersacji	28
5.2.1. Nauczenie platformy Wit.ai.....	29
5.2.2. Wykonanie zapytania do platformy	31
5.2.3. Analiza wyników oceny.....	32
5.2.4. Odpowiedź chatbota	34
5.2.5. Budowanie wypowiedzi użytkownika.....	35

5.3.	Algorytm oceny emocji w wypowiedzi	35
5.3.1.	Usunięcie elementów niemożliwych do poddania ocenie.....	36
5.3.2.	Podział na listę zdań i słów.....	36
5.3.3.	Stemming wyrazów	37
5.3.4.	Stworzenie listy wyrażeń.....	38
5.3.5.	Ocena anotacji emocjonalnej wyrażeń.....	39
5.3.6.	Wystawienie oceny końcowej.....	40
6.	Podsumowanie	42
Literatura.....		44

1. Wstęp

W ostatnich latach znacząco zwiększyło się wykorzystanie sztucznej inteligencji. Jednym z jej zastosowań, które obecnie znacząco zyskuje na popularności są chatboty. Według szacowań do 2025 roku wartość rynku chatbotów będzie wzrastać o około 25% rocznie. Wykorzystanie sztucznej inteligencji jako rozmówcy największe zastosowanie znalazło jako wirtualny doradca, szczególnie w sektorze firm zajmujących się handlem. Jednakże coraz bardziej popularne stają się również inne zastosowanie chatbota, jak np. wykorzystanie go w procesie rekrutacji pracowników.

W dzisiejszych czasach aby chatbot zyskał popularność na rynku nie może korzystać z szablonów które zawierają gotowe odpowiedzi na pytania, ponieważ rozmówca może odnieść wrażenie rozmowy z komputerem. Aby tego uniknąć należy wyposażyć bota w tzw. „inteligencję”. Można to zrobić wykorzystując technologie NLP (Natural Language Processing) czyli przetwarzanie języka naturalnego. Na dzień dzisiejszy najpopularniejsze platformy umożliwiające analizę języka pozwalają na analizę języka polskiego na bardzo podstawowym poziomie, przez co konstruowanie botów w naszym języku ojczystym jest zadaniem o bardzo wysokim poziomie trudności.

Problem wykrywania emocji w tekście jest kolejną z napotkanych trudności podczas tworzenia tej pracy. W związku z tym, że emocja jest pojęciem bardzo abstrakcyjnym bardzo często mamy problem z jej odpowiednim odczytaniem oraz sklasyfikowaniem. Do najłatwiejszych sposobów odczytywania emocji należą mikroekspresje czyli niewielkie ruchy mięśni twarzy oraz wysokość i barwa głosu [1]. Niestety tych sposobów nie możemy wykorzystać do analizy emocji w tekście pisanym. Tutaj z pomocą przychodzi nam klasyfikacja natężenia emocji oraz zaimplementowany algorytm oceny wydźwięku wypowiedzi, który dokonując analizy użytych wyrażen przez rozmówcę jest w stanie wystawić ocenę końcową anotacji emocjonalnej ocenianego tekstu.

Celem pracy było zaimplementowanie aplikacji webowej w technologii Java z wykorzystaniem jej najpopularniejszych frameworków oraz bibliotek umożliwiającej użytkownikowi ocenę wybranego filmu poprzez przeprowadzenie konwersacji z chatbotem oraz otrzymanie końcowej oceny swojej recenzji w skali od 0 do 10 po analizie wydźwięku wypowiedzi przez zaimplementowany w aplikacji algorytm oceniający. Umożliwienie przeprowadzenia dialogu z botem wykonane zostało przy użyciu platformy analizy języka naturalnego, która oferowała analizę języka polskiego – Wit.ai. Do oceny wydźwięku emocjonalnego wypowiedzi wykorzystano słownik anotacji emocjonalnej, który jest częścią słownika semantycznego „Słowosieć”, a zbudowany został na Politechnice Wrocławskiej

i jak podaje oficjalna strona projektu jest dostępny „bezpłatnie do powszechnego (również komercyjnego) użytku w oparciu o licencję wzorowaną na licencji Princeton WordNet” [2] oraz bibliotekę „Morfologik”, która oferuje możliwość stemmingu [3] czyli usunięcia końcówki fleksyjnej wyrazu dla języka polskiego.

2. Chatboty

2.1. Historia chatbotów

Pierwszy chatbot o nazwie ELIZA powstał w 1966. Jego autorem był Joseph Weizenbaum. Pełnił on rolę psychoterapeuty, ale nie posiadał żadnej wiedzy medycznej. Używał on bardzo prostych reguł. Wychwytywał słowa kluczowe, podstawiał do wzorca i zwracał odpowiedź zawierającą dużą część tego co powiedział rozmówca [4]. Przełom w tworzeniu botów nastąpił w roku 2000. Wtedy na wzór chatbota ELIZA powstał program o nazwie A.L.I.C.E. stworzony przez dr Richarda Wallace. Program starał się oddawać heurystykę zwykłej konwersacji. Wykorzystywał on język AIML. Kod programu jest darmowy, dlatego na jego podstawie powstają inne chatboty. Bot A.L.I.C.E nie był jeszcze w stanie przejść wymyślonego w 1950 roku przez Alana Turinga testu zwanego „Testem Turinga” [5]. Test ten to sposób na określenie zdolności maszyny posługiwania się językiem naturalnym. Podczas kilkuminutowego testu bot poprzez rozmowę musi sprawić, że sędziowie nie będą w stanie jednoznacznie ocenić, czy rozmawiali z maszyną czy człowiekiem. Pierwszym botem któremu udało się pomyślnie ukończyć test był program który udawał 12-letniego chłopca Eugene’a Gostmana w 2014 roku. Udało mu się przekonać ponad 33% sędziów, że jest człowiekiem.

W chwili obecnej w internecie znajduje się bardzo duża ilość chatbotów, które są tworzone przy wykorzystaniu gotowych platform i można z nimi porozmawiać na bardzo różne tematy. Jednym z najpopularniejszych obecnie chatbotów jest CleverBot. Bot powstał w 1997 roku. Główną rzeczą odróżniającą go od pozostałych jest to, że nie ma zaprogramowanych odpowiedzi. Zamiast tego uczy się na podstawie wcześniejszych dialogów i na tej podstawie tworzy odpowiedź. Według statystyk bot rozmawia z około 80 000 osób jednocześnie.

2.2. Wykorzystanie chatbotów

Podstawowe wykorzystanie chatbotów to użycie ich do uzupełnienia działań marketingowych firm. Implementowanie chatbotów odbywa się na stronach internetowych oraz najpopularniejszym obecnie komunikatorze Facebook Messenger [6]. Aplikacja ta ma zasięg ponad miliarda osób na całym świecie. Chatbot sprawdza się przy wykonywaniu wielu czynności.

Przykładowe z nich to:

- Obsługa klienta
- Zastępowanie newslettera
- Przeprowadzanie loterii oraz konkursów
- Spersonalizowane powiadomienia
- Doradca produktowy

Innym ważnym zastosowaniem chatbota jest możliwość budowania marki firmy na rynku. Bardzo dobrym tego przykładem jest gra „Milionerzy” która wykorzystuje chatbota w aplikacji Messenger. Przez pierwsze dwa tygodnie wysłano do niego około 8,5 mln wiadomości. Przyczyniło się to znacząco do wzrostu popularności teleturnieju o takiej samej nazwie jak bot nadawanego przez stację TVN.

Kolejnym miejscem, gdzie wykorzystywanie chatbotów staje się coraz bardziej popularne jest proces rekrutacji pracownika. Jego głównym atutem w wyżej wymienionym procesie jest zapamiętywanie historii konwersacji. W każdej chwili możemy prześledzić rozmowę z potencjalnym pracownikiem i przypomnieć sobie informacje o kandydacie. Innym atutem jest to, że po każdym wybraniu oferty przez kandydata uczy on sztuczną inteligencję jakie parametry posiadają kandydaci rekrutujący na dane stanowisko.

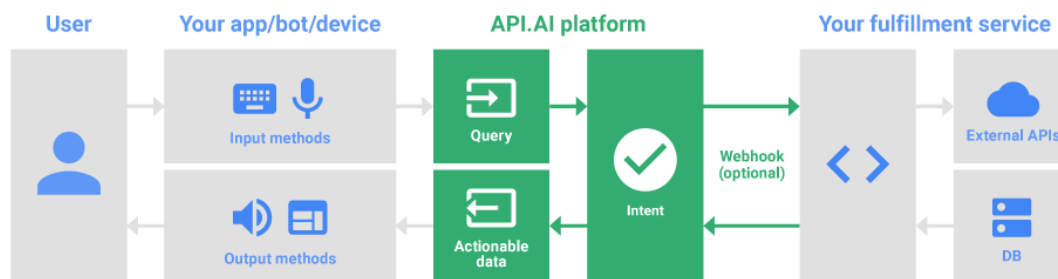
Popularność chatbotów wzrasta z każdym rokiem już dziś oprócz wyżej wymienionych zastosowań możemy spotkać go rezerwując przejazd taksówką, zapisując się do fryzjera oraz w komunikacji wewnątrz firmowej. Możliwości wykorzystania bota są praktycznie nieograniczone. Dziś możemy się tylko zastanawiać, gdzie jeszcze chatboty znajdą zastosowanie w przyszłości.

2.3. Platforma Wit.ai

Jedną z najważniejszych a zarazem najtrudniejszych czynności podczas tworzenia chatbota jest przetwarzanie języka naturalnego (ang. Natural Language Processing). Rynek oferuje wiele platform i narzędzi umożliwiające rozwiązanie tego problemu. Na potrzeby stworzenia aplikacji do tego celu wybrano narzędzie Wit.ai [7] które jest w pełni darmowe oraz oferuje wsparcie dla ponad 50 języków, w tym dla najbardziej nas interesującego – języka polskiego. Wsparcie na chwilę obecną jest co prawda w wersji beta przez co napotkano wiele problemów z dokładną analizą, ale jest to jedyna platforma, na rynku która

oferuję wsparcie dla naszego języka ojczystego. Głównym zastosowaniem Wit.ai jest tworzenie chatbotów na aplikację Facebook Messenger, która de facto jest właścicielem tego narzędzia. Jednak dzięki swojemu rozbudowanemu API możemy za pomocą zapytania http wykorzystywać tę platformę w innego typu aplikacjach.

Zasada działania platformy jest bardzo prosta i została pokazana na poniższym rysunku.



Rysunek 2-1 Zasada działania platformy Wit.ai

Źródło: <http://site.clairvoyantsoft.com/google-api-artificial-intelligent-assistant/>

Najpierw użytkownik wprowadzając przykłady uczy platformę rozpoznawania naszych intencji. Następnie podczas wysłania zapytania system sam rozpoznaje intencję na podstawie wyuczonego modelu, i zwraca wyniki rozpoznania wraz z pewnością z jaką dana intencja została rozpoznana w wypowiedzi jako odpowiedź w formacie JSON. Czym bardziej nasz model został wytrenowany tym bardziej precyzyjne będą zwracane odpowiedzi. Ciekawą funkcjonalnością oferowaną przez platformę jest tzw. „pudełko” do którego trafiają wszystkie zapytania do naszej platformy. Możemy w niej potwierdzić odpowiedzi bota lub ją poprawić zwiększając przy tym poziom wytrenowania naszego modelu.

2.4. Pozostałe platformy tworzenia chatbotów

Na rynku istnieje wiele rozwiązań pozwalających w prosty sposób tworzyć chatboty oraz przetwarzać mowę naturalną. Często są one bardziej rozwinięte od wybranej przez nas platformy oraz tworzone przez prawdziwych potentatów na rynku technologicznym jak IBM, Google czy Microsoft. Wszystkie te platformy posiadają jedną dyskryminującą je dla nas wadę, a mianowicie nie oferują analizy języka polskiego.

Poniżej przedstawiony został opis niektórych z odrzuconych przez autora rozwiązań:

Watson – jest to bardzo rozbudowany system przetwarzania języka naturalnego stworzony przez firmę IBM. Główną zaletą tego rozwiązania jest bardzo duża liczba danych wykorzystana do wytrenowania modeli. System nie jest w pełni darmowy, lecz oferuje bezpłatnie 10 000 połączeń miesięcznie.

Dialogflow (dawniej Api.ai, Speaktioit) – jest to rozwiązanie, którego właścicielem jest firma Google. Standardowy pakiet, który jest całkowicie darmowy oferuje możliwość wysłania nielimitowanej ilości zapytań tekstowych. System ten jest nieznacznie niżej oceniany w rankingach od Watsona.

Luis – jest to usługa interpretacji języka naturalnego, którą oferuje nam firma Microsoft. Firma oferuje darmowy pakiet korzystania z systemu z określoną maksymalną ilością jednoczesnych zapytań do 5 w ciągu sekundy. Jego zaletą jest łatwość integracji z innymi rozwiązaniami dostępnymi na platformie Azure.

3. Analiza emocji w tekście

3.1. Przetwarzanie języka naturalnego

Przetwarzanie języka naturalnego (ang. Natural Language Processing, NLP) jest dziedziną, która łączy w sobie elementy sztucznej inteligencji oraz językoznawstwa [8]. Dziedzina ta wykorzystywana jest do wielu zadań m.in. tłumaczenia maszynowego, korekty pisowni, streszczania tekstów, rozstrzygania autorstwa dokumentów, wykrywania spamu czy chatbotów.

Analiza języka naturalnego jest zadaniem bardzo skomplikowanym. Do jej prawidłowego wykonania potrzebna jest wiedza o świecie oraz umiejętność posługiwania się językiem, której maszyny nie posiadają ponieważ jest to cecha typowo ludzka. Dlatego kilkuletnie dziecko mimo nieznamości reguł gramatycznych potrafi budować poprawne zdania. Z tego powodu wynikają trudności w opisanu algorytmicznym i zaimplementowaniu reguł analizy w komputerze.

Trudności przetwarzania języka naturalnego występują na wielu poziomach np.

- **Składni** – relację pomiędzy słowami w zdaniu można określać na kilka sposobów co prowadzi do różnych interpretacji tekstu.
- **Semantyki** – to samo słowo może mieć wiele znaczeń w zależności od kontekstu wypowiedzi.
- **Pragmatyki** – występowanie metafor, ironii, tautologii, żartów.

Nad tworzeniem coraz lepszych narzędzi do przetwarzania języka naturalnego pracują dzisiaj prawdziwi potentaci w branży technologicznej, którzy zatrudniają całe zespoły programistów oraz inwestują bardzo duże sumy pieniędzy, aby ich rozwiązanie stało na wyższym poziomie niż rozwiązania konkurencyjnych firm.

3.2. Klasyfikacja natężenia emocji

Pojęcie emocji jest pojęciem bardzo abstrakcyjnym przez co nie posiada jednoznacznej definicji. Jest ono przedmiotem ciągłych badań psychologów. W latach 70 XX wieku Amerykański psycholog Paul Ekman wyróżnił sześć podstawowych emocji do których należą: radość, zaskoczenie, złość, smutek, strach oraz wstręt [9]. Na podstawie każdej z tych emocji powstają inne jak np. podekscytowanie, rozczerowanie, zdezorientowanie, sfrustrowanie czy przygnębienie. Jednoznaczne określenie jakie emocje

w danym momencie towarzyszą człowiekowi jest zadaniem bardzo trudnym. Najłatwiej rozpoznaje się emocję po mimice twarzy oraz wysokości i barwie głosu. W przypadku tekstu pisanego rozpoznanie i właściwa klasyfikacja emocji autora tekstu jest właściwie niemożliwa. Tutaj z pomocą przychodzi nam inny sposób klasyfikowania emocji, czyli klasyfikacja natężenia. Stara się ona określić tylko stopień intensywności nastawienia jak np. silnie-negatywne, negatywne, umiarkowane, pozytywne, silnie-pozytywne, lub w pewnej skali np. od 0 do 10. Takie sklasyfikowanie emocji jest zadaniem dużo prostszym zarówno dla człowieka jak i maszyny.

3.3. Stosowane podejścia analizy wydźwięku

Istnieje bardzo duża ilość algorytmów stosowanych w celu przeprowadzenia analizy wydźwięku tekstu. Algorytmy te możemy podzielić na cztery kategorie. Trzy z nich do swojego działania wykorzystują słowniki wydźwięku. Znajdują się w nich wyrażenia wraz z przypisanymi do nich wydźwiękiem lub prawdopodobieństwem wydźwięku pozytywnego oraz negatywnego.

Algorytmy z pierwszej kategorii polegają na znajdowaniu słów kluczowych oraz późniejszej klasyfikacji na podstawie ilości wystąpień słów o nacechowaniu negatywnym oraz pozytywnym. Algorytmy należące do tej grupy są bardzo często stosowane z uwagi na szybkość ich działania oraz łatwość użycia, jednak są to metody dające nieprecyzyjne, często odbiegające od stanu faktycznego wyniki.

Druga kategoria to analiza nacechowania emocjonalnego. Algorytmy należące do niej są oparte już nie tylko o badanie słów o określonym wydźwięku, ale także na przypisywaniu tym słowom prawdopodobieństwa z jaką emocją dane słowo występuje najczęściej. Metoda ta często jest skuteczniejsza od podejścia stosowanego w pierwszej grupie, lecz ona również wielokrotnie przynosi nie zadowalające nas rezultaty.

Do trzeciej kategorii należą algorytmy oparte na metodach statystycznych. Znajdują się tu głównie algorytmy z dziedziny uczenia maszynowego jak np. klasyfikator Bayesowski czy sieci neuronowe. Grupa ta do swojego działania nie potrzebuje słowników. Używa jedynie danych trenujących do wytrenowania modelu. Zbiór ten musi być bardzo dużych rozmiarów, aby poprawność modelu stała na wysokim poziomie i możliwe było uzyskanie satysfakcjonujących wyników analizy. Najsłabsze wyniki uzyskiwane przy wykorzystaniu

algorytmu należącego do tej grupy otrzymujemy w przypadku pojedynczych zdań, fraz oraz krótkich tekstów.

Ostatnia kategoria zawiera algorytmy które analizują tekst na poziomie znaczeniowym. Używają one uporządkowanej wiedzy ze słowników, encyklopedii oraz innych baz danych do określenia znaczenia wyrazu w danym kontekście oraz zbadania jego wydźwięku poprzez związki z innymi wyrazami w tym zdaniu. Taka analiza jest w pełni zależna od wielkości posiadanych przez nas danych. Bardzo często algorytmy tej grupy sprowadzają się tylko do analizy podstawowych tekstów, ponieważ zgromadzone dane nie pozwalają na właściwe określenie kontekstu wyrazów tekstu napisanego językiem potocznym lub napisanego na wybrany, wyspecjalizowany temat.

3.4. Zastosowany algorytm oceny

Algorytm zastosowany do analizy wydźwięku na potrzeby tej pracy pozwala na określenie anotacji emocjonalnej tekstu w skali od 0 do 10. Do poprawnego działania wykorzystuję najlepszy polski słownik nastawienia emocjonalnego, który zawiera ponad 180 000 tysięcy wyrazów z określonym wydźwiękiem. Algorytm jest rozbudowaną wersją algorytmów należących do kategorii pierwszej i opisanych w poprzednim podrozdziale. Wyszukuje on w tekście nie pojedyncze wyrazy, lecz całe wyrażenia według zaimplementowanych w nim reguł np. przymiotnik oraz występujący po nim rzeczownik, ocenia na podstawie słownika anotację emocjonalną całego wyrażenia po czym wystawia ocenę na podstawie ocen wszystkich wyrażen występujących w tekście. Szczegóły implementacji oraz zasad działania algorytmu znajdują się w rozdziale opisującym projekt aplikacji.

4. Technologie, biblioteki, narzędzia

4.1. Java

Java według rankingu Tiobe jest obecnie najpopularniejszym językiem na świecie. Należy ona do grupy języków wysokiego poziomu w których wiele rzeczy wykonuje się automatycznie bez konieczności ingerencji programisty. Język ten jest silnie typowanym, obiekowym językiem programowania stworzonym przez firmę Sun Microsystems pod kierownictwem Jamesa Goslinga w 1995 roku. Największymi jego zaletami są popularność, dzięki czemu łatwiej znaleźć można rozwiązanie nietrywialnych problemów oraz jego przenośność, która umożliwia uruchomienie kodu na każdym sprzęcie i systemie, na którym zainstalowana jest wirtualna maszyna javy [10].

4.1.1. Java 10

Aktualna wersja stabilna która została wykorzystana do zaimplementowanie kodu aplikacji czyli Java 10 na rynek została wypuszczona w marcu 2018 roku. Nie wnosi ona tylu nowości co poprzednia wersja, ale pozwala między innymi na lokalne wnioskowanie o typie zmiennej dzięki słowu kluczowemu *var* [11].

4.1.2. Spring Boot

Spring Boot jest projektem, który bazuje na Springu. Zapewnia on prostszy i szybszy sposób konfigurowania oraz uruchamiania aplikacji wykonanych przy wykorzystaniu frameworka Spring. Dzięki niemu nie musimy samemu dbać o prawidłowe zależności, pomiędzy bibliotekami które dołączamy do projektu oraz nie musimy posiadać zewnętrznego serwera np. WildFly lub Tomcat do uruchamiania i testowania aplikacji. Spring jest platformą Javową która posiada wsparcie dla wielu technologii i stanowi jeden z kluczowych sposobów tworzenia aplikacji w Javie [12][13].

4.1.3. Maven

Apache Maven jest narzędziem, które automatyzuje tworzenie oprogramowania na platformę Java. Jest on rozpowszechniany na licencji Apache License tak jak inne produkty pochodzące z fundacji Apache. Oprócz bardzo dużych oszczędności czasu programisty to

narzędzie posiada wiele innych zalet jak np. pilnowanie bibliotek i ich wersji, automatyczne wykonywanie testów, generowanie pliku jar, automatyczne pobieranie odpowiednich wersji bibliotek czy generowanie dokumentacji.

4.1.4. Thymeleaf

Thymeleaf jest silnikiem szablonów Java które służą do tworzenia stron w technologiach HTML, CSS oraz JavaScript. Narzędzie jest w pełni kompatybilne z frameworkiem Spring oraz rozszerzalne przy pomocy nowych dialektów. Jego głównym celem jest wprowadzenie estetycznych szablonów które mogą być poprawnie wyświetlane w przeglądarkach oraz działać jako statyczne prototypy.

4.2. MySQL

MySQL to wolno dostępny system służący do zarządzania relacyjnymi bazami danych. W chwili obecnej system ten jest rozwijany przez firmę Oracle. Do największych zalet tego silnika baz danych należy na pewno zaliczyć skalowalność, uniwersalność oraz ochronę danych. W projekcie system został wykorzystany do obsługi bardzo prostej, składającej się z dwóch tabel bazy danych, która przechowuje informację o filmach oraz ich recenzjach.

4.3. Platforma Wit.ai

Platforma Wit.ai służy do przetwarzania języka naturalnego. W aplikacji zostało wykorzystane udostępniane przez tą platformę API. Dzięki niemu możemy za pomocą zapytań wykorzystujących protokół http kierować wypowiedzi użytkowników do przeanalizowania przez platformę oraz otrzymywać rezultat analizy w formacie JSON do dalszego wykorzystania podczas działania aplikacji.

4.4. Biblioteka Morfologik

Biblioteka Morfologik jest projektem w całości tworzonym przez polskich programistów. Umożliwia ona wykonanie stemmingu czyli wydobyciu z wyrazu jego

rdzenia oraz lematyzacji czyli sprowadzenie grupy wyrazów która stanowi odmianę danego zwrotu do wspólnej postaci. Ponadto określa ona do jakiej części mowy należy dany wyraz. Biblioteka Morfologik jest obecnie jedynym dostępnym na rynku narzędziem, które potrafi przeprowadzić wyżej wymienione operacje na wyrazach pochodzących z języka polskiego. Jest ona udostępniana na licencji, która umożliwia wykorzystanie jej bez ponoszenia żadnych kosztów.

4.5. Słownik anotacji emocjonalnej

Słownik anotacji emocjonalnej który został wykorzystany na potrzeby stworzenia tej aplikacji jest częścią słownika semantycznego Słowosieć. „Polski wordnet” jak jest nazywany ten słownik jest tworzony wspólnym wysiłkiem leksykografów oraz informatyków z Grupy Technologii Językowych Politechniki Wrocławskiej. Decyzją władz uczelni został udostępniony bezpłatnie do powszechnego wtedy również komercyjnego użytku. Obecnie słownik liczy ponad 80 000 tysięcy różnych wyrazów które mają określoną anotację emocjonalną.

5. Projekt aplikacji

5.1. Działanie aplikacji

Główną funkcjonalnością aplikacji jest możliwość wystawienia recenzji wybranej przez siebie produkcji filmowej poprzez przeprowadzenie konwersacji z chatbotem oraz otrzymania końcowej oceny filmu na podstawie analizy użytych słów o wydźwięku emocjonalnym w swojej wypowiedzi. Aplikacja zawiera cztery podstrony wyświetlające listę filmów, listę recenzji, stronę do przeprowadzenia rozmowy z chatbotem oraz stronę podsumowującą opinię. Została stworzona również strona główna na której znajdują się informacje o autorze oraz celu wykonania aplikacji. Główną wartość aplikacji stanowią wykonane algorytmy prowadzenia konwersacji oraz analizy wydźwięku. Zostały one szczegółowo opisane w następnych podrozdziałach.

5.1.1. Wybór filmu

Na stronie wyboru filmu użytkownik może wybrać film, którego chciałby dokonać oceny z tabeli zawierającej listę filmów dodanych przez administratora. W tabeli zostały umieszczone informacje o tytule, reżyserze oraz roku, w którym dana produkcja filmowa miała swoją premierę. Aby przejść do podstrony zawierającej bardziej szczegółowe informacje o danym filmie należy w tabeli kliknąć przycisk „Więcej” w rzędzie zawierającym informacje o danej produkcji. Rysunek 5-1 ukazuje tabelę z przykładowymi filmami.

 Uniwersytet Rzeszowski

Praca inżynierska
Implementacja aplikacji Chatbot reagującej na emocje rozmówcy

Lista filmów


Poniżej znajduje się lista filmów dodanych do serwisu. Aby dokonać oceny filmu wejdź w jego szczegóły.

L.p	Tytuł	Reżyser	Rok	Szczegóły
1	Najlepszy	Lukasz Palkowski	2017	Więcej
2	Pianista	Roman Polański	2002	Więcej
3	Wołyń	Wojciech Smarzowski	2016	Więcej
4	Jak rozpętałem drugą wojnę światową	Tadeusz Chmielewski	1969	Więcej
5	Chłopaki nie płaczą	Olaf Lubaszenko	2000	Więcej

Rysunek 5-1 Podstrona zawierająca listę filmów

5.1.2. Historia recenzji

Na podstronie zawierającej szczegółowe informacje o filmie możemy zobaczyć historię dodanych ocen dla danego filmu. Oceny te zawierają wartość liczbową w skali od 0 do 10 wystawioną przez algorytm oraz pełną treść wypowiedzi recenzenta o danym filmie. W centralnym punkcie strony znajdują się przycisk „Oceń film” po kliknięciu, którego użytkownik zostanie przeniesiony do strony umożliwiającej dodanie własnej oceny do filmu. Na poniższym rysunku 5-2 została pokazana strona z przykładową recenzją dodaną do filmu pt. „Wołyń” w reżyserii Wojciecha Smarzowskiego.

 Uniwersytet Rzeszowski

Praca Inżynierska
Implementacja aplikacji Chatbot reagującej na emocje rozmówcy

Tytuł: Wołyń
Reżyser: Wojciech Smarzowski
Rok produkcji: 2016

Oceń film

Dodane oceny

Film został oceniony na: 8.9/10

Gra aktorska była na zróżnicowanym poziomie. Michalina Łabacz, odtwórczyni głównej roli w filmie wypadła w sposób fenomenalny. Znakomicie odegrała emocję w odpowiednich scenach filmu. Natomiast aktorzy drugoplanowi całkowicie nie popisali się swoimi występami pozostawiając swoje postacie bezbarwne.. Fabuła zatacza koło. Rozpoczyna się przecież we żniwa i kończy się w żniwa. Tuż przed napaścią na wieś banderowców sołtys Szuma mówi do Zosi, że trzeba zakończyć żniwa. Z kolei film rozpoczyna się podczas żniw, weselem, a kończy się śmiercią ówczesnych państwa młodych i mistycznymi zaślubinami Zosi z Petrem. Narzędzia rolnicze i zwyczaje weselne używane na początku do zabaw weselnych, na koniec służą do mordów. Wesele na początku kończy odjazd wozu ślubnego i taki sam wóz widzimy na końcu filmu. Zwrot akcji następuje po śmierci księdza i Skiby, wiodących postaci pierwszego wesela. Od tego momentu może rozpocząć się mistyczna droga do zaślubin Zosi z Petrem. Oczywiście poprzez śmierć, no bo przecież Petro nie żyje. Na końcu pojawia się znów motyw raju, z tym, że zupełnie innego niż na początku.. Niby wsiąka się w wołyńską wieś przedstawioną w obiektywie Piotra Sobocińskiego Jr. oraz tamtejszą atmosferą – perfekcyjna scenografia, kostiumy i język ludności plus oszczędnie dawkowane brzmienia klarnetu Mikołaja Trzaski robią wrażenie (brawa za pieczołowicie oddany świat epoki), ale ilość pojawiających się i znikających bohaterów na dalszym planie i brak większej głębi czy relacji między nimi nie pozwala widzowi jeszcze mocniej przejąć się ich losami.


Copyright 2018 © Jakub Kusnierz


Rysunek 5-2 Podstrona przedstawiająca wystawione dane filmowi oceny


5.1.3. Przeprowadzanie dialogu

Na stronie służącej ocenie filmu jesteśmy proszeni o udzielanie treściwych odpowiedzi na pytania zadawane na przez chatbota. Swoje odpowiedzi umieszczamy w przygotowanym do tego oknie umieszczonym pod ostatnią wiadomością w konwersacji oraz wysyłamy je przyciskiem „Wyślij”. Wypowiedzi chatbota są oznaczone kolorem jasnoszarym i zawierają po lewej stronie ikonę robota (patrz rysunek 5-3). Natomiast wypowiedzi użytkownika są koloru ciemnoszarego i po prawej stronie posiadają ikonę człowieka. Chatbot zadaje pytania w sposób losowy pytając o wymagane do końcowej oceny kategorie po każdym pytaniu oczekując na odpowiedź użytkownika. W przypadku udzielenia odpowiedzi nie na temat lub za krótkiej do oceny w końcowej analizie wydźwięku bot poinformuje nas o tym oraz zada inne pytanie dotyczące tego samego tematu. Zostało to ukazane w końcowych wypowiedziach na rysunku 5-3.


Oceniasz film "Wołyń" w reżyserii Wojciech Smarzowski


Witaj! Dziękuję za chęć oceny filmu. Proszę aby Twoje odpowiedzi na zadawane pytania były pełnymi zdaniami.
A więc zaczynamy!


Jak oceniasz fabułę filmu?

Twórcom jakoś udało się tak poprowadzić fabułę, by widz nie dał rady wszystkiego przewidzieć. Akcja rozwija się dynamicznie i jest całkiem emocjonująca, a gdy już na naszych oczach rozgrywa się drugi punkt zwrotny, ma się wrażenie, że wspomniany brak przewidywalności jest tak samo zaletą, co wadą.

Jak oceniasz grę aktorów?

Fabuła była bardzo interesująca

Twoja odpowiedź jest nie na temat. Postaraj się udzielać bardziej dokładnych odpowiedzi.
Jak ocenisz grę głównego bohatera?



Rysunek 5-3 Fragment rozmowy z chatbotem

5.1.4. Podsumowanie oceny

Podstrona podsumowania oceny filmu jest najbardziej rozbudowaną w treść stronę aplikacji. Jej najważniejszą część stanowią utworzona z pojedynczych wypowiedzi podczas konwersacji opinia użytkownika zawierająca ocenę całego filmu oraz wystawiona przez

algorytm analizujący anotację emocjonalną w tekście końcowa ocena filmu. Zostało to pokazane na poniższym rysunku 5-4.

Podsumowanie oceny filmu "Wołyń" w reżyserii Wojciech Smarzowski

Dziękujemy za Twoją opinię o filmie. Poniżej możesz zobaczyć jaką ocenę filmu wystawił algorytm na podstawie Twojej opinii, oraz zobaczyć w jaki sposób ta ocena została wystawiona.

Twoja opinia:

Gra aktorska była na zróżnicowanym poziomie. Michalina Łabacz, odtwórczyni głównej roli w filmie wypadła w sposób fenomenalny. Znakomicie odegrała emocję w odpowiednich scenach filmu. Natomiast aktorzy drugoplanowi całkowicie nie popisali się swoimi występami pozostawiając swoje postacie bezbarwne. Fabuła zatacza koło. Rozpoczyna się przecież we żniwa i kończy się w żniwa. Tuż przed napadnięciem na wieś banderowców sołtys Szuma mówi do Zosi, że trzeba zakończyć żniwa. Z kolei film rozpoczyna się podczas żniw, weselem, a kończy się śmiercią ówczesnych państwa młodych i mistycznymi zaślubinami Zosi z Petrem. Narzędzia rolnicze i zwyczajnie weselne używane na początku do zabaw weselnych, na koniec służą do mordów. Wesele na początku kończy odjazd wozu ślubnego i taki sam wóz widzimy na końcu filmu. Zwrot akcji następuje po śmierci księdza i Skiby, wiodących postaci pierwszego wesela. Od tego momentu może rozpocząć się mistyczna droga do zaślubin Zosi z Petrem. Oczywiście poprzez śmierć, no bo przecież Petro nie żyje. Na końcu pojawia się znów motyw rajdu, z tym, że zupełnie innego niż na początku.. Niby wsiąka się w wołyńską wieś przedstawioną w obiektywie Piotra Sobocińskiego Jr. oraz tamtejszą atmosferę – perfekcyjna scenografia, kostiumy i język ludności plus oszczędnie dawkowane brzmienia klarnetu Mikołaja Trzaski robią wrażenie (brawa za pieczętowanie oddany świat epoki), ale ilość pojawiających się i znikających bohaterów na dalszym planie i brak większej głębi czy relacji między nimi nie pozwala widzowi jeszcze mocniej przejąć się ich losami.

Wystawiona ocena filmu: 8.9/10

Rysunek 5-4 Fragment strony podsumowania oceny filmu

Pozostałą treść tej podstrony stanowi skrócony opis algorytmu zawierający wyjaśnienie na jakiej podstawie została wystawiona taka a nie inna ocena. Zostały w nim zawarte informacje o wysokości prawdopodobieństwa klasyfikowania poszczególnych wypowiedzi użytkownika do odpowiednich kategorii pytań oraz ocena wszystkich wyrażenń znalezionych w wypowiedzi z wyszczególnioną oceną każdej ze składowych wyrażenia. Dane te zostały zaprezentowane w kolejnych podrozdziałach, w których szczegółowo opisano działanie algorytmów wykorzystanych w aplikacji.

5.2. Algorytm przeprowadzania konwersacji

Algorytm przeprowadzania konwersacji jest pierwszym z dwóch najważniejszych algorytmów zaimplementowanych na potrzeby wykonania aplikacji. Jego zadaniem jest automatyczne przeprowadzenie dialogu z użytkownikiem w taki sposób, żeby uzyskać odpowiedzi na wszystkie konieczne do późniejszej analizy wydźwięku emocjonalnego pytania. W algorytmie została wykorzystana platforma Wit.ai w której został wytrenowany

do poziomu dającego satysfakcjonujące wyniki model potrafiący poprawnie klasyfikować wypowiedzi użytkownika do wyuczonych kategorii. Szczegółowy opis algorytmu został umieszczony w poniższych podrozdziałach, gdzie krok po kroku zostały opisane wszystkie przeprowadzane operacje.

5.2.1. Nauczenie platformy Wit.ai

Pierwszy krok wykonania algorytmu to wytrenowanie modelu klasyfikującego wypowiedzi użytkownika do różnych kategorii oceny filmu. Platforma oferuje możliwość wykrywania znanych już jej intencji w wypowiedziach jak np. wiek osoby, czas, kwota do zapłaty czy lokalizacja lub tworzenie nowych intencji i trenowanie ich samemu. Aby wytrenować model wpisujemy tekst liczący poniżej 280 znaków i potwierdzamy znalezione przez sztuczną inteligencję tzw. Encję lub intencję. W przypadku niepoprawnego rozpoznania intencji przed potwierdzeniem korygujemy ręcznie błędy. Na rysunku 5-5 ukazany został proces trenowania modelu na pojedynczym przykładzie.

Twórcom jakoś udało się tak **poprowadzić fabułę**, by **widz nie dał rady wszystkiego przewidzieć**. Akcja rozwija się dynamicznie i jest **całkiem emocjonująca**, a gdy już na naszych oczach rozgrywa się drugi **punkt zwrotny**, ma się wrażenie, że wspomniany **brak przewidywalności** jest tak zaletą.

<input type="radio"/> fabuła	poprowadzić fabułę	0.958
<input type="radio"/> fabuła	widz nie dał rady wszystkiego ...	0.958
<input type="radio"/> fabuła	Create new value "Akcja rozwi..."	
<input type="radio"/> fabuła	punkt zwrotny	0.956
<input type="radio"/> fabuła	brak przewidywalności	0.842

➕ Add a new entity

✓ Validate

Rysunek 5-5 Trenowanie modelu na przykładowej wypowiedzi

Źródło: www.wit.ai

Na potrzeby przeprowadzenia dialogu w aplikacji zostało na platformie wit.ai wytrenowane rozpoznawanie w modelu trzech encji. Są to:

- Fabuła
- Gra aktorska
- Efekty specjalne

Encję zostały wytrenowane dużą ilością przykładów pochodzącymi z własnej inwencji twórcy aplikacji oraz z komentarzy pod filmami oraz serialami na platformie filmweb.pl która jest największym polskim serwisem filmowym zawierającym największą bazę filmów oraz aktorów. Na rysunku 5-6 przedstawiony został wykaz wytrenowanych encji z platformy wit.ai wraz z przykładowymi wartościami na podstawie których następuje późniejsza klasyfikacja (kolumna „Values” na rysunku).

Your app uses 3 entities

Entity	Description	Values
fabula → LOOKUP STRATEGIES free-text & keywords	User-defined entity	Akcja rozwija się dynamicznie, Film jest wielowątkowy, Niesamowicie wciągają przygody, Fabuła może wydawać się, brak przewidywalności, Akcja rozwija się, punkt zwrotny, widz nie dał rady wszystkiego przewidzieć, poprowadzić fabułę, Zakonczenie historii było znakomite
Gra_aktorska → LOOKUP STRATEGIES free-text & keywords	User-defined entity	Froda, Role wymagały poświęcenia, bohaterów będzie odgrywało rolę, jej nadnaturalne zdolności, Jest osobiwością, Millie Bobby Brown, objawienie pierwszego sezonu, charyzmy, posiada on jednak nieograniczone zasoby uroku osobistego, Sceny z jego udziałem
efekty_specjalne → LOOKUP STRATEGIES free-text & keywords	User-defined entity	będą cieszyć zmysły widza przez cały seans, doskonale zdjęcia, Świetny montaż, niewątpliwych walorach technicznych, tytułową piosenkę, za sprawą tak genialnych kawałków jak "Imperial March", "The Battle of Hoth" oraz niezapomnianego tematu tytułowego, The Battle of Hoth, niezapomnianego tematu tytułowego, Imperial March, genialnych kawałków

Rysunek 5-6 Wytrenowane encję na potrzeby aplikacji

Źródło: www.wit.ai

Ważną rolę w procesie trenowania modelu odgrywa tzw. Pudełko. Trafiają do niego wszystkie wiadomości wysłane zapytaniem http do platformy wit.ai. Dzięki temu administrator dostaje możliwość późniejszego poprawiania wytrenowania modelu, jeżeli zauważy błąd w klasyfikowaniu wypowiedzi do odpowiedniej intencji. Takie rozwiązanie udostępnione przez platformę znacząco ułatwia poprawienie skuteczności analizy wypowiedzi w momencie, gdy z chatu korzystają inne osoby i zwiększa się napływ nowych, różnorodnych wypowiedzi użytkowników. Na rysunku 5-7 ukazane zostały przykładowe wiadomości, które trafiły do pudełka w procesie testowania aplikacji poprzez konwersacji z chatbotem i oczekują na zatwierdzenie lub skorygowanie błędów przez administratora.

Wprowadź niektóre dialogi i **monologi są nieco zbyt przerysowane**, ale za to z pasją ogląda się zwłaszcza **sceny** batalistyczne. Zachwyca świat Śródziemia. **Błędy scenografów**, operatorów i montażyistów można policzyć na palcach jednej ręki

<input type="radio"/> fabula	Create new value "monologi s..."	0.683
<input type="radio"/> fabula	sceny	1.000
<input type="radio"/> efekty_spejalne	Create new value " Błędy scen..."	

gra aktorska była bardzo dobra

<input type="radio"/> Gra_aktorska	Gra aktorska	0.999
------------------------------------	--------------	-------

Rysunek 5-7 Wykaz wiadomości z tzw. pudełka

Źródło: www.wit.ai

5.2.2. Wykonanie zapytania do platformy

Wykonanie requestu czyli zapytania zostało zaimplementowane w Javie i odbywa się przy wykorzystaniu klasy `URLConnection`. Aby zapytanie zostało poprawnie wykonane należy jako nagłówek „Authorization” dołączyć wygenerowany przez platformę `wit.ai` dla naszego konta indywidualny, niepowtarzalny token. Implementacja requestu sprawdza czy w odpowiedzi otrzymaliśmy oczekiwany przez nas kod odpowiedzi protokołu `http` – 200, który oznacza poprawność wykonania zapytania. W przypadku nie satysfakcjonującego nas kodu otrzymanej odpowiedzi, algorytm informuje nas o tym. Jeżeli zapytanie do platformy zostało wykonane poprawnie zostaje odczytana zwrócona odpowiedź serwera w formacie `JSON` która zawiera w sobie informację o wszystkich rozpoznanych w wypowiedzi encjach wraz z pewnością z jaką zostały one rozpoznane.

Jedną z napotkanych trudności przy wykonywaniu zapytań jest ograniczona do 280 znaków długość wypowiedzi jaką można przeanalizować na platformie w pojedynczym zapytaniu. Aby umożliwić analizę dłuższych wypowiedzi został zaimplementowany prosty algorytm dzielący taką wypowiedź na zdania o długości mieszczącej się w określonym limicie oraz wykonujący dla nich odpowiednią ilość jednoczesnych zapytań, sumując wartość otrzymanych pojedynczych odpowiedzi. Dzięki czemu aplikacja nie posiada maksymalnego limitu długości odpowiedzi użytkownika na pytania zadawane przez chatbota.

Na rysunku 5-8 przedstawiony został kod odpowiadający za obsłużenie wysłania zapytania oraz odczytania odpowiedzi serwera.

```
try {
    message = URLEncoder.encode(message, "UTF-8").replace(" ", "%20");
    URL url = new URL("https://api.wit.ai/message?v=20170218&q=" + message);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setRequestMethod("GET");
    conn.setRequestProperty("Accept", "application/json");
    conn.setRequestProperty("Authorization", "Bearer AUM4D36AW3YRXDCUDETHT7LC4Y4SOXM");
    //Sprawdzenie odpowiedzi z kodem błędu
    if (conn.getResponseCode() != 200) {
        throw new RuntimeException("Failed : HTTP error code : "
            + conn.getResponseCode());
    }
    //Odczytanie treści odpowiedzi
    BufferedReader br = new BufferedReader(new InputStreamReader(
        conn.getInputStream()));
    //Złożenie odpowiedzi w całość
    String temp;
    StringBuilder response = new StringBuilder();
    while ((temp = br.readLine()) != null) {
        response.append(temp);
    }
    conn.disconnect();
    return String.valueOf(response);
} catch (IOException e) {
    e.printStackTrace();
}
```

Rysunek 5-8 Kod wykonujący zapytanie do platformy wit.ai

5.2.3. Analiza wyników oceny

Po otrzymaniu odpowiedzi od platformy zawierającej wyniki analizy wysłanej przez użytkownika odpowiedzi na zadane pytanie przystępujemy do jej analizowania. Pierwszą czynnością analizy jest obliczenie sumy prawdopodobieństw określonych dla wszystkich wystąpień w wypowiedzi słów kluczowych po jakich dana wypowiedź została zakwalifikowana do określonej przez platformę intencji. Algorytm obliczania opisanej czynności przedstawiony został na rysunku 5-9.

```
private static void calculateSumConfidence(Map<String, Double> sumConfidence, WitResponse witResponse) {
    for (Entities entity : witResponse.getEntitiesArrayList()) {
        double sum = 0;
        for (Value value : entity.getValueArrayList()) {
            sum += value.getConfidence();
        }
        if (!sumConfidence.containsKey(entity.getName())) {
            sumConfidence.put(entity.getName(), sum);
        } else sumConfidence.replace(entity.getName(), sumConfidence.get(entity.getName()) + sum);
    }
}
```

Rysunek 5-9 Obliczenie sumy prawdopodobieństw

Następnym krokiem analizy wyników oceny jest określenie dla której intencji odpowiedź użytkownika uzyskała największe prawdopodobieństwo sklasyfikowania. Odbywa się to poprzez znalezienie w strukturze danych wartości maksymalnej oraz zwrócenie nazwy intencji dla której ta wartość została znaleziona.

```
private static String getMaxKey(Map<String, Double> sumConfidence) {
    if (sumConfidence.size() > 1) {
        return Collections.max(sumConfidence.entrySet(), Comparator.comparingDouble(Map.Entry::getValue)).getKey();
    } else if (sumConfidence.size() == 1) return sumConfidence.entrySet().iterator().next().getKey();
    else return "";
}
```

Rysunek 5-10 Znajdywanie wartości maksymalnej

Aby suma prawdopodobieństwa została uznana za przekonującą przez algorytm musi dawać minimum 50% szansy, że wypowiedź została sklasyfikowana do oczekiwanej przez nas kategorii. W przeciwnym wypadku odpowiedź użytkownika zostaje uznana za wypowiedź nie na temat, która nie odpowiada poprawnie na pytanie zadane przez chatbota.

Rysunek 5-11 jest częścią strony zawierającej podsumowanie oceny użytkownika. Przedstawia on do której z intencji została zakwalifikowana każda z odpowiedzi oceniającego oraz jakie miała prawdopodobieństwo sklasyfikowania do każdej z wytrenowanych w modelu encji.

Kategoria	Cześć wypowiedzi	Klasyfikacja do kategorii
Fabula	Fabula zatacza koło. Rozpoczyna się przecież we żniwa i kończy się w żniwa. Tuż przed napadnięciem na wieś banderowców sołtys Szuma mówi do Zosi, że trzeba zakończyć żniwa. Z kolei film rozpoczyna się podczas żniw, weselem, a kończy się śmiercią ówczesnych państwa młodych i mistycznymi zaślubinami Zosi z Petrem.	Gra aktorska: 0.0% Fabula: 100.0% Efekty specjalne: 0.0%
Gra aktorska	Gra aktorska była na różnicowanym poziomie. Michalina Łabacz, odtwórczyni głównej roli w filmie wypadła w sposób fenomenalny. Znakomicie odegrała emocje w odpowiednich scenach filmu. Natomiast aktorzy drugoplanowi całkowicie nie popisali się swoimi występami pozostawiając swoje postacie bezbarwne.	Gra aktorska: 95.0% Fabula: 5.0% Efekty specjalne: 0.0%
Efekty specjalne	Perfekcyjna scenografia, kostiumy i język ludności plus oszczędnie dawkowane brzmienia kłarnetu Mikołaja Trzaski robią wrażenie (brawa za pieczołowicie oddany świat epoki), ale ilość pojawiających się i znikających bohaterów na dalszym planie i brak większej głębi czy relacji między nimi nie pozwala widzowi jeszcze mocniej przejąć się ich losami.	Gra aktorska: 26.0% Fabula: 0.0% Efekty specjalne: 74.0%

Rysunek 5-11 Fragment podsumowania oceny przedstawiający do której kategorii zostały zakwalifikowane dane wypowiedzi oraz z jakim prawdopodobieństwem

5.2.4. Odpowiedź chatbota

W aplikacji istnieją trzy możliwe warianty reakcji chatbota na naszą wiadomość. Są one uzależnione od wyników analizy opisaney w poprzednim podrozdziale.

Pierwszy wariant jest stosowany w momencie, gdy odpowiedź użytkownika będzie liczyła powyżej 120 znaków oraz podczas analizy zostanie uznana, że odpowiada na pytanie zadane przez chatbota w sposób wystarczająco dobry. W takim przypadku zostanie zadane wylosowane pytanie z kolejnej kategorii lub gdy będzie to ostatnia z kategorii chatbot podziękuję za udzielone odpowiedzi i przeniesie oceniającego to strony podsumowującej rozmowę. Wariant ten jest najbardziej oczekiwany przez algorytm, ponieważ uzyskuje wypowiedź oceniającą daną kategorię i przesuwa on rozmowę do przodu w celu wydobycia od użytkownika kolejnych informacji. Na rysunku 5-12 przedstawiono część kodu odpowiadającą za wylosowanie następnej kategorii pytań.

```
private static void randomCategoryQuestion(DialogProgress dialogProgress) throws NotFoundException {
    ArrayList<String> categoryList = new ArrayList<>();
    if (!dialogProgress.getActing()) categoryList.add(CategoryQuestions.ACTING.getValue());
    if (!dialogProgress.getStory()) categoryList.add(CategoryQuestions.STORY.getValue());
    if (!dialogProgress.getSpecialEffects()) categoryList.add(CategoryQuestions.SPECIALEFFECTS.getValue());

    if (categoryList.isEmpty()) throw new NotFoundException("Nie znaleziono dostępnych kategorii");
    else if (categoryList.size() == 1) dialogProgress.setCurrentCategoryQuestions(categoryList.get(0));
    else {
        Random generator = new Random();
        int index = generator.nextInt( bound: categoryList.size() - 1);
        dialogProgress.setCurrentCategoryQuestions(categoryList.get(index));
    }
    dialogProgress.setNumberCurrentQuestions(0);
}
```

Rysunek 5-12 Losowanie kategorii pytań

Drugi wariant odpowiedzi zastosowany zostanie w momencie, gdy odpowiedź użytkownika zostanie sklasyfikowana jako odpowiedź nie zawierająca wystarczająco dużo informacji na zadane przez chatbota pytanie. W takim wypadku chatbot poinformuje użytkownika, że udzielił złej odpowiedzi oraz poprosi, aby w przyszłości bardziej szczegółowo odpowiadał on na zadawane pytania. Zostanie również zadane kolejne pytanie dotyczące tego samego tematu, na który udzielona została niepoprawna odpowiedź.

Trzeci wariant odpowiedzi zostanie wyświetlony, gdy udzielona przez użytkownika odpowiedź będzie precyzyjnie odpowiadała na zadane pytanie, lecz jej długość będzie liczyła poniżej 120 znaków. Chatbot poprosi wtedy o rozwinięcie swojej wypowiedzi wylosowanym pytaniem ze zbioru tak zwanych pytań pomocniczych jak np. „czy możesz

rozwinąć swoją wypowiedź?”. Na rysunku 5-13 został przedstawiony algorytm odpowiedzialny za wylosowanie pytania pomocniczego.

```
private static String randomQuestionAuxiliary(DialogProgress dialogProgress) throws NotFoundException {
    ArrayList<String> questionsList = dialogProgress.getQuestionsAuxiliary();

    if (questionsList.size() == 0) throw new NotFoundException("Nie znaleziono żadnych pytań");
    else if (dialogProgress.getNumberCurrentQuestionAuxiliary() >= questionsList.size())
        dialogProgress.setNumberCurrentQuestionAuxiliary(0);

    dialogProgress.setNumberCurrentQuestionAuxiliary(dialogProgress.getNumberCurrentQuestionAuxiliary() + 1);
    return questionsList.get(dialogProgress.getNumberCurrentQuestionAuxiliary());
}
```

Rysunek 5-13 Algorytm losujący pytanie pomocnicze

5.2.5. Budowanie wypowiedzi użytkownika

Po uzyskaniu pozytywnych odpowiedzi na wszystkie zadane pytania tworzona jest podsumowująca wypowiedź użytkownika zawierająca jego odpowiedzi, które zostały uznane, że dotyczą tego samego tematu co zadane pytanie. Przykładowa recenzja, która została zbudowana z wiadomości zebranych podczas konwersacji z chatbotem i zawiera w sobie ocenę fabuły, gry aktorskiej oraz efektów specjalnych została ukazana na rysunku 5-4. Stworzona recenzja zostaje następnie przekazana do algorytmu oceniającego wydźwięk emocjonalny. Algorytm ten został szczegółowo opisany w następnym podrozdziale.

Fabula zatacza koło. Rozpoczyna się przecież we żniwa i kończy się w żniwa. Tuż przed napaścią na wieś banderowców soltys Szuma mówi do Zosi, że trzeba zakończyć żniwa. Z kolei film rozpoczyna się podczas żniw, weselem, a kończy się śmiercią ówczesnych państwa młodych i mistycznymi zaślubinami Zosi z Petrem.. Gra aktorska była na różnicowanym poziomie. Michalina Łabacz, odtwórczyni głównej roli w filmie wypadła w sposób fenomenalny. Znakomicie odegrała emocję w odpowiednich scenach filmu. Natomiast aktorzy drugoplanowi całkowicie nie popisali się swoimi występami pozostawiając swoje postacie bezbarwne.. Perfekcyjna scenografia, kostiumy i język ludności plus oszczędnie dawkowane brzmienia klarnetu Mikołaja Trzaski robią wrażenie (brawa za pieczołowicie oddany świat epoki), ale ilość pojawiających się i znikających bohaterów na dalszym planie i brak większej głębi czy relacji między nimi nie pozwala widzowi jeszcze mocniej przejąć się ich losami..

Rysunek 5-14 Przykładowa gotowa do oceny recenzja zbudowana poprzez rozmowę z chatbotem

5.3. Algorytm oceny emocji w wypowiedzi

Algorytm oceniający nastawienie emocjonalne w wypowiedzi jest drugim z najważniejszych algorytmów, które zostały zaimplementowane w aplikacji. Jego zadaniem jest wystawienie oceny filmu w skali od 0 do 10, poprzez przeanalizowanie wyrażen pod kątem zabarwienia emocjonalnego w wypowiedzi użytkownika. Skuteczność poprawności analizy wzrasta wraz z długością ocenianej wypowiedzi, ponieważ czym dłuższa wypowiedź zostaje poddana analizie tym więcej znajduje się w niej wyrażen, które można poddać ocenie, co znacząco wpływa na wystawioną notę końcową. Najważniejsze

kroki algorytmu zostały szczegółowo opisane w poniższych podrozdziałach, według kolejności ich wykonywania.

5.3.1. Usunięcie elementów niemożliwych do poddania ocenie

Pierwszym krokiem algorytmu jest usunięcie z wypowiedzi elementów, których nie można poddać ocenie. Do takich elementów należą cyfry oraz znaki specjalne. Jedynymi wyjątkami są kropki, przecinki, wykrzykniki oraz wykrzykniki które służą do sygnalizowania końca zdania oraz rozdzielania jego składowych. Te znaki specjalne zostają wykorzystane w kolejnych krokach oceny wydźwięku do podzielenia wypowiedzi na mniejsze części. Usunięcie niepotrzebnych nam znaków zostało zaimplementowane z wykorzystaniem wyrażeń regularnych. Zostało to pokazane na rysunku 5-15.

[illegible]

Rysunek 5-15 Usuwanie znaków specjalnych oraz cyfr

5.3.2. Podział na listę zdań i słów

Kolejnym krokiem algorytmu, który służy przygotowaniu tekstu pod stworzenie listy wyrażen jest podzielenie go najpierw na zdania, a następnie podzielenie każdego zdania na listę pojedynczych wyrazów. Do podzielenia wypowiedzi na zdania wykorzystujemy pozostawione w poprzednik kroku znaki kończące zdania. Zamieniamy je na znak specjalny '#', który wykorzystamy jako wyrażenie w funkcji *split* przy dzieleniu zdań na wyrazy. Dzięki podzieleniu tekstu na zdania unikniemy podczas dalszych kroków zbudowania wyrażen w których wyrazy nie należą do tego samego zdania, przez co mogą mieć inny kontekst. Z kolei wydzielenie poszczególnych wyrazów umożliwi nam swobodne iterowanie po nich ułatwiając tworzenie listy wyrażen do oceny wydźwięku emocjonalnego. Na rysunku 5-16 ukazany został kod odpowiedzialny za wydzielenie zdań oraz wyrazów.

```

private ArrayList<String[]> separationWordInSentences(String text) {
    text = separationSentences(text);
    var wordList = new ArrayList<String[]>();

    String[] sentences = text.split( regex: "#");
    for (String sentence : sentences) {
        String[] words = sentence.trim().split( regex: " ");
        if (words.length > 3) {
            wordList.add(words);
        }
    }
    return wordList;
}

private String separationSentences(String text) {
    text = deleteSpecialCharsAndDigits(text);
    text = text.replaceAll( regex: "[\\.,\\!\\?]", replacement: " #");
    text = text.trim().replaceAll( regex: "+", replacement: " ");
    return text;
}

```

Rysunek 5-16 Podzielenie wypowiedzi na zdania oraz wyrazy

5.3.3. Stemming wyrazów

Kolejnym krokiem algorytmu jest wykonanie stemmingu na wydzielonych wcześniej wyrazach. Stemming jest to proces który usuwa końcówkę fleksyjną z danego wyrazu, przez co sprowadza go do formy podstawowej, którą możemy spotkać w słowniku. Dokonanie tej operacji umożliwia nam klasa PolishStemmer pochodząca z opisywanej we wcześniejszych rozdziałach biblioteki morfologik. Zaimplementowanie stemmingu przy wykorzystaniu wyżej opisanej klasy nie wymaga pisania dużej ilości kodu i jest możliwe do wykonania w kilku liniach. Zostało to przedstawione na rysunku 5-17.

```

private ArrayList stemming(ArrayList<String> words) {
    ArrayList<HashMap<String, String>> result = new ArrayList<>();
    for (String word : words) {
        var stemmingWord = stem(word);
        if (stemmingWord.length >= 2) {
            HashMap<String, String> stem = new HashMap<>();
            stem.put("word", stemmingWord[0]);
            stem.put("tag", tagFormat(stemmingWord[1]));
            result.add(stem);
        }
    }
    return result;
}

private String[] stem(String word) {
    IStemmer s = new PolishStemmer();
    ArrayList<String> result = new ArrayList<>();
    for (WordData wd : s.lookup(word)) {
        result.add(wd.getStem().toString());
        result.add(wd.getTag().toString());
    }
    return result.toArray(new String[result.size()]);
}

```

Rysunek 5-17 Stemming oraz określenie części mowy

Dodatkową zaletą wykorzystania powyższego rozwiązania jest jednocześnie przypisanie każdemu wyrazowi, który został poddany stemmingowi części mowy, do której on należy. Zestaw znaczników przypisywanych przez bibliotekę morfologik jest bardzo rozbudowany. Został on ukazany na rysunku 5-18. W dalszej części algorytmu będą nas interesowały tylko słowa, którym zostały przypisane podstawowe części mowy jak rzeczownik, przymiotnik, przysłówki czy czasownik.

Zestaw znaczników jest zbliżony do zestawu korpusu NKJP (www.nkjp.pl).

```
* adj - przymiotnik (np. „niemiecki”)
* adja - przymiotnik przyprzymiotnikowy (np. „niemiecko”, w wyrażeniach typu „niemiecko-chiński”)
* adjc - przymiotnik predykatywny (np. „ciekaw”, „dłużen”)
* adjp - przymiotnik poprzyimkowy (np. „niemiecku”)
* adv - przysłówki (np. „głupio”)
* burk - burkinostka (np. „Burkina Faso”)
* depr - forma deprecjatywna
* ger - rzeczownik odsłowny
* conj - spójnik łączący zdania współrzędne
* comp - spójnik wprowadzający zdanie podrzędne
* num - liczebnik
* pact - imiesłów przymiotnikowy czynny
* pant - imiesłów przysłówkowy uprzedni
* pcon - imiesłów przysłówkowy współczesny
* ppas - imiesłów przymiotnikowy bierny
* pprom12 - zaimek nielrzedzioosobowy
* pprom3 - zaimek lrrzedzioosobowy
* pred - predykatyw (np. „trzeba”)
* prep - przyimek
* siebie - zaimek "siebie"
* subst - rzeczownik
* verb - czasownik
* brev - skrót
* interj - wykrzyknienie
* qub - kublik (np. „nie” lub „tak”)
```

Rysunek 5-18 Oznaczenia części mowy przypisywanych przez bibliotekę morfologik

Źródło: www.github.com/morfologik/

5.3.4. Stworzenie listy wyrażeń

Stworzenie listy wyrażeń jest kolejnym krokiem w algorytmie oceniającym wydźwięk emocjonalny tekstu. Odbywa się ono poprzez znalezienie w przygotowanych listach, które zawierają wyrazy oraz odpowiadające im części mowy, wzorców wyrażeń których poszukujemy. Każde wyrażenie zawiera dwa następujące po sobie w tekście słowa, które należą do odpowiednich części mowy. W aplikacji zostało zaimplementowane pięć podstawowych wzorców które najczęściej tworzą wyrażenia o zabarwieniu emocjonalnym [14]. Są to:

- przymiotnik wraz z następującym po nim rzeczownikiem
- przysłówki przed przymiotnikiem, po którym nie ma rzeczownika

- dwa przymiotniki, po których nie występuję rzeczownik
- rzeczownik przed przymiotnikiem, po którym nie ma innego rzeczownika
- przysłówki przed czasownikiem

Na rysunku 5-19 przedstawiona została metoda, która odpowiada za wyszukiwanie wyżej opisanych wzorców na przygotowanych w poprzednich krokach algorytmu listach.

```
private void checkPhrase(ArrayList<String> tags, ArrayList<String> words) {
    // First ADJ, SUBST/GER, anything
    for (int i = 0; i < tags.size() - 1; ++i) {
        if ("adj".equals(tags.get(i).substring(0, 3)) && "subst".equals(tags.get(i + 1))) {
            phraseList.add(words.get(i) + " " + words.get(i + 1));
        } else if ("adj".equals(tags.get(i).substring(0, 3)) && "ger".equals(tags.get(i + 1))) {
            phraseList.add(words.get(i) + " " + words.get(i + 1));
        }
    }
    // First ADV, ADJ, NOT SUBST/GER
    for (int i = 0; i < tags.size() - 2; ++i) {
        if ("adv".equals(tags.get(i)) && "adj".equals(tags.get(i + 1).substring(0, 3))
            && (!("subst".equals(tags.get(i + 2))) && !("ger".equals(tags.get(i + 2))))) {
            phraseList.add(words.get(i) + " " + words.get(i + 1));
        }
    }
    // First ADJ, ADJ, NOT SUBST/GER
    for (int i = 0; i < tags.size() - 2; ++i) {
        if ("adj".equals(tags.get(i).substring(0, 3)) && "adj".equals(tags.get(i + 1).substring(0, 3))
            && (!("subst".equals(tags.get(i + 2))) && !("ger".equals(tags.get(i + 2))))) {
            phraseList.add(words.get(i) + " " + words.get(i + 1));
        }
    }
    // First SUBST/GER, ADJ, NOT SUBST/GER
    for (int i = 0; i < tags.size() - 2; ++i) {
        if (("subst".equals(tags.get(i)) || "ger".equals(tags.get(i))) && "adj".equals(tags.get(i + 1).substring(0, 3))
            && (!("subst".equals(tags.get(i + 2))) && !("ger".equals(tags.get(i + 2))))) {
            phraseList.add(words.get(i) + " " + words.get(i + 1));
        }
    }
    // First ADV, VERB
    for (int i = 0; i < tags.size() - 1; ++i) {
        if ("adv".equals(tags.get(i)) && "verb".equals(tags.get(i + 1))) {
            phraseList.add(words.get(i) + " " + words.get(i + 1));
        }
    }
}
```

Rysunek 5-19 Algorytm odpowiadający za wyszukiwanie wyrażeń w tekście

5.3.5. Ocena anotacji emocjonalnej wyrażeń

Kiedy lista wyrażeń zostanie utworzona kolejnym krokiem jest ocenienie każdego jej elementu pod względem wydźwięku emocjonalnego. Na początku każde z dwóch słów w wyrażeniu zostaje ocenione indywidualnie, a następnie zostaje wystawiona jego ocena końcowa. Przy wystawianiu noty finalnej jest wyliczana suma z ocen pojedynczych wyrazów. Jeżeli słowa w wyrażeniu mają przeciwny wydźwięk emocjonalny zostają one uznane jako niemożliwe do jednoznacznego sklasyfikowania i jego ocena ustawiana jest na neutralną.

Do oceny słów został wykorzystany opisywany we wcześniejszych rozdziałach słownik anotacji emocjonalnej. Dane zapisane w nim są w formacie csv, zgodnie ze

standardem RFC 4180. Przed dokonaniem pierwszej oceny następuje wczytanie wszystkich wierszy ze słownika do struktury danych którą jest HashMapa. Fragment kodu odpowiedzialny za wykonanie tej operacji został przedstawiony na rysunku 5-20.

```
private void doEmotionsMap() throws IOException {
    Reader in = new FileReader( fileName: "src/main/resources/slownik_anotacji_emocjonalnej.csv");
    Iterable<CSVRecord> records = CSVFormat.RFC4180.withFirstRecordAsHeader().parse(in);

    for (CSVRecord record : records) {
        emotions.put(record.get("lemat"), record.get("stopien_nacechowania"));
    }
}
```

Rysunek 5-20 Wczytywanie danych ze słownika

Lista wyrażzeń znalezionych w wypowiedzi wraz ze szczegółową oceną zostają przedstawiona użytkownikowi na stronie podsumowania. Zawiera ona wyrażenie z wyrazów sprowadzonych do podstawowej formy, ocenę pierwszego i drugiego wyrazu, oraz wystawioną notę finalną. Na rysunku 5-21 przedstawiony został fragment tabeli zawierającej znalezione w recenzji wyrażenia.

Wyrażenie	Ocena 1 wyrazu	Ocena 2 wyrazu	Ocena całego wyrażenia
zabawa weselny	-1	0	-1
sam wóz	0	0	Neutralne
wóz ślubny	0	0	Neutralne
ten moment	0	0	Neutralne
mistyczny droga	0	1	1
zupełnie inny	0	0	Neutralne
wołyński wieś	0	0	Neutralne
tamtejszy atmosfera	0	0	Neutralne
perfekcyjny scenografia	0	0	Neutralne
oddany świat	2	0	2
daleki plan	1	0	1

Rysunek 5-21 Tabela przedstawiająca ocenę przykładowych wyrażzeń

5.3.6. Wystawienie oceny końcowej

Wystawienie oceny końcowej jest rzeczą najbardziej oczekiwaną przez użytkownika aplikacji. Jest ono zwieńczeniem całego algorytmu oceniającego wydźwięk emocjonalny

recenzji użytkownika. Odbywa się poprzez wyliczenie średniej oceny z sum punktów jakie otrzymały wyrażenia w poprzednim kroku algorytmu. Nota końcowa mieści się w przedziale od 0 do 10 z dokładnością do części dziesiętnych. Na rysunku 5-22 zostało pokazane wyliczanie wyników końcowego, wraz z wykorzystaniem funkcji służącej do odpowiedniego zaokrąglenia wyniku.

```
double result = (positivePoints / (negativePoints * -1 + positivePoints));  
return Math.round(result * 100.00) / 100.00 * 10.00;
```

Rysunek 5-22 Obliczenie oceny końcowej wydźwięku recenzji

6. Podsumowanie

Celem niniejszej pracy było zaimplementowanie aplikacji chatbot wraz z analizowaniem wydźwięku emocjonalnego w wysyłanych wiadomościach. Temat pracy okazał się bardzo interesujący i był prawdziwym programistycznym wyzwaniem.

Chatboty są bardzo prężnie rozwijającym się sektorem rynku nowych technologii. Z każdym dniem wizjonerzy znajdują nowe miejsca, gdzie można je zaimplementować. Na dzień dzisiejszy chatboty są już w stanie zastąpić w niektórych zadaniach pracę człowieka. Rozwój sztucznej inteligencji doprowadzi do tego, że któregoś dnia nie będziemy w stanie rozróżniać czy za rozmówcę mamy drugiego człowieka, czy maszynę. Aby to było możliwe znacznej poprawie musi ulec rozpoznawanie emocji przez chatbota. Maszyny, które potrafią dostosowywać swoje wypowiedzi do nastawienia emocjonalnego rozmawiającego z nimi człowieka są na razie tylko wizją odległej przyszłości.

Rozpoznawanie emocji jest typowo ludzkim zjawiskiem. Zależy ono nie tylko od budowy gramatycznej wypowiedzi, ale także kontekstu, znajomości świata, ostatnich wydarzeń oraz wielu innych czynników składowych. Wszystkie te czynniki nie pozwalają na jednoznaczne zalgorytmizowanie wypowiedzi człowieka. Na chwilę obecną nikomu nie udało się stworzyć algorytmu potrafiącego rozpoznawać emocje w tekście pisanym w sposób, który można by uznać za bardzo satysfakcjonujący. Dostępne rozwiązania, oraz rozwiązanie zaimplementowane na potrzeby wykonania niniejszej pracy starają się rozwiązywać ten problem. Niestety nie robią tego w sposób doskonały. Wymagają one jeszcze wielu prób, oraz innych spojrzeń na ten problem, aby można uznać, że potrafią rozpoznawać emocje przynajmniej na poziomie człowieka.

Główną trudnością w realizacji projektu było odnalezienie oraz odpowiednie dobranie wykorzystanych w procesie tworzenia aplikacji narzędzi. Problem ten wynika z aktualnego niedoboru rozwiązań umożliwiających przetwarzanie języka polskiego. Jedyne znalezione rozwiązanie oferuje tę możliwość w bardzo wąskiej, okrojonej wersji w porównaniu np. do języka angielskiego. Dlatego niemożliwym było bezpośrednie wykorzystanie tego narzędzia do analizowania wydźwięku wiadomości. Doprowadziło to do konieczności zaimplementowania własnego algorytmu służącego do analizowania emocji w tekście pisanym. Rozwiązanie to mimo kilku swoich niedoskonałości wynikających m.in. z braku możliwości jednoznacznego określenia wydźwięku dla słów, które mają wiele znaczeń, spełnia swoje zadanie w sposób bardzo dobry.

Zaproponowane rozwiązanie problemu mogłoby zostać wykorzystane na stronach zawierających bazy filmów i seriali. Po dokonaniu niewielkich modyfikacji oraz wytrenowaniu modelu rozpoznającego inne, oczekiwane przez nas encję można go również użyć do innych celów, jak np. badanie nastawienia do tematów publicznych, przeprowadzanie ankiet czy opiniowania dokonań rządu oraz innych organów i osób publicznych.

Kod aplikacji został udostępniony na licencji GNU General Public License w wersji 3 na serwisie Github i jest dostępny pod adresem <https://github.com/Decooo/Chatbot>. Wybrana licencja ma na celu zagwarantowanie użytkownikom swobody do udostępniania i zmieniania kodu czyli dania pewności, że oprogramowanie jest wolno dostępne dla każdego użytkownika [15].

Autor za wkład własny uważa wykonanie serwisu webowego umożliwiającego użytkownikowi ocenę wybranego filmu poprzez przeprowadzenie konwersacji z chatbotem oraz zaimplementowanie algorytmów przeprowadzania dialogu i analizowania wydźwięku emocjonalnego w wypowiedziach.

Literatura

- [1] Psychologia wyglądu, <http://www.psychologiawygladu.pl/2015/07/jak-czytac-emocje-siedem-podstawowych.html>, dostęp: styczeń 2019
- [2] Słownik Słowosieć, <http://plwordnet.pwr.wroc.pl/wordnet/about>, dostęp: styczeń 2019
- [3] Stemming, <https://xapian.org/docs/stemming.html>, dostęp: styczeń 2019
- [4] Inżynieria wiedzy, <http://www.inzynieriawiedzy.pl/systemy-inteligentne/programy-konwersacyjne/istota-historia-ewolucja>, dostęp: styczeń 2019
- [5] Test Turinga, https://pl.wikipedia.org/wiki/Test_Turinga, dostęp: styczeń 2019
- [6] Łukasz Lenart – „W świecie botów czyli po co nam SI”, prezentacja na konferencji Confitura 2018, <https://www.youtube.com/watch?v=TOG85kssyqg>
- [7] Dokumentacja platformy wit.ai, <https://wit.ai/docs>, dostęp: styczeń 2019
- [8] Richard M. Reese: Natural Language Processing with Java, Packt, 2015
- [9] Emocje podstawowe, <http://wiecjestem.us.edu.pl/category/tagi/emocje-podstawowe>,
dostęp: styczeń 2019
- [10] Cay S. Horstmann, Gary Cornell: Java. Podstawy. Wydanie IX., Helion, 2015
- [11] Dokumentacja Javy, <https://docs.oracle.com/javase/10/>, dostęp: styczeń 2019
- [12] Rod Johnson.: Spring framework : profesjonalne tworzenie oprogramowania w Javie, Helion, Gliwice. 2006
- [13] Dokumentacja Spring Boota, <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>, dostęp: styczeń 2019
- [14] Peter D. Turney: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, Philadelphia, 2002,
<http://www.aclweb.org/anthology/P02-1053.pdf>
- [15] Licencja GNU, <https://www.gnu.org/licenses/gpl-3.0.html>, dostęp: styczeń 2019