UNIWERSYTET EKONOMICZNY W KATOWICACH

KIERUNEK INFORMATYKA I EKONOMETRIA

**TOMASZ CHMIEL**

**145019**

**Opracowanie chatbota odpowiadającego na pytania dotyczące przepisów gry w piłkę nożną.**

The development of a chatbot answering questions about football game laws.

Praca licencjacka napisana

pod kierunkiem dr inż. Grzegorza Dziczkowskiego

KATOWICE 2023

**OŚWIADCZENIE PROMOTORA**

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia wymogi stawiane pracom dyplomowym.

Pracę akceptuję

…………………………………… ……………………………………

(data) (podpis promotora)

Tomasz Chmiel

*Imię i nazwisko* Katowice, dnia ...............................

Informatyka i ekonometria *Kierunek*

145019 *Nr albumu*

**OŚWIADCZENIE**

Mając świadomość odpowiedzialności prawnej oświadczam, że złożona praca licencjacka/inżynierska/magisterska pt.: „Opracowanie chatbota odpowiadającego na pytania dotyczące przepisów gry w piłkę nożną.” została napisana przeze mnie samodzielnie. Równocześnie oświadczam, że praca ta nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych oraz dóbr osobistych chronionych prawem. Ponadto praca nie zawiera informacji i danych uzyskanych w sposób niedozwolony i nie była wcześniej przedmiotem innych procedur związanych z uzyskaniem dyplomów lub tytułów zawodowych uczelni wyższej. Wyrażam zgodę na nieodpłatne udostępnienie mojej pracy w celu oceny jej oryginalności przez Jednolity System Antyplagiatowy prowadzony przez ministra właściwego ds. szkolnictwa wyższego oraz przechowywania jej w Ogólnopolskim Repozytorium Prac Dyplomowych oraz wewnętrznej bazie prac dyplomowych Uniwersytetu Ekonomicznego w Katowicach. Oświadczam, że poinformowano mnie o zasadach dotyczących oceny oryginalności pracy dyplomowej przez Jednolity System Antyplagiatowy. Oświadczam także, że ostateczna wersja pracy przesłana przeze mnie drogą elektroniczną jest zgodna z plikiem poddanym ocenie w Jednolitym Systemie Antyplagiatowym. Jednocześnie oświadczam, że jest mi znany przepis art. 233 § 1 Kodeksu karnego określający odpowiedzialność za składanie fałszywych zeznań.

......................................................

(podpis składającego oświadczenie)

**SPIS TREŚCI**

[WSTĘP 5](#_Toc137625386)

[1. Aktualny stan wiedzy 7](#_Toc137625387)

[1.1. Chatboty jako narzędzie komunikacji 7](#_Toc137625388)

[1.2. Rozwój i znaczenie przepisów gry w piłkę nożną 15](#_Toc137625389)

[2. Podstawowe cele i funkcjonalności chatbota ’Refi’ 19](#_Toc137625390)

[2.1. Wstęp 19](#_Toc137625391)

[2.2. Architektura chatbota 19](#_Toc137625392)

[2.3. Cel pracy 20](#_Toc137625393)

[2.4. Wymagania funkcjonalne 20](#_Toc137625394)

[2.5. Wymagania pozafunkcjonalne 21](#_Toc137625395)

[3. Wykorzystane narzędzia i technologie 22](#_Toc137625396)

[3.1. Microsoft Visual Studio 22](#_Toc137625397)

[3.2. C# 22](#_Toc137625398)

[3.3. Windows Forms 23](#_Toc137625399)

[3.4. Pandorabot 24](#_Toc137625400)

[3.5. AIML 24](#_Toc137625401)

[3.6. AIMLBot 25](#_Toc137625402)

[3.7. GIMP 26](#_Toc137625403)

[4. Implementacja 27](#_Toc137625404)

[4.1. Struktura projektu 27](#_Toc137625405)

[4.2. Baza wiedzy 29](#_Toc137625406)

[4.3. Graficzny interfejs użytkownika 32](#_Toc137625407)

[4.4. Generowanie odpowiedzi przez chatbota 35](#_Toc137625408)

[5. Testy i analiza wyników 37](#_Toc137625409)

[5.1. Testowanie logowania 37](#_Toc137625410)

[5.2. Testowanie poprawności generowania odpowiedzi 39](#_Toc137625411)

[5.3. Testowanie wydajności 40](#_Toc137625412)

[5.4. Testowanie użyteczności 41](#_Toc137625413)

[5.5. Wnioski z testów 43](#_Toc137625414)

[ZAKOŃCZENIE 44](#_Toc137625415)

[LITERATURA 45](#_Toc137625416)

[SPIS RYSUNKÓW I TABEL 47](#_Toc137625417)

WSTĘP

Ostatnimi czasy sztuczna inteligencja stanowi najbardziej pasjonujący i dynamicznie rozwijający się obszar w dziedzinie informatyki. Zdolność sztucznej inteligencji do analizowania danych, uczenia się, podejmowania decyzji, a nawet interakcji z ludźmi, otwiera szereg nowych możliwości w najróżniejszych sferach życia. Postęp sztucznej inteligencji niesie za sobą mnogość szans i perspektyw, ale jednocześnie rodzi wiele kontrowersji dotyczących wpływu sztucznej inteligencji na społeczeństwo. Kwestie związane z etyką, prywatnością danych, prawami autorskimi czy ogólnie pojętą odpowiedzialnością, stanowią nieodzowny element dyskusji o sztucznej inteligencji. Tworzenie programów konwersacyjnych, nazywanych chatbotami, jest jednym z obszarów, w którym sztuczna inteligencja znajduje bardzo szerokie zastosowanie i w którym może wykorzystać swój olbrzymi potencjał. Chatboty to programy komputerowe, które są zaprojektowane do prowadzenia rozmowy z człowiekiem, w sposób przypominający międzyludzką komunikację interpersonalną.

Teoretycznym celem niniejszej pracy jest przedstawienie najważniejszych kwestii związanych z chatbotami oraz przepisami gry w piłkę nożną. Celem praktycznym jest opracowanie programu konwersacyjnego odpowiadającego na pytania dotyczące przepisów gry w piłkę nożną. Piłka nożna, będąca najpopularniejszym sportem na świecie, posiada złożone oraz momentami bardzo skomplikowane zasady, które mogą być niejasne oraz trudne do zrozumienia dla części szerokiego grona entuzjastów tej dyscypliny sportu. Chatbot, który zostanie stworzony przez autora, zagwarantuje nieograniczony dostęp do szybkiej i wartościowej pomocy w kwestiach dotyczących przepisów gry w piłkę nożną. Opracowanie programu konwersacyjnego, który prawidłowo odpowiada na pytania dotyczące przepisów gry, stanowi wyzwanie nie tylko techniczne, lecz również społeczne. Jeśli chatbot udzielałby błędnych lub nieprecyzyjnych odpowiedzi, istniałoby ryzyko podważenia autorytetu sędziów piłki nożnej oraz utraty wiarygodności podejmowanych przez nich decyzji.

Praca składa się z pięciu rozdziałów, które w sposób uporządkowany opisują różne aspekty dotyczące chatbota, będącego tematem pracy autora. Na początku zaprezentowano aktualny stan wiedzy odnoszący się do programów konwersacyjnych oraz przepisów gry w piłkę nożną. Następnie skupiono się na prezentacji podstawowych celów chatbota, jak również jego wymagań, z podziałem na funkcjonalne i pozafunkcjonalne. W dalszym ciągu szczegółowo przedstawiono architekturę programu konwersacyjnego. Kolejno skoncentrowano się na narzędziach oraz technologiach wykorzystanych do stworzenia chatbota. Po raporcie technicznym wnikliwie opisano implementację: strukturę projektu, bazę wiedzy, graficzny interfejs użytkownika, jak również proces generowania odpowiedzi przez chatbota. Na końcu szczególną uwagę skierowano na przeprowadzone testy i analizę uzyskanych wyników.

1. Aktualny stan wiedzy

* 1. Chatboty jako narzędzie komunikacji

Definiując w sposób przystępny, chatbot to rozwinięty program komputerowy, który  może prowadzić rozmowy z człowiekiem. Chcąc z niego skorzystać, użytkownik zadaje  pytanie lub formułuje zdanie twierdzące, a zadaniem programu konwersacyjnego jest udzielić sensownej  odpowiedzi w sposób jak najbardziej zbliżony do ludzkiego. W niniejszym podrozdziale przedstawiono temat chatbotów jako narzędzia skutecznej komunikacji. Na początku przedstawiono czynniki, które przyczyniły się do ogromnego zainteresowania programami konwersacyjnymi w dzisiejszych czasach. Następnie zaprezentowano chronologiczny przegląd różnych programów konwersacyjnych, począwszy od pionierów w tej dziedzinie, takich jak ELIZA i PARRY, a kończąc na bardziej nowoczesnych modelach, w tym Mitsuku oraz ChatGPT. Kolejno omówiono dwa sposoby działania chatbotów: oparte na regułach i oparte na sztucznej inteligencji. Na końcu przedstawiono ograniczenia i trudności, z jakimi programy konwersacyjne mierzą się w trakcie rozmowy z człowiekiem. Wśród tych trudności można wymienić między innymi błędne interpretacje kontekstu wypowiedzi czy powtarzalność udzielanych odpowiedzi.

* + 1. Wprowadzenie

Niezwykle dynamiczny rozwój technologii doprowadził do momentu, w którym wykorzystanie sztucznej inteligencji w różnych obszarach życia człowieka, to już nie utopijna wizja przyszłości, a codzienność[1]. Oczywistą tego konsekwencją stała się olbrzymia potrzeba na rozwiązania zawierające elementy sztucznej inteligencji, będące bezpieczne oraz intuicyjne także dla osób nie obytych z nowoczesnymi technologiami. W tej sytuacji doskonale odnalazł się rynek chatbotów. Obecnie są one wykorzystywane w wielu różnych branżach. Chatboty potrafią prowadzić rozmowy z klientami, odpowiadać na ich pytania, a także udzielać informacji na temat produktów i usług. Programy konwersacyjne bardzo dobrze zbierają informacje od użytkowników z którymi rozmawiają. Dzięki temu, mogą pozyskiwać cenne dane o preferencjach, potrzebach oraz oczekiwaniach użytkowników, co stanowi nieocenioną wartość dla organizacji.

1.1.2. Chronologiczny przegląd chatbotów

Historia programów konwersacyjnych sięga lat sześćdziesiątych dwudziestego wieku i jest efektem długoletnich eksperymentów i badań naukowych nad sztuczną inteligencją oraz interakcjami człowieka z komputerem. Przez kolejne dekady rozwijano różne formy programów konwersacyjnych, począwszy od bardzo prostych systemów opartych na regułach i algorytmach, aż po znacznie bardziej zaawansowane modele oparte na sztucznej inteligencji i uczeniu maszynowym. Grafika[Rysunek 1] przedstawia oś czasu z wyszczególnionymi datami utworzenia najważniejszych chatbotów.

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Rysunek 1. Oś czasu z datami utworzenia najważniejszych chatbotów. Źródło: Opracowanie własne

1.1.1.1. ELIZA

ELIZA to program symulujący psychoanalityka stworzony przez Josepha Weizenbauma w 1966 roku. Chatbot ELIZA jest uznawany za pierwszego chatbota w historii informatyki, mimo że termin chatbot jeszcze w tamtym okresie nie istniał, bo po raz pierwszy został użyty przez Michael Mauldiniego w 1994 roku. Chatbot ELIZA działa na zasadzie rozpoznawania słów kluczowych lub fraz z danych wejściowych, a następnie wyświetla wcześniej stworzone odpowiedzi pasujące do wspomnianych słów kluczowych[2]. Na przykład, jeśli człowiek powiedział: „Bardzo obawiam się jutrzejszego egzaminu”. ELIZA wybierała słowo „obawiam” i odpowiadała, „Opowiedz coś więcej o swoich lękach.” To bardzo imponowało jej pierwszym użytkownikom, gdyż stwarzało iluzję zrozumienia i interakcji z prawdziwym człowiekiem, chociaż proces był całkowicie zmechanizowany. Pomimo prostoty i bazy wiedzy nie opierającej się na żadnej wiedzy psychiatrycznej, pierwsi testerzy programu konwersacyjnego byli zachwyceni jego działaniem. Wydawało im się, że rozmawiają z innym człowiekiem, więc chętnie opowiadali ELIZIE o swoich problemach i otwierali się emocjonalnie przed programem konwersacyjnym. Eksperci deklarowali wtedy, że za kilka lat chatboty będą nie do odróżnienia od ludzi. Dzisiaj wiemy, że ich przewidywania były błędne.

1.1.1.2. PARRY

W 1972 roku, kiedy Atari wydawało „Pong”, pierwszą grę wideo do użytku domowego, w Kaliforni psychiatra Kenneth Colby skończył programować chatbota PARRY, stworzonego w celu jak najdokładniejszej symulacji wzorców myślenia paranoidalnego schizofrenika. Program potrafił reagować na sygnały emocjonalne i angażował się w rozmowę na szeroki zakres tematów. Jednak chatbot bardzo często błędnie interpretował wypowiedzi użytkowników i prawie zawsze zakładał, że osoba, z którą rozmawia, ma złe zamiary oraz mówi nieprawdę[3]. Mimo to PARRY był pierwszym chatbotem, który pomyślnie przeszedł specjalną wersję testu Turinga. Chatbot Colbiego przyczynił się także do rozpoczęcia ważnej dyskusji odnośnie aspektów etycznych sztucznej inteligencji. Debaty w tej sprawie trwają do dnia dzisiejszego i wzbudzają różniące się emocje. Obecnie temat etycznych aspektów sztucznej inteligencji jest nie tylko modny ale przede wszystkim niezwykle ważny. Według badań, rozwój sztucznej inteligencji do 2030 roku, będzie wciąż koncentrował się przede wszystkim wokół maksymalizacji zysków, a etyczne tworzenie sztucznej inteligencji będzie nadal sprawą drugorzędną[4].

1.1.1.3. Jabberwacky

Jabberwacky to dzieło brytyjskiego programisty Rollo Carpentera, powstałe w 1981 roku. Zadaniem chatbota było odzwierciedlenie naturalnej ludzkiej rozmowy w zabawny sposób. Chatterbot znacząco różnił się od konkurencji, gdyż koncentrował się wyłącznie na rozrywce, co w tamtych czasach było dość nietypowe. W swoich konwersacjach bardzo często wprowadzał wątki humorystyczne, czym starał się rozbawić rozmówcę. Warto zaznaczyć, że w latach 2005 i 2006 Jabberwacky, a w zasadzie jego kolejno powstałe osobowości, wygrywały niezwykle prestiżową nagrodę Loebnera. Początkowym założeniem Hugho Loebnera, twórcy wspomnianej nagrody było nagrodzenie kwotą 100000 dolarów i 18-karatowym złotym medalem, pierwszego programisty, któremu uda się napisać program, potrafiący zaliczyć kompletny test Touringa[5]. Z biegiem lat założenia konkursu się zmieniały i obecnie nagrodą Loebnera są corocznie honorowane chatboty, których konwersacje najbardziej przypominają te prowadzone z człowiekiem.

1.1.1.4. A.L.I.C.E.

Chatbot ALICE jest uznawany za najbardziej wpływowy program konwersacyjny w historii ludzkości. Jego autor Richard Wallace, jest także twórcą języka programowania AIML. Wallace stworzył i wykorzystał AIML do reprezentowania wiedzy chatbota w formie, która jest zrozumiała dla komputera. Język programowania AIML charakteryzujący się prostotą oraz uniwersalnością, bardzo szybko stał się popularny wśród twórców chatbotów i jest do dziś używany w wielu aplikacjach związanych z przetwarzaniem języka naturalnego. Chatbot ALICE swoje pierwsze kroki stawiał w 1995 roku[6]. Prowadził rozmowy na różne tematy, będąc przy tym imponująco naturalny. W swoim działaniu starał się wykorzystywać modele matematyczne, heurystyki naturalnej konwersacji. Chatbot Wallace’a zdobył nagrodę Loebnera trzykrotnie, w 2000, 2001 i 2004 roku. W kręgach informatycznych uznaje się ALICE za najbardziej udany program konwersacyjny jaki kiedykolwiek powstał. Jego open source kod, przyczynił się do powstania tysięcy innych chatbotów, w tym wyjątkowo popularnego Mitsuku.

1.1.1.5. Mitsuku

Mitsuku jest programem konwersacyjnym, który został stworzony w 2005 roku, przez Steve’a Worswicka, cenionego programistę z Wielkiej Brytanii. Dzięki zaawansowanym algorytmom przetwarzania języka naturalnego (NLP), chatbot jest w stanie prowadzić realistyczne i responsywne rozmowy z użytkownikami. Kluczową cechą Mitsuku jest zdolność do zapamiętywania wcześniejszych rozmów oraz wykorzystywanie tych informacji do personalizowania konwersacji. W rezultacie rozmowa staje się znacznie bardziej spójna i angażująca. Chatbot cechuje się także wyjątkową wszechstronnością, co stanowi kolejną jego ważną zaletę. Możliwość integracji z różnymi platformami i usługami, sprawia, że Mitsuku jest chętnie wybieranym rozwiązaniem przez firmy i organizacje, które pragną wdrożyć chatboty w swojej obsłudze klienta lub strategiach marketingowych. Chatbot stworzony przez Steve’a Worswicka zwyciężył w konkursie Loebnera aż pięciokrotnie[7].

1.1.1.6. ChatGPT

ChatGPT to chatbot oparty na modelu językowym GPT-3 stworzonym przez OpenAI. Model językowy GPT-3 został opracowany dzięki wykorzystaniu różnych technik uczenia maszynowego, w tym uczenia przez wzmacnianie oraz nadzorowanego uczenia maszynowego. ChatGPT został wprowadzony na rynek w czerwcu 2020 roku i niezwykle szybko zyskał ogromną popularność. Chatbot zdobył ponad milion użytkowników w ciągu pięciu dni od momentu udostępnienia, co jest rekordowym wynikiem, w porównaniu z innymi popularnymi platformami społecznościowymi i strumieniowymi, takimi jak Netflix, Facebook czy Instagram. Przykładowo, Netflix potrzebował 41 miesięcy na zdobycie miliona użytkowników, Facebook 10 miesięcy, a Instagram 2,5 miesiąca[8]. Chatbot stworzony przez OpenAI, posiada zdolność prowadzenia rozmów na szeroką gamę tematów - od tych ogólnych, po te wymagające szczegółowej wiedzy specjalistycznej. Program konwersacyjny działa w czasie rzeczywistym i jest w stanie generować odpowiedzi na zadane pytania w ciągu zaledwie kilku sekund. Jedną z unikalnych cech ChatGPT jest jego zdolność do uczenia się z interakcji z użytkownikami. Oznacza to, że wraz z każdą kolejną rozmową, ChatGPT zdobywa nową wiedzę, analizuje ją i wykorzystuje do ulepszania swojego modelu językowego. W ten sposób ChatGPT ciągle się rozwija i doskonali swoje umiejętności, poprawiając tym samym efektywność komunikacji oraz zwiększając poziom satysfakcji klientów.

* + 1. Sposób działania chatbotów

Sposób działania chatbotów może się różnić w zależności od ich konkretnego zastosowania czy technologii, na której są oparte. Najstarsze programy konwersacyjne, powstałe w latach 60. i 70. XX wieku były bardzo proste w swoim działaniu. Programiści tych chatbotów musieli określić i zaprogramować każdą możliwą ścieżkę dialogową, co oznaczało, że chatbot miał określony zestaw odpowiedzi dla każdego pytania lub tematu, który mógł pojawić się w rozmowie. Współczesne chatboty korzystają z przetwarzania języka naturalnego (Natural Language Processing, NLP), który jest jednym z najszybciej rozwijających się sektorów w dziedzinie sztucznej inteligencji oraz lingwistyki. Wykorzystanie przetwarzania języka naturalnego oznacza, że programy konwersacyjne posiadają zdolności rozumienia i interpretacji języka stosowanego przez ludzi do komunikacji interpersonalnej[9]. Mówiąc dosłownie przetwarzanie języka naturalnego umożliwia interakcję między komputerem a językiem ludzkim. Rysunek[Rysunek 2] prezentuje ogólny sposób działania chatbotów, przedstawiony w mocno uproszczonej formie.

Obraz zawierający tekst, zrzut ekranu, Czcionka, Prostokąt

Opis wygenerowany automatycznie

Rysunek 2. Ogólny sposób działania chatbotów Źródło: Opracowanie własne przy pomocy[10]

Ze względu na sposób w jaki chatboty działają można wyróżnić ich dwa rodzaje:

**Chatbot blokowy** – jest oparty na regułach, przez co potrafi zrozumieć wyłącznie ograniczony zakres treści. W swoim działaniu używa bazy danych oraz zaprogramowanych scenariuszy odpowiedzi, do których przypisane są słowa kluczowe. Na przykład, jeśli podczas rozmowy z programem konwersacyjnym użytkownik użyje słowa ‘otwarcie’, to chatbot wyświetli mu pełną informację o godzinach otwarcia miejsca, którego dotyczy rozmowa.

**Chatbot konwersacyjny** – jest oparty na sztucznej inteligencji, dlatego potrafi zrozumieć nawet złożone zapytania. W swoim działaniu wykorzystują algorytmy uczenia maszynowego, jak również przetwarzanie języka naturalnego. To sprawia, że w trakcie rozmowy z użytkownikiem rozpoznaje kontekst wypowiedzi i potrafi trafnie zareagować. Chatbot konwersacyjny, stale poprawia swoje umiejętności, ponieważ posiada zdolność „uczenia się” na podstawie odbytych dialogów.

* + 1. Ograniczenia i problemy chatbotów

Mimo że chatboty z roku na rok stają się coraz „mądrzejsze”, to wciąż nie można ich określać mianem programów idealnych. Trudności i wyzwania, z którymi zmagali się programiści ponad pięćdziesiąt lat temu, są w dużej mierze nadal aktualne, a współcześni programiści wciąż szukają dla nich optymalnych rozwiązań. Nawet „najinteligentniejsze” chatboty dwudziestego pierwszego wieku generują różnego rodzaju problemy oraz posiadają ograniczenia, które zostanę opisane w niniejszym podrozdziale.

1.1.3.1. Powtarzalność odpowiedzi

Jednym z problemów chatbotów jest ich tendencja do powtarzania tych samych odpowiedzi. Szczególnie często dzieje się tak w przypadku programów konwersacyjnych opartych na regułach, które nie wykorzystują algorytmów uczenia maszynowego. Zwracanie tych samych treści jest spowodowane ograniczoną bazą wiedzy. Konkretyzując, programy konwersacyjne nie potrafią odpowiedzieć na zapytania, których ich nie nauczono. Dlatego w sytuacji, kiedy treść zapytania jest dla nich całkowicie nowa, odpowiadają tym samym czym odpowiedziały na wcześniejsze, zbliżone tematycznie pytanie[11]. To nierzadko powoduje irytację użytkownika, który nieusatysfakcjonowany otrzymaną odpowiedzią kończy konwersację z programem.

1.1.3.2. Brak zrozumienia kontekstu

Znajomość oraz właściwa interpretacja kontekstu jest niezwykle istotna dla prawidłowego zrozumienia wiadomości. Kontekst obejmuje między innymi czas, miejsce, sytuację, czy relację między ludźmi. Wymienione elementy nie rzadko wpływają na to, jak informacja jest postrzegana i odbierana przez rozmówcę. Chatboty nie potrafią kontekstowo interpretować języka naturalnego oraz brakuje im zdolności poznawczych ludzi. Dlatego często odpowiadają nieadekwatnie lub niezrozumiale, w sytuacjach, w których ów kontekst ma duże znaczenie[12]. Załóżmy, że użytkownik wprowadzi następującą wiadomość: „Miałem dziś fatalny dzień w pracy. Mój szef przerósł samego siebie”. W takiej sytuacji nawet najinteligentniejsze programy konwersacyjne, będą miały olbrzymie problemy z właściwą reakcją, gdyż nie rozumieją, że określenie „przeróść samego siebie” ma w powyższym przykładzie pejoratywne zabarwienie.

1.1.3.3. Potrzeba ciągłej aktualizacji

Programy konwersacyjne powinny być stale aktualizowane i uzupełniane o nowe informacje, aby móc trafnie odpowiadać na pytania użytkowników. To wymaga zaangażowania zespołu programistów i dostawców oprogramowania, którzy muszą regularnie poszerzać bazę wiedzy chatbotów. Jeśli by tego nie robili, konwersacje mogłyby stać się mocno problematyczne. Przykładowo baza wiedzy programu konwersacyjnego firmy zakładów bukmacherskich musi zostać zaktualizowana, gdy tylko bieżąca oferta lub pakiety promocyjne ulegną zmianie. Jeśli chatbot nie zostanie odpowiednio dostosowany do nowości, może dostarczać nieaktualne lub nawet nieprawdziwe informacje, co może prowadzić do nieporozumień i frustracji użytkowników.

1.1.3.4. Zagrożenia bezpieczeństwa

Zagrożenia bezpieczeństwa chatbotów mogą mieć różne formy, a zorientowanie się, że program konwersacyjny z którym prowadzona jest rozmowa został zaatakowany jest praktycznie niemożliwe. Istnieje wiele rodzajów ataków, które mogą być skierowane przeciwko chatbotom, takich jak:

* Ataki typu DDoS (Distributed Denial of Service), które polegają na zablokowaniu dostępu do chatbotów poprzez wysyłanie dużej liczby żądań do ich serwerów, co powoduje wyczerpanie zasobów obliczeniowych programu konwersacyjnego[13]. Tego typu atak można trafnie przyrównać do sytuacji, w której korkuje się ulica ze względu na nadmiar pojazdów, które się nią poruszają.
* Ataki hakerskie, które polegają na włamaniu się do systemu chatbotów, a następnie modyfikowaniu ich oprogramowania lub bazy danych.
* Ataki związane z ludzkim błędem, takie jak ujawnienie hasła lub innych wrażliwych danych o użytkowniku, przez osoby kontrolujące i modyfikujące funkcjonalności programów konwersacyjnych.

Aby zabezpieczyć chatboty przed wyżej wymienianymi atakami, konieczne jest wdrożenie odpowiednich środków bezpieczeństwa takich jak firewalle, czy szyfrowanie danych. Istotne jest również, aby osoby odpowiedzialne za poprawne działanie programów konwersacyjnych były świadome potencjalnych zagrożeń i umiały skutecznie reagować w przypadku ataków cybernetycznych.

* 1. Rozwój i znaczenie przepisów gry w piłkę nożną

W niniejszym podrozdziale przybliżony zostanie temat przepisów gry w piłkę nożną. Opracowane przez IFAB ‘Przepisy Gry 22/23’ to najnowsze wydanie książki zawierającej wszystkie zasady najpopularniejszego sportu na świecie, jakim jest piłka nożna. To właśnie ‘Przepisy Gry 22/23’ były jedynym źródłem informacji wykorzystanym do stworzenia obszernej bazy wiedzy chatbota ‘Refi’, będącego przedmiotem pracy dyplomowej autora. Na początku podrozdziału ‘Rozwój i znaczenie przepisów gry w piłkę nożną’ wprowadzono w fascynujący świat zasad boiskowych, zaznaczając co stanowi największą wartość w świecie futbolu. Następnie wyjaśniono, czym w ogóle są przepisy gry oraz przedstawiono ich dokładną zawartość. Kolejno zaprezentowano bardzo ciekawy proces ewolucji przepisów gry na przestrzeni lat. Ostatni akapit podrozdziału poświęcony został roli sędziego w piłce nożnej, która bezsprzecznie jest silnie związana z przepisami gry.

* + 1. Wprowadzenie

Edson Arantes do Nascimento, najwybitniejszy piłkarz w historii znany jako Pele, powiedział kiedyś: "Piłka nożna to nie tylko sport. To styl życia, to uczucie, to pasja." Ciężko nie zgodzić się z Brazylijczykiem, skoro szacuje się, że ponad 300 milionów ludzi na całym świecie gra regularnie w piłkę nożną, a ponad 4 miliardy ogląda mecze piłkarskie[14]. Żaden inny sport nie może pochwalić się tak olbrzymimi liczbami. Niewątpliwie futbol to gra globalna, wzbudzająca emocje na każdym kontynencie. Co istotne przepisy gry są jednakowe na całym świecie, począwszy od finału Ligi Mistrzów UEFA, a skończywszy na meczach dzieci w Afryce Południowej, czy oldbojów w Azji Wschodniej. Ta jednolitość jest uznawana za największą wartość futbolu, która dla dobra tej dyscypliny powinna być pielęgnowana.

* + 1. Czym są przepisy gry?

Przepisy gry w piłkę nożną to zbiór reguł określających sposób gry w piłkę nożną oraz zasad obowiązujących na boisku oraz w strefach technicznych podczas meczów. Są one ustanawiane i uchwalane przez Międzynarodową Radę Piłkarską (IFAB) i obowiązują na całym świecie[15]. Co roku udostępniana jest książka, zawierająca aktualne przepisy gry, obowiązujące w bieżącym sezonie rozgrywkowym. Główna jej część skupia się wokół najważniejszych zasad gry w piłkę nożną. Są one podzielone na siedemnaście artykułów, a każdy z nich jest bardzo szczegółowo omówiony. Rysunek[Rysunek 3] przedstawia najważniejszy fragment spisu treści przepisów gry w piłkę nożną.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 3. Spis artykułów znajdujących się w przepisach gry w piłkę nożną Źródło:[16]

Książka Przepisy Gry zawiera również dokładnie wytłumaczone zmiany w zasadach gry, które zostały wprowadzone w stosunku do roku poprzedniego. Na końcu znajduje się słowniczek terminologii piłkarskiej oraz zestaw praktycznych wskazówek dla sędziego, które stanowią uzupełnienie informacji zawartych w Przepisach Gry.

* + 1. Ewolucja przepisów gry

Kalendarium zmian w przepisach gry w piłkę nożną prezentuje się następująco:

1848– Osoby określające się ojcami nowej dyscypliny sportu, spotkały się na uniwersytecie w Cambridge, aby ustanowić jej przepisy[17]. Pierwszy oraz najważniejszy z nich głosił, że zabronione jest dotykanie piłki rękami w czasie gry. Niestety tej i wielu innych zasad nigdzie nie opublikowano, przez co wiele z nich zaginęło.

1863 - Powstaje związek piłkarski zwany Football Association (FA), który opracowuje pierwsze oficjalne przepisy gry w piłkę nożną. Przepisy te zostają opublikowane w formie kodeksu zwanego "Laws of the Game".

1904 - Powstaje Międzynarodowa Federacja Piłki Nożnej (FIFA), która zaczyna sprawować nadzór nad przepisami gry w piłkę nożną na arenie międzynarodowej. Obecnie FIFA skupia 211 związków piłkarskich z całego świata, jej najważniejszym zadaniem jest organizacja Mistrzostw Świata w piłce nożnej.

1938 - Przepisy gry w piłkę nożną zostają ujednolicone i uproszczone. Był to bardzo ważny moment dla rozwoju piłki nożnej, bo od tego momentu te same zasady zaczynają obowiązywać na całym świecie.

1970 – Wprowadzone zostały nowe przepisy dotyczące używania rąk przez bramkarzy podczas gry. Natomiast na mundialu w Meksyku po raz pierwszy do temperowania boiskowych zachowań piłkarzy użyto żółtych i czerwonych kartek.

2009 – Na zielonej murawie zadebiutowali sędziowie zabramkowi. Funkcja ta okazała się jedną z najbardziej nietrafionych i niepotrzebnych w futbolu na najwyższym poziomie.

2017 - Po raz pierwszy oficjalnie użyto technologii "VAR" (z ang. Video Assistant Referee), czyli asystenta wideo dla sędziów, który ma pomagać w podejmowaniu trudnych decyzji podczas meczu i zredukować do minimum ilość popełnianych błędów przez arbitrów.

* + 1. Rola sędziego w piłce nożnej

‘Refi’ to nazwa chatbota, będącego tematem pracy. Nazwa ta jest pochodną słowa „referee”, które w języku angielskim oznacza sędziego. Podczas meczu piłki nożnej, to właśnie sędzia jest niekwestionowanym decydentem, mającym olbrzymi wpływ na ostateczne rozstrzygnięcie zawodów. Można śmiało powiedzieć, że jest on najważniejszą osobą na boisku. Do sędziego należy obowiązek egzekwowania przepisów gry i podejmowania trudnych decyzji, dotyczących między innymi rzutów karnych, fauli, bramek, czy kar indywidualnych. Oczywiście przepisy gry nie są w stanie uwzględnić wszystkich możliwych zdarzeń boiskowych, dlatego gdy przepisy nie definiują wzorcowego postępowania w danej sytuacji, sędzia musi podjąć decyzję zgodną z duchem gry. Duch gry to pojęcie silnie oparte na idei uczciwej rywalizacji, polegające na przestrzeganiu najważniejszych wartości i zasad, które powinny towarzyszyć w trakcie meczu zawodnikom, sędziom oraz osobom funkcyjnym. Należy pamiętać, że sporo zdarzeń boiskowych wymagających interwencji arbitra jest subiektywnych. Ten fakt generuje nierzadko sporo emocji, często silnie negatywnych, które potęgują się z każdą kolejną minutą meczu. Należy jednak pamiętać, że sędzia to tylko człowiek, mający prawo popełnić błąd, dlatego jego decyzje muszą być szanowane, nawet w okolicznościach temu nieprzychylnych. Kluczową cechą, którą powinien posiadać każdy sędzia piłki nożnej jest bezstronność. Oznacza to, iż arbiter nie może faworyzować jakiejkolwiek drużyny czy zawodnika. Równie istotną cechą jest odporność psychiczna. Zarówno w meczach na szczeblu centralnym, jak i spotkaniach niższych lig amatorskich, arbiter narażony jest na olbrzymią presję ze strony rywalizujących drużyn. Oprócz tego z trybun może usłyszeć niezliczoną ilość wyzwisk i wulgaryzmów, będących głosem niezadowolenia kibiców z podjętych przez sędziego decyzji. Zachowanie dobrej formy fizycznej jest także bardzo ważne dla każdego arbitra, ponieważ pozwala to wytrzymać tempo spotkania, czyli mówiąc w żargonie piłkarskim, wybiegać mecz. Dodatkowo dobrze wyglądającą, sportową sylwetką sędzia zyskuje zaufanie i szacunek w oczach zawodników oraz kibiców.

1. Podstawowe cele i funkcjonalności chatbota ’Refi’
   1. Wstęp

W dalszej części pracy przedstawiono projekt autora, mianowicie chatbota o nazwie ‘Refi’, który udziela odpowiedzi na pytania dotyczące przepisów gry w piłkę nożną. Szczegółowo opisano proces tworzenia oraz sposób działania wspomnianego programu konwersacyjnego. Niniejszy rozdział koncentruje się głównie na zachowaniach i zadaniach chatbota w konkretnych sytuacjach. W pierwszej kolejności przedstawiono kluczowe elementy każdego programu konwersacyjnego. Następnie określono cel oraz zakres pracy. Na końcu zaprezentowano najistotniejsze wymagania chatbota 'Refi', dzieląc je na funkcjonalne i pozafunkcjonalne.

* 1. Architektura chatbota

Program typu chatbot, przeważnie tworzony jest w formie aplikacji desktopowej, mobilnej lub webowej. Autor pracy zdecydował się na wybór pierwszej opcji, czyli aplikacji desktopowej, która charakteryzuje się tym, że działa na laptopach bądź komputerach stacjonarnych, a nie przez przeglądarkę, czy na urządzeniach mobilnych. Chatboty zazwyczaj składają się z trzech podstawowych elementów. Pierwszym elementem, który jest kluczowy dla prawidłowego działania programu konwersacyjnego jest jego silnik, będący ładnie określany sercem programu. Jest on odpowiedzialny za rozpoznawanie, przetwarzanie i właściwą interpretację wprowadzanych zdań. W zależności od złożoności programu, silnik może opierać się na prostych regułach lub wykorzystywać zaawansowane modele uczenia maszynowego. Drugim elementem chatbota jest interfejs, czyli element programu, za pomocą którego użytkownicy mogą się z nim komunikować. Najczęściej interfejs składa się z okna konwersacji, gdzie wyświetlana jest rozmowa pomiędzy chatbotem a użytkownikiem, oraz pola do wprowadzania tekstu, czyli zapytania na które program powinien zareagować. Nierzadko w interfejsie znajduje się także grafika lub zdjęcie przedstawiające wizualizację chatbota. Trzecim, a zarazem najistotniejszym elementem, któremu programista musi poświęcić zdecydowanie najwięcej czasu jest baza wiedzy. Są to zasoby, z których chatbot czerpie odpowiedzi na pytania zadawane przez użytkownika. Bazę wiedzy można zaimplementować na różne sposoby, na przykład jako pliki tekstowe lub relacyjne bazy danych. W przypadku programów konwersacyjnych największą popularnością ze względu na swoją prostotę, dostępność i kompatybilność cieszą się te, które są oparte na języku AIML.

* 1. Cel pracy

Celem pracy było opracowanie chatbota, który potrafi odpowiadać na pytania dotyczące przepisów gry w piłkę nożną. Aby umożliwić użytkownikom korzystanie z programu konwersacyjnego, został stworzony prosty interfejs graficzny. Dzięki niemu użytkownicy mogą wpisywać swoje pytania, a następnie wyświetlana jest im odpowiedź. Chatbot potrafi ekspresowo, ale co najważniejsze poprawnie odpowiadać na różnego rodzaju pytania dotyczące przepisów gry w piłkę nożną, takie jak:

* Czy każdy rzut wolny bezpośredni zawsze wykonywany jest z miejsca przewinienia?
* Jak długo trwa przerwa przed dogrywką?
* Ile zawodników musi otrzymać czerwoną kartkę, aby sędzia zakończył zawody przed upływem regulaminowego czasu gry?
* Kto ma prawo towarzyszyć zawodnikom rezerwowym w trakcie rozgrzewki?

Zakres pracy obejmuje pełny proces tworzenie programu konwersacyjnego zintegrowanego z interfejsem graficznym, od projektowania do implementacji i testowania.

* 1. Wymagania funkcjonalne

Wymagania funkcjonalne są opisem cech, które system musi posiadać, aby spełnić potrzeby i oczekiwania użytkowników. Chatbot „Refi” posiada następujące wymagania funkcjonalne:

* Możliwość rozpoczęcia nowej konwersacji.
* Dostępność poprzez prosty interfejs graficzny, w którym użytkownik będzie mógł wprowadzać swoje zapytania.
* Możliwość odpowiadania na zadane przez użytkownika pytania.
* Obsługa sytuacji, w których użytkownik zadaje pytanie niezwiązane z przepisami gry w piłkę nożną.
* Obsługa sytuacji, w których użytkownik zadaje pytanie, które jest niejasne i nie można na nie udzielić konkretnej odpowiedzi.
  1. Wymagania pozafunkcjonalne

Wymagania pozafunkcjonalne to cechy, które system musi posiadać, aby spełniać określone kryteria jakości. Chatbot „Refi” posiada następujące wymagania pozafunkcjonalne:

* Chatbot odpowiadając na pytania powinien używać języka polskiego.
* Chatbot powinien być łatwy w obsłudze oraz posiadać intuicyjny interfejs.
* Chatbot powinien być w stanie działać bez przerw.
* Powinna być możliwość poszerzania bazy wiedzy chatbota o nowe informacje
* Powinna być możliwość modyfikowania bieżącej bazy wiedzy chatbota.

1. Wykorzystane narzędzia i technologie

W niniejszym rozdziale przedstawiono techniczną stronę chatbota „Refi”, którego celem jest udzielanie odpowiedzi na pytania dotyczące przepisów gry w piłkę nożną. Rozdział ten skupił się na narzędziach oraz technologiach wykorzystanych do stworzenia programu konwersacyjnego będącego tematem pracy autora. Każdemu z tych rozwiązań poświęcono osobny podrozdział, w którym opisano jego właściwości oraz możliwe zastosowania.

3.1. Microsoft Visual Studio

Przy realizacji aplikacji jako środowisko programistyczne zostało użyte Visual Studio 2022 w wersji Community, która jest dostępna za darmo dla niekomercyjnych projektów. Visual Studio jest używane między innymi do tworzenia oprogramowania konsolowego, desktopowego, mobilnego czy webowego. Aplikacje mogą być pisane na systemy operacyjne między innymi takie jak: Microsoft Windows, Windows Phone, Linux, czy MacOS. Microsoft Visual Studio umożliwia także pisanie aplikacji na platformę programistyczną .NET, jak również konsolę XBOX, czyli urządzenie do gier działające na systemie Windows[18]. Microsoft Visual Studio to oprogramowanie wykorzystujące licencję open source, co oznacza, że może być używane za darmo przez osoby prywatne, niekomercyjne grupy deweloperów oraz instytucje edukacyjne. W ramach Visual Studio znajduje się szereg narzędzi i funkcjonalności przeznaczonych dla programistów, takich jak edytor kodu z funkcją IntelliSense, narzędzie do refaktoryzacji kodu, czy zintegrowany debugger. Dodatkowo, Visual Studio zawiera designer do tworzenia aplikacji Windows Forms, WPF i web oraz narzędzia umożliwiające tworzenie klas czy projektowania baz danych. Funkcjonalności Visual Studio mogą być stale rozszerzana przy pomocy licznych dodatków i wtyczek dostępnych w sklepie Visual Studio Marketplace.

3.2. C#

C# to język programowania, który pozwala na pracę w różnych paradygmatach, zaprojektowany przez zespół pod kierownictwem Andersa Hejlsberga dla firmy Microsoft. Jego nazwa pochodzi od znaku ♯ (krzyżyk), powszechnego w notacji muzycznej, który zwiększa wysokość dźwięku o pół tonu. Może to sugerować, że C# został zaprojektowany jako rozwinięcie języków programowania C oraz C++. Jednym z kluczowych elementów C# jest wsparcie dla zasad programowania obiektowego, takich jak enkapsulacja, dziedziczenie i polimorfizm. Pozwala to programistom na tworzenie skomplikowanego, modułowego kodu, który jest łatwy do utrzymania i ponownego wykorzystania. C# obsługuje również inne paradygmaty programowania, takie jak programowanie funkcjonalne i sterowanie zdarzeniami, dzięki czemu jest on uniwersalnym językiem do różnych zastosowań. Język C# zawiera również sporo wbudowanych funkcji, które upraszczają typowe zadania programistyczne, takie jak automatyczne zarządzanie pamięcią, co eliminuje potrzebę ręcznego zarządzania alokacją i zwalnianiem pamięci przez programistów[19]. Posiada także szeroką gamę typów danych. Innym ważnym aspektem języka C# jest jego zdolność do obsługi kodu natywnego i zarządzanego. Kod natywny to kod, który działa bezpośrednio na sprzęcie komputera, natomiast kod zarządzany to kod, który działa w środowisku wykonawczym, takim jak .NET Framework. To pozwala na użycie języka C# w różnych rodzajach aplikacji. Oprócz bogatego zestawu funkcji język C# korzysta z olbrzymiej i aktywnej społeczności programistów, która zapewnia liczne zasoby i pomoce. Obejmuje to szeroką gamę bibliotek i platform otwartoźródłowych jak również wiele przydatnych narzędzi programistycznych.

3.3. Windows Forms

Windows Forms jest składnikiem platformy .NET Framework, który umożliwia tworzenie interfejsu graficznego dla aplikacji okienkowych poprzez wybór z kolekcji dostępnych elementów. Takie aplikacje charakteryzują się wysoką reaktywnością na działania użytkownika, co oznacza, że są w stanie dynamicznie dostosowywać się i reagować na wykonane przez niego czynności, które określane są zdarzeniami. Takie zdarzenia to między innymi kliknięcie przycisku myszy, przesunięcie suwaka czy naciśniecie konkretnego klawisza na klawiaturze. Po zajściu określonego zdarzenia aplikacja wykonuje konkretne zadania, zdefiniowane przez programistę. Na przykład po wprowadzeniu pytania do pola tekstowego, odpowiedź zostanie zwrócona użytkownikowi dopiero po kliknięciu odpowiedniego przycisku. WinForms świetnie sprawdza się przy budowaniu chatbotów, gdyż pozwala na szybkie i tworzenie pól tekstowych, przycisków, które są podstawowymi elementami graficznego interfejsu programów konwersacyjnych. Dzięki temu, że Windows Forms jest częścią .NET Framework, możliwa jest integracja z innymi narzędziami takimi jak bazy danych lub usługi sieciowe[20].

3.4. Pandorabot

Pandorabot to platforma open-source umożliwiająca budowanie chatbotów. Posiada ona szeroki zakres narzędzi i zasobów do tworzenia, wdrażania i zarządzania chatbotami. Platforma Pandorabot posiada bibliotekę gotowych modułów, które umożliwiają sprawną realizację skomplikowanych zadań, takich jak przetwarzanie języka naturalnego czy uczenie maszynowe[21]. Pandorabot wykorzystuje Artificial Intelligence Markup Language (AIML) jako standardowy format reprezentujący reguły i wzorce rozmowy. Jest to bardzo ważny element platformy. Pozwala on programistom na intuicyjne tworzenie chatbotów, które potrafią rozumieć i poprawnie odpowiadać na zapytania języka naturalnego. Dodatkowo dopuszcza modyfikowanie i rozszerzanie funkcjonalności już istniejących poprzez dodawanie lub zmienianie kategorii AIML. Najważniejsze funkcje które oferuje platforma Pandorabot prezentują się następująco:

* Możliwość tworzenie chatbotów za pomocą interfejsu graficznego, bez konieczności posiadania zaawansowanych umiejętności z zakresu programowania.
* Integracja z aplikacjami zewnętrznymi za pomocą API.
* Rozumienie i odpowiadanie na pytania użytkowników z wykorzystaniem sztucznej inteligencji i analizy języka naturalnego
* Udostępnienie narzędzi do zarządzania i śledzenia chatbotów po ich wdrożeniu, dzięki czemu można kontrolować ich działanie i efektywność.
* Dostępność dokumentacji zawierającej między innymi przykłady ułatwiające tworzenie chatbotów.

3.5. AIML

Artificial Intelligence Markup Language, w skrócie AIML jest językiem znaczników opartym na języku XML. AIML jest wykorzystywany przede wszystkim do tworzenia baz wiedzy chatbotów. Artificial Intelligence Markup Language opiera się na zestawie predefiniowanych kategorii zawierających wzorzec będący zestawem słów kluczowych lub fraz, oraz szablon będący odpowiedzią jaką udzieli program konwersacyjny, w sytuacji kiedy wzorzec zostanie odpowiednio dopasowany. W tabeli[Tabela 1] przedstawiono niektóre z najważniejszych elementów języka AIML.

Tabela 1. Najważniejsze elementy języka AIML

|  |  |
| --- | --- |
| **Znacznik** | **Opis** |
| <category> | Element główny AIML, który zawiera parę pytanie-odpowiedź. |
| <pattern> | Zawiera pytanie, które ma być dopasowane przez silnik AIML. |
| <template> | Zawiera odpowiedź, która ma być zwrócona przez silnik AIML po dopasowaniu pytania. |
| <think> | Pozwala silnikowi AIML na wykonanie pewnych działań w tle, bez zwracania odpowiedzi użytkownikowi. |
| <srai> | Umożliwia silnikowi AIML przesłanie pytania do innej kategorii, zamiast odpowiadać na nie bezpośrednio. |
| <random> | Pozwala silnikowi AIML na losowe wybieranie odpowiedzi z kilku dostępnych opcji. |
| <set> | Pozwala na ustawienie zmiennej do dalszego użycia w konwersacji. |
| <get> | Pozwala na pobranie wartości zmiennej z poprzedniego ustawienia. |
| <condition> | Umożliwia silnikowi AIML na wykonanie różnych działań w zależności od określonych warunków. |
| <aiml> | Rozpoczyna i kończy każdy dokument AIML. |

Źródło: [22]

Dodatkowo język AIML umożliwia tworzenie tak zwanych wildcard patterns, czyli wzorców zawierających symbole takie jak '\*' czy '\_'. Symbole te pozwalają na dopasowywanie wielu różnych fraz do jednej, konkretnej odpowiedzi. Dzięki temu możliwe jest przetwarzanie zapytań w sposób znacznie bardziej elastyczny i uniwersalny. Wildcard patterns są szczególnie przydatne w kontekście tworzenia baz wiedzy chatbotów, ponieważ pozwalają dopasować prawidłową odpowiedź do różnych zapytań użytkowników, bez potrzeby tworzenia osobnej reguły dla każdej możliwej formy zapytania.

3.6. AIMLBot

AIMLBot to otwartoźródłowa biblioteka napisana w języku C#. Umożliwia ona tworzenie chatbotów opierających się na języku AIML. Biblioteka AIMLBot zawiera gotowe przykłady i szablony, które mogą być używane do tworzenia własnych chatbotów. Programiści mogą również dostosowywać i rozszerzać bibliotekę, aby spełniała ich wymagania. AIMLBot umożliwia łatwe implementowanie reguł AIML i interakcji z użytkownikiem, co pozwala tworzyć proste ale również bardziej zaawansowane chatboty z interaktywnymi funkcjami. Firmą rozwijającą bibliotekę jest Ntoll, a jej ostatnia aktualizacja miała miejsce 8 sierpnia 2022 roku.

3.7. GIMP

Do tworzenia elementów graficznych chatterbota, autor pracy wykorzystał GNU Image Manipulation Program, w skrócie GIMP. Jest to otwartoźródłowy oraz co ważne darmowy program graficzny. Jest uznawany za najpopularniejsze narzędzie do obróbki graficznej, a o prym rywalizuje z płatnymi odpowiednikami takimi jak: Adobe Photoshop, PaintShop Pro, czy Affinity Photo. Warto podkreślić, że GIMP jest programem do edycji grafiki rastrowej, czyli ​​tworzy i edytuje piksele, a nie wektory tak jak w przypadku edytorów grafik wektorowych[23]. Oznacza to, że program GIMP miałby spore problemy z skuteczną realizacją projektów przeznaczonych do druku w dużych formatach oraz skalowaniem grafiki bez utraty jakości. W programie GIMP domyślna przestrzeń koloru to RGB. Jest to skrót od Red, Green, Blue. RGB to model koloru, który składa się z trzech barw: czerwonej, zielonej i niebieskiej. W modelu RGB, każdy składnik jest reprezentowany przez liczbę od 0 do 255, co daje możliwość wygenerowania ponad 16 milionów różnych odcieni kolorów. Program umożliwia wykonywanie różnorodnych operacji, takich jak: retuszowanie, rysowanie, dodawanie tekstu, praca na warstwach i wiele innych. GIMP to narzędzie graficzne oferujące szeroki zakres możliwości. Umożliwia tworzenie prostych grafik, ale również świetnie sprawdza się przy realizacji znacznie bardziej skomplikowanych zadań.

1. Implementacja

W rozdziale ‘Implementacja’ przedstawiono program konwersacyjny 'Refi', który został stworzony przez autora niniejszej pracy dyplomowej. Chatbot ‘Refi’ potrafi szybko i precyzyjnie odpowiadać na pytania dotyczące zasad gry w piłkę nożną. Program konwersacyjny w swoich odpowiedziach cytuje fragmenty Przepisów Gry 2022/2023, które zostały wydane przez IFAB, czyli organizację zajmująca się tworzeniem oraz nowelizacją przepisów gry w piłkę nożną. Na początku niniejszego rozdziału dogłębnie opisano strukturę całej aplikacji. Następnie przedstawiono bazę wiedzy chatbota ‘Refi’, która składa się z reguł i wzorców dotyczących przepisów gry w piłkę nożną. W dalszej części zaprezentowano graficzny interfejs użytkownika, który umożliwia łatwe i intuicyjne zadawanie pytań chatbotowi. Graficzny interfejs użytkownika został przedstawiony w szczegółowy sposób, z uwzględnieniem wszystkich wykorzystanych elementów graficznych, takich jak pola tekstowe, przyciski, czy okna dialogowe. Na końcu tego rozdziału opisano proces generowania odpowiedzi przez chatbota, który stanowi kluczową funkcjonalność każdego programu konwersacyjnego.

* 1. Struktura projektu

Chatbot 'Refi' jest aplikacją desktopową, czyli działającą na komputerze stacjonarnym lub laptopie, a nie na urządzeniu mobilnym, bądź w przeglądarce internetowej. Aplikacje desktopowe, w przeciwieństwie do aplikacji webowych nie wymagają połączenia z internetem i są zazwyczaj bardziej wydajne. Program konwersacyjny autora został zaimplementowany w języku C#, z wykorzystaniem otwartoźródłowego frameworka Windows Forms. Klasą główną chatbota ‘Refi’, zawierającą między innymi kod uruchamiania aplikacji, jest klasa Program.cs. Ponadto, chatbot posiada cztery formularze, które są integralnymi częściami programu konwersacyjnego. Każdy formularz ma określoną specyficzną funkcję, a odpowiednie połączenie wszystkich formularzy tworzy spójną i przyjazną dla użytkownika całość aplikacji.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 4. Struktura projektu Chatbot Refi Źródło: Opracowanie własne

4.1.1. Klasa Program.cs

Klasa Program.cs jest klasą główną programu konwersacyjnego ‘Refi’. Klasa ta zawiera metodę startową Main, która uruchamia program i inicjalizuje jego działanie. Metoda Main jest zdefiniowana jako metoda statyczna. Dzięki temu środowisko .NET Framework potrafi rozpoznać ją jako punkt wejścia do programu. Wymóg zdefiniowania metody Main jako statycznej jest spowodowane faktem, iż w momencie uruchomienia aplikacji nie istnieje jeszcze żadna instancja klasy, a więc nie można wywołać metody niestatycznej. W metodzie Main są wywoływane trzy metody:

**Application.EnableVisualStyles()** – to metoda, która włącza stylizację graficzną dla kontrolek wizualnych w aplikacji. Domyślnie, bez wywołania tej metody, kontrolki wizualne są stylizowane w sposób prosty oraz podstawowy. Włączenie tej stylizacji powoduje że wygląd całej aplikacji jest spójny, profesjonalny i bardziej estetyczny.

**Application.SetCompatibleTextRenderingDefault(false)** - to metoda, która ustawia kompatybilność renderowania tekstu na domyślną wartość false. Oznacza to, że tekst renderowany jest przez bibliotekę GDI. Zachowanie renderowania tekstu domyślnie ustawione jest na kompatybilność z poprzednimi wersjami .NET, co może powodować problemy z czytelnością tekstu i wydajnością całej aplikacji.

**Application.Run(new LoginForm())** - to metoda, która uruchamia formularz logowania, który jest pierwszym oknem aplikacji, więc wyświetla się użytkownikowi bezpośrednio po uruchomieniu programu konwersacyjnego ‘Refi’.

4.1.2. Formularz LoginForm

LoginForm to formularz, który wyświetla się bezpośrednio po uruchomieniu programu konwersacyjnego ‘Refi’. Formularz "LoginForm" został stworzony w celu umożliwienia użytkownikom autoryzowanego dostępu do aplikacji poprzez funkcję logowania. Poprawność loginu oraz hasła jest weryfikowana w bazie danych, którą połączono z aplikacją wykorzystując obiekt SqlConnection. Bazą danych może zarządzać wyłącznie administrator systemu. Tylko zalogowani użytkownicy mają możliwość interakcji z chatbotem ‘Refi’. Jeżeli użytkownik wprowadzi błędne dane logowania, zostanie mu wyświetlony stosowny komunikat, zawierający informacją o niepowodzeniu przy próbie zalogowania się do systemu.

4.1.3. Formularz WelcomeForm

Po pomyślnym zalogowaniu użytkownik zostaje przekierowany do formularza WelcomeForm, w którym zostaje przywitany przez chatbota. W formularzu WelcomeForm znajduje się także krótki opis przedstawiający program konwersacyjny ‘Refi’. Na końcu tekstu powitalnego, chatbot zaprasza użytkownika do zadawania pytań.

4.1.4. Formularz QuestionForm

QuestionForm to formularz w którym użytkownik ma możliwość zadać chatbotowi ‘Refi’ dowolne pytanie dotyczące przepisów gry w piłkę nożną, wpisując je w specjalnie przygotowane do tego pole tekstowe.

4.1.5. Formularz AnswerForm

W formularzu AnswerForm pojawia się pytanie zadane w formularzu QuestionForm oraz prezentowana jest odpowiedź wygenerowana przez program konwersacyjny. Chatbot ‘Refi’ generuje odpowiedzi na podstawie swojej obszernej bazy wiedzy, która została szczegółowo opisana w kolejnym podrozdziale.

* 1. Baza wiedzy

Baza wiedzy programu konwersacyjnego ‘Refi’ jest przechowywana na dysku w katalogu o nazwie aiml. Baza wiedzy została rozdzielona na 17 części, z których każda część odpowiada jednemu artykułowi z Przepisów Gry w piłkę nożną. Taki podział sprawia, że baza wiedzy jest znacznie bardziej przejrzysta i łatwa do zarządzania, co pozytywnie wpływa na jej użyteczność. ”Przepisy Gry 22/23” były jedynym źródłem informacji wykorzystanym do stworzenia obszernej bazy wiedzy chatbota ‘Refi”. Wykorzystanie najnowszej wersji przepisów gry daje pewność, że baza wiedzy programu konwersacyjnego jest aktualna i zgodna z obowiązującymi zasadami. Taka baza wiedzy wzbudza zaufanie użytkowników, ponieważ mają oni pewność, że otrzymane odpowiedzi są zgodne z obowiązującymi przepisami gry.

Do stworzenia spójnej oraz klarownej bazy wiedzy, użyto platformy Pandorabots, która jest bardzo często wykorzystywana przez programistów do tego typu celów. Platforma Pandorabots zapewnia intuicyjne środowisko programistyczne, umożliwiające tworzenie, zarządzanie i rozwijanie chatbotów, których baza wiedzy opiera się na języku AIML. Przy użyciu Pandorabots, autor mógł łatwo definiować wzorce i reguły dla różnych pytań, potencjalnie zadanych przez użytkownika. Pandorabots oferuje również narzędzia do testowania i weryfikacji działania chatbota, co umożliwiło sprawdzenie poprawności odpowiedzi oraz pomogło w zlokalizowaniu błędów w kodzie.

Baza wiedzy została zaimplementowana w języku AIML, który jest językiem programowania służącym do tworzenia zaawansowanych systemów konwersacyjnych. Wszystkie pliki zawierające pytania i odpowiedzi, zostały utworzone z wykorzystaniem standardowej składni języka AIML, która opiera się na regułach i wzorcach. Pliki te posiadają rozszerzenie \*.aiml. Tworzenie bazy wiedzy programu konwersacyjnego jest niezwykle pracochłonnym procesem, dlatego autor zdecydował się szczegółowo opracować wyłącznie dwa artykuły: artykuł 7 - Czas trwania zawodów, jak również artykuł 15 - Wrzut. Plik z rozszerzeniem .aiml, skupiający się na artykule Czas trwania zawodów, składa się ze zbioru 198 reguł, nazywanych w języku AIML kategoriami. Całkowita liczba linijek kodu w tym pliku wynosi 1002. Natomiast plik z kodem AIML, dotyczącym artykułu Wrzut, zawiera 171 kategorii oraz 916 linijek kodu. Tak duże liczby świadczą o niezwykłej dokładności i staranności autora przy tworzeniu bazy wiedzy chatbota ‘Refi’. Fragment kodu AIML, który dotyczy artykułu 15 – Wrzut przedstawiono na rysunku [Rysunek 5].

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 5. Fragment kodu AIML pliku 15\_Wrzut.aiml Źródło: Opracowanie własne

Każda kategoria w języku AIML składa się z dwóch głównych elementów: wzorca pytania (pattern) i szablonu odpowiedzi (template)[24].

1. Wzorzec pytania (pattern) - definiuje, jakie zapytania będą dopasowane do danej kategorii. Może zawierać symbole wieloznaczne takie jak ‘\*’, które reprezentują dowolne słowo lub wyrażenie.
2. Szablon odpowiedzi (template) – jest to treść, która zostanie wyświetlona jako odpowiedź na zapytania, które pasują do wzorca pytania.

Kategoria, która na rysunku [Rysunek 5] została przedstawiona jako pierwsza, posiada następujący wzorzec pytania, który prezentuje się następująco:

* *KTO \* WRZUT \**

W efekcie, ta reguła będzie stosowana do pytań użytkowników, rozpoczynających się od słowa "KTO", zawierających dowolne wyrażenie przed słowem "WRZUT" i dowolne wyrażenie po nim. Na przykład, pytania "Kto wykona wrzut jeśli ostatni zagrał piłkę bramkarz gospodarzy?" lub "Kto powinien otrzymać wrzut jeśli piłka opuściła pole gry w powietrzu?" będą pasowały do wzorca, który został przedstawiony powyżej. Reguły w języku AIML są niezależne od wielkości liter. Oznacza to, że nie ma znaczenia, czy wzorce pytania lub zapytania użytkownika są napisane małymi literami, dużymi literami bądź jednocześnie małymi i dużymi literami, ponieważ język AIML traktuje je jako identyczne.

Autor ma w planach rozwijanie chatbota ‘Refi,’ poprzez rozbudowanie jego bazy wiedzy o szczegółowe opracowywania kolejnych rozdziałów Przepisów Gry. Autor ma również w planach rozszerzanie bazy wiedzy o kolejne symbole wieloznaczne. Aktualnie, w bazie wiedzy chatbota używany jest tylko jeden symbol wieloznaczny reprezentowany przez znak asterysk '\*'. Jednak autor zamierza wprowadzić większą różnorodność i precyzję w rozpoznawaniu złożonych zapytań poprzez dodawanie kolejnych symboli wieloznacznych, takich jak '\_' czy '#'. Planowane rozbudowanie i rozszerzenie bazy wiedzy chatbota 'Refi' umożliwi udzielanie odpowiedzi na pytania dotyczące pozostałych rozdziałów przepisów gry. Dodatkowo, wprowadzenie kolejnych wzorców wieloznacznych pozwoli na znacznie precyzyjniejsze dopasowywanie odpowiedzi do różnorodnych zapytań użytkowników, zapewniając tym samym wyższą jakość interakcji.

* 1. Graficzny interfejs użytkownika

W skład graficznego interfejsu użytkownika chatbota ‘Refi’ wchodzą cztery formularze[Rysunek 6], których funkcjonalności zostały szczegółowo opisane w podrozdziale struktura aplikacji. Każdy z tych formularzy został zaprojektowany w taki sposób, aby umożliwić użytkownikom proste i przyjemne korzystanie z programu konwersacyjnego. Wszystkie cztery formularze aplikacji zostały zaprojektowane z tłem w postaci zdjęcia murawy boiska piłkarskiego. Ta charakterystyczna grafika nie tylko bardzo ładnie wygląda, ale również kojarzy się z sędzią piłki nożnej, co nawiązuje do tematyki, z jaką aplikacja jest związana.

Obraz zawierający tekst, zrzut ekranu, Prostokąt, design

Opis wygenerowany automatycznie

Rysunek 6. Interfejs graficzny użytkownika chatbota ‘Refi’ Źródło: Opracowanie własne

Logowanie jest nieodłącznym elementem wielu aplikacji. Możliwość logowania się do programu konwersacyjnego 'Refi' została wprowadzona w celu zapewnienia dostępu do aplikacji jedynie uprawnionym użytkownikom oraz umożliwienia chatbotowi odwołania się w wiadomości do użytkownika na podstawie jego loginu. Formularz logowania[Rysunek 6a] zawiera dwa pola tekstowe w które użytkownik może wprowadzić swoje dane logowania: login i hasło. Pole tekstowe z hasłem posiada właściwość PasswordChar, która pozwala na ustawienie znaku (dla chatbota ‘Refi’ jest to znak ‘\*’), który zastępuje wprowadzane litery. Wykorzystanie właściwości PasswordChar zwiększa bezpieczeństwo aplikacji, ponieważ osoby trzecie, znajdujące się w bezpośrednim otoczeniu użytkownika, nie mają możliwości odczytania hasła. Na formularzu powitalnym[Rysunek 6b] znajduje się tekst, w którym chatbot ‘Refi’ przedstawia się użytkownikowi i zaprasza go do zadawania pytań dotyczących przepisów gry w piłkę nożną. Poniżej tekstu umieszczono przycisk ‘Zadaj mi pytanie’. Kolejny formularz[Rysunek 6c] zawiera edytowalne pole tekstowe, w którym użytkownik może wpisać swoje pytanie. Poniżej pola tekstowego znajduje się przycisk ‘Wyślij’, po którego kliknięciu wyświetla się formularz z odpowiedzią[Rysunek 6d]. To właśnie na tym formularzu wyświetlana jest odpowiedź na zadane wcześniej pytanie. Dodatkowo w prawym górnym rogu umieszczono przycisk ‘X’. Kliknięcie tego przycisku powoduje zamknięcie bieżącego formularza i powrót do formularza, w którym użytkownik może wpisać swoje pytanie.

Całość graficznego interfejsu użytkownika została zaprojektowana w sposób intuicyjny i łatwy w obsłudze, co pozwala na skuteczne korzystanie z systemu bez konieczności posiadania szczegółowej wiedzy technicznej. W górnej części pierwszych trzech formularzy aplikacji, umieszczono identyfikację wizualną chatbota ‘Refi’[Rysunek 7]. Jest to autorska grafika stworzona z wykorzystaniem oprogramowania GIMP. Identyfikacja wizualna programu konwersacyjnego 'Refi' wyróżnia się charakterystycznym elementem - oczami, a konkretnie źrenicami, które mają postać piłek do piłki nożnej. To oczywiste nawiązanie do futbolowej tematyki chatbota.

Obraz zawierający krąg, zrzut ekranu, Grafika, kreskówka

Opis wygenerowany automatycznie

Rysunek 7. Identyfikacja wizualna chatbota ‘Refi’ Źródło: Opracowanie własne

* 1. Generowanie odpowiedzi przez chatbota

Niewątpliwie najważniejszą funkcjonalnością każdego programu konwersacyjnego jest umiejętność generowania właściwych odpowiedzi na zadawane przez użytkownika pytania. Chatbot zwracający błędne odpowiedzi staje się bezużyteczny, a użytkownik bardzo szybko traci chęć na dalszą interakcją z takim wadliwym programem. Za generowanie poprawnych odpowiedzi przez chatbota ‘Refi’, odpowiedzialna jest funkcja GetBotResponse, której kod został przedstawiony na rysunku [Rysunek 8]. Funkcja GetBotResponse wykorzystuje bibliotekę AIMLBot, napisaną w języku C#. Biblioteka AIMLBot pozwala na implementowanie reguł i zasad, które decydują o tym, w jaki sposób chatbot będzie odpowiadał na zadane przez użytkownika pytania.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 8. Kod metody generującej odpowiedź chatbota ‘Refi’ Źródło: Opracowanie własne

Implementacja funkcji GetBotResponse zaczyna się od utworzenia obiektu klasy Bot, reprezentującego chatbota. Następnie, z wykorzystaniem funkcji loadSettings, chatbot ładuje ustawienia z folderu konfiguracyjnego. Kolejnym krokiem jest załadowanie plików AIML za pomocą funkcji loadAIMLFromFiles. Po wykonaniu tych operacji, chatbot jest gotowy do interakcji z użytkownikiem. W kolejnym etapie funkcja GetBotResponse wykorzystuje obiekt klasy User do reprezentowania użytkownika, który wprowadza zapytanie. Domyślnie, refi.isAcceptingUserInput jest ustawione na wartość false, co oznacza, że użytkownik nie może wprowadzać danych. Po utworzeniu obiektu klasy User, pole isAcceptingUserInput przyjmuje wartość true, co oznacza, że użytkownik może wprowadzać dane. Następnie, funkcja GetBotResponse tworzy obiekt klasy Request, który reprezentuje zapytanie użytkownika. W ramach tej funkcji, zostaje przekazany tekst wprowadzony przez użytkownika, obiekt klasy User oraz obiekt klasy Bot. Na tym etapie, chatbot wykorzystuje reguły AIML, które zostały zdefiniowane w bazie wiedzy, do generowania odpowiedzi na zadane pytania. Wynik zwracany przez chatbota jest reprezentowany przez obiekt klasy Result. W ramach tej funkcji, obiekt klasy Result przechowuje odpowiedź na zadane pytanie. Ostatecznie, wynik generowany przez chatbota jest przekazywany do formularza AnswerForm. Formularz AnswerForm wyświetla pytanie zadane przez użytkownika oraz odpowiedź wygenerowaną przez chatbota ‘Refi’.

1. Testy i analiza wyników

W rozdziale „Testy i analiza wyników” szczegółowo przedstawiono proces testowania programu konwersacyjnego ‘Refi’. Testowanie chatbota jest kluczowym elementem w fazach jego projektowania oraz wdrażania. Na etapie projektowania, testowanie pozwala na wcześniejsze wykrycie i naprawę błędów, co zapobiega pojawianiu się problemów w przyszłości. Natomiast na etapie wdrażania, testowanie jest niezbędne, aby upewnić się, że aplikacja działa poprawnie na rzeczywistym środowisku. Testy przeprowadzone na obu etapach w sposób skrupulatny i precyzyjny, pozwalają skutecznie zweryfikować poprawność działania programu konwersacyjnego. Udana weryfikacja poprawności działania chatbota, daje praktycznie pełną gwarancję, że program spełni oczekiwania użytkowników. Na końcu niniejszego rozdziału dokonano wnikliwej analizy wyników testów programu konwersacyjnego ‘Refi’.

* 1. Testowanie logowania

W podrozdziale "Testowanie logowania" przedstawione zostaną testy przeprowadzone na formularzu logowania, którego interfejs zawiera dwa pola tekstowe HasloTB i LoginTB, a także przycik LoginButton. Przed przystąpieniem do opisu testów logowania, warto dokładniej przyjrzeć się metodzie LoginButton\_Click(), której kod został przedstawiony na rysunku[Rysunek 9].

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 9. Kod metody sprawdzającej poprawność danych logowania Źródło: Opracowanie własne

Metoda LoginButton\_Click() jest wywoływany bezpośrednio po kliknięciu przycisku LoginButton. Kod znajdujący się w tej metodzie jest odpowiedzialny za sprawdzenie poprawności danych logowania użytkownika. Aby sprawdzić poprawność danych logowania, tworzone jest połączenie z bazą danych. Połączenie tworzone jest z wykorzystaniem obiektu SqlConnection. Następnie, wewnątrz metody SqlDataAdapter, wykonywane jest zapytanie SQL, które pobiera z tabeli LoginData wiersz, w którym wartość pola Login jest równa wartości pola LoginTB, a wartość pola Haslo jest równa wartości pola HasloTB. Jeśli zapytanie zwróci wartość równą 1, to znaczy, że podane dane logowania są poprawne i użytkownik zostanie przekierowany do okna powitalnego WelcomeForm poprzez wywołanie konstruktora tej klasy oraz wywołanie funkcji Show(). Natomiast jeśli zapytanie zwróci wartość inną niż 1, to pojawi się okno z informacją o błędzie logowania. Po zakończeniu procesu logowania połączenie z bazą danych zostanie zamknięte przy użyciu funkcji Con.Close().

5.1.1. Test poprawnego logowania

Test ten ma na celu sprawdzenie, czy formularz logowania działa poprawnie, gdy użytkownik wprowadzi poprawne dane logowania.

**Warunki testu:**

* W polu tekstowym LoginTB wprowadzono poprawny login.
* W polu tekstowym HasloTB wprowadzono poprawne hasło.
* Kliknięto przycisk LoginButton.

**Oczekiwany rezultat:**

* Powinno zostać wyświetlone okno powitalne WelcomeForm.
* Okno logowania powinno zostać zamknięte.
* Połączenie z bazą danych powinno zostać zamknięte.

5.1.2. Test niepoprawnego logowania

Test ten ma na celu sprawdzenie, czy formularz logowania działa poprawnie, gdy użytkownik wprowadzi niepoprawne dane logowania.

**Warunki testu:**

* W polu tekstowym LoginTB wprowadzono niepoprawny login.
* W polu HasloTB wprowadzono niepoprawne hasło.
* Kliknięto przycisk LoginButton.

**Oczekiwany rezultat:**

* Powinno zostać wyświetlone okno z informacją o błędzie logowania.
* Połączenie z bazą danych powinno zostać zamknięte.
  1. Testowanie poprawności generowania odpowiedzi

Niniejszy podrozdział poświęcono testowaniu poprawności generowanych odpowiedzi przez program konwersacyjny ‘Refi’. W celach testowych zadano chatbotowi kilka pytań dotyczących Przepisów Gry w piłkę nożną, a konkretnie artykułów 7 i 15. Artykuł 7 opisuje zagadnienia związane z czasem trwania zawodów, natomiast artykuł 15 dotyczy wrzutów. Wybrane zostały właśnie te dwa artykuły, ponieważ dotychczas tylko do nich autor stworzył kompletną bazę wiedzy. Dzięki temu chatbot powinien być w stanie poprawnie odpowiadać na wszystkie pytania związane z tymi dwoma artykułami.

5.2.1. Odpowiedzi dotyczące artykułu Czas trwania zawodów

Na rysunku[Rysunek 10] zaprezentowano zrzuty ekranu rozmów z chatbotem Refi, które koncentrują się na artykule *Czas trwania zawodów*. Zrzuty ekranu przedstawiają zarówno proste pytanie o czasu trwania meczu piłki nożnej, jak również pytania bardziej szczegółowe dotyczące doliczania straconego czasu gry. Chatbot Refi udziela wyczerpujących odpowiedzi na wszystkie zadane pytania.

***Obraz zawierający tekst

Opis wygenerowany automatycznie***

Rysunek 10. Pytania i odpowiedzi dotyczące artykułu Czas trwania zawodów Źródło: Opracowanie własne

5.2.2. Odpowiedzi dotyczące artykułu *Wrzut*

Na rysunku[Rysunek 11] przedstawiono zrzuty ekranu rozmów z chatbotem Refi, które koncentrują się na artykule *Wrzut*. Chatbot Refi udziela wyczerpujących odpowiedzi na każde z zadanych pytań. Udzielane odpowiedzi są cytatami zaczerpniętymi z Przepisów Gry w piłkę nożną.

***Obraz zawierający tekst

Opis wygenerowany automatycznie***

Rysunek 11. Pytania i odpowiedzi dotyczące artykułu Wrzut Źródło: Opracowanie własne

* 1. Testowanie wydajności

W celu oceny wydajności programu konwersacyjnego 'Refi', przeprowadzono serię testów, mających na celu zmierzenie szybkości udzielania odpowiedzi przez chatbota. Testy te miały na celu sprawdzenie, czy 'Refi' spełnia oczekiwania użytkowników pod względem czasu oczekiwania na odpowiedzi.

**Warunki testu:**

W ramach testu, przygotowano różnorodne pytania, które zostały zadane chatbotowi 'Refi'. Każde pytanie zostało poddane pomiarowi czasu, począwszy od momentu, w którym pytanie zostało zadane, aż do momentu otrzymania odpowiedzi od chatbota. W trakcie testów uwzględniono zróżnicowaną złożoność pytania oraz czas potrzebny na wyszukanie informacji w bazie wiedzy.

**Oczekiwany rezultat:**

Oczekuje się, że chatbot 'Refi' będzie udzielał odpowiedzi w odpowiednio krótkim czasie, tak aby użytkownicy nie odczuwali długiego oczekiwania. Czas udzielania odpowiedzi powinien być zgodny z oczekiwaniami użytkowników, biorąc pod uwagę złożoność zadanych pytań oraz czas potrzebny na przeszukanie bazy wiedzy.

**Wyniki:**

Poniższa tabela [Tabela 2.] przedstawia wyniki testów wydajności chatbota 'Refi'.

Tabela 2. Wyniki testów wydajności chatbota 'Refi'

|  |  |  |
| --- | --- | --- |
| **Nr testu** | **Pytanie** | **Czas oczekiwania na odpowiedź (w sekundach)** |
| **1** | *Jaki jest czas trwania meczu piłkarskiego?* | **1,03** |
| **2** | *Kto pokazuje doliczony czas w trakcie meczu?* | **1,01** |
| **3** | *Ile trwa odpoczynek w przerwie między połowami?* | **0,97** |
| **4** | *Czy wrzut z autu można wykonać jedną ręką?* | **1,20** |
| **5** | *Kto wykona wrzut jeśli piłka odbiła się od sędziego?* | **1,11** |

Źródło: Opracowanie własne

Średni czas oczekiwania na udzielenie odpowiedzi przez chatbota 'Refi' wyniósł 1,06 sekundy. Te wyniki sugerują, że chatbot 'Refi' jest w stanie udzielać odpowiedzi w odpowiednio krótkim czasie, zapewniając użytkownikom szybką reakcję na zadane pytania dotyczące wrzutów z autu i czasu trwania zawodów piłkarskich.

* 1. Testowanie użyteczności

Testy użyteczności pozwalają dowiedzieć się w jaki sposób aplikacja jest używana, jakie budzi emocje oraz jakie wywołuje reakcje[25]. Celem testowania użyteczności jest identyfikacja potencjalnych problemów, trudności lub ograniczeń, które mogą wystąpić podczas użytkowania aplikacji. Testy użyteczności dostarczają cenne dla rozwoju systemu informacje i sugestie dotyczące potencjalnych ulepszeń, bądź aktualizacji.

Do testowania użyteczności programu konwersacyjnego ‘Refi’ zaproszono dwóch sędziów, posiadających aktualną licencję Polskiego Związku Piłki Nożnej oraz dwóch zawodników grających w klasie okręgowej, która jest szóstym poziomem rozgrywkowym w Polsce. Każdy z uczestników zadał chatbotowi pięćdziesiąt pytań, które dotyczyły artykułów 7 i 15 przepisów gry w piłkę nożną, a następnie zebrano wyniki oraz opinie testerów.

W celu oceny odpowiedzi udzielanych przez program konwersacyjny ‘Refi’, zastosowano system punktacji, gdzie za poprawną odpowiedź przyznawano jeden punkt, za częściowo poprawną odpowiedź przyznawano 0,5 punktu, a za niepoprawną odpowiedź przyznawano 0 punktów. W przypadku niepoprawnych odpowiedzi, należy zauważyć, że oznacza to, iż program konwersacyjny błędnie zrozumiał kontekstu pytania lub było one niejednoznaczne. Baza wiedzy chatbota opiera się wyłącznie na Przepisach Gry, co gwarantuje, że każda udzielona odpowiedź jest wiarygodna, czyli pozbawiona jakichkolwiek fałszywych treści, niezgodnych z oficjalnymi zasadami gry w piłkę nożną. W tabeli [Tabela 2] przedstawiono wyniki testów użyteczności programu konwersacyjnego 'Refi'

Tabela 3. Wyniki testów użyteczności programu konwersacyjnego 'Refi'

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **SĘDZIA 1** | **SĘDZIA 2** | **ZAWODNIK 1** | **ZAWODNIK 2** |
| Ilość zadanych pytań | 50 | 50 | 50 | 50 |
| Ilość poprawnych odpowiedzi | 42 | 40 | 45 | 46 |
| Ilość częściowo poprawnych odpowiedzi | 4 | 7 | 3 | 4 |
| Ilość niepoprawnych odpowiedzi | 4 | 3 | 2 | 0 |
| **Suma punktów** | **44** | **43,5** | **46,5** | **48** |

Źródło: Opracowanie własne

Chatbot ‘Refi’ udzielił całkowicie poprawnych odpowiedzi:

* Sędziemu 1 na 84% zadanych pytań.
* Sędziemu 2 na 80% zadanych pytań.
* Zawodnikowi 1 na 90% zadanych pytań.
* Zawodnikowi 2 na 92% zadanych pytań.

Wyniki te wskazują na bardzo wysoką skuteczność chatbota w udzielaniu odpowiedzi na zadawane pytania. Jednocześnie wyniki sugerują, że sędziowie zadawali bardziej złożone i skomplikowane pytania niż piłkarze, co przyczyniło się do gorszego wyniku punktowego arbitrów. Jednak ogólnie rzecz biorąc, wyniki testów potwierdzają wiarygodność i użyteczność programu konwersacyjnego 'Refi' w kontekście udzielania precyzyjnych i satysfakcjonujących informacji na temat przepisów gry w piłkę nożną.

* 1. Wnioski z testów

Podczas przeprowadzonych testów programu konwersacyjnego ‘Refi’, skupiono się na dwóch kluczowych aspektach: testowaniu logowania oraz testowaniu poprawności generowanych odpowiedzi przez chatbota.

W ramach pierwszej części przetestowano poprawność funkcjonowania mechanizmu logowania do programu konwersacyjnego. W tym celu przeprowadzono szereg testów, w których wprowadzano różne dane logowania. Testy te wykazały, że mechanizm logowania działa prawidłowo, a aplikacja poprawnie identyfikuje poprawne i niepoprawne dane logowania. Druga część dotyczyła testów poprawności generowania odpowiedzi przez program konwersacyjny. W tym celu zadano chatbotowi kilka pytań dotyczących wybranych artykułów z Przepisów Gry w piłkę nożną, a następnie sprawdzono, czy wygenerowane odpowiedzi były poprawne. Testy wykazały, że chatbot odpowiada w sposób właściwy, a generowane odpowiedzi są zgodne z Przepisami Gry w piłkę nożną. Kolejno przeprowadzono testowanie wydajności programu konwersacyjnego 'Refi'. W tym celu zmierzono czas oczekiwania od zadania pytania do otrzymania odpowiedzi. Seria testów obejmowała pytania dotyczące wyłącznie wrzutu z autu oraz czasu trwania zawodów w piłce nożnej. Natomiast na końcu przetestowano użyteczność chatbota.

Wnioski z przeprowadzonych testów są pozytywne. Program konwersacyjny autora działa bez zarzutu i spełnia wszystkie postawione przed nim wymagania. Można więc uznać, że chatbot ‘Refi’ jest w pełni gotowy na interakcję z użytkownikami, którzy szukają odpowiedzi na pytania dotyczące przepisów gry w piłkę nożną.

ZAKOŃCZENIE

Celem niniejszej pracy było opracowanie chatbota, czyli programu komputerowego umożliwiającego interakcję z użytkownikiem w formie rozmowy. Autor stworzył chatbota, który potrafi udzielać odpowiedzi na pytania dotyczące przepisów gry w piłkę nożną i nazwał go ‘Refi’. Teoretyczna część pracy koncentrowała się na dwóch obszarach tematycznych. Pierwszy z nich był z dziedziny informatyki, bowiem odnosi się do programów konwersacyjnych. Drugi zaś dotyczył sportu, a konkretnie przepisów gry w piłkę nożną. Przed przystąpieniem do tworzenia chatbota, konieczne było gruntowne zgłębienie obu tematów. Po wyczerpującej części teoretycznej skupiono się na wyznaczeniu celów oraz funkcjonalności chatbota 'Refi'. Praca obejmowała również implementację programu konwersacyjnego, w tym strukturę projektu, bazę wiedzy, graficzny interfejs użytkownika, jak również proces generowania odpowiedzi.

Postawione cele zostały zrealizowane pomyślnie. Chatbot 'Refi' został stworzony i potwierdził swoją zdolność do udzielania odpowiedzi na pytania związane z przepisami gry w piłkę nożną. Za sprawą programu konwersacyjnego autora, użytkownicy mogą bardzo łatwo uzyskiwać informacje dotyczące zasad gry w piłkę nożną. Stanowi to istotne wsparcie oraz ułatwienie dla licznej grupy entuzjastów tego najpopularniejszego na świecie sportu.

Jednym z problemów, napotkanych podczas tworzenia programu konwersacyjnego 'Refi' była bazy wiedzy chatbota, a w zasadzie jej tworzenie, które okazało się ogromnie czasochłonnym zajęciem. Uporządkowanie i przeanalizowanie wszystkich przepisów gry w piłkę nożną okazało się wyzwaniem, które wymagało gigantycznych zasobów czasowych. Aby rozwiązać ten problem, zdecydowano się ograniczyć bazę wiedzy chatbota wyłącznie do dwóch wybranych rozdziałów przepisów gry.

W przyszłości chatbot 'Refi' będzie poddawany dalszemu rozwojowi. Istnieje wiele możliwości usprawnienia oraz rozszerzenia funkcjonalności programu konwersacyjnego autora. Jednym z kierunków rozwoju może być rozbudowanie jego bazy wiedzy o kolejne artykuły przepisów gry w piłkę nożną. Ponadto, można rozważyć integrację chatbota z platformami komunikacyjnymi, aby zapewnić większą dostępność i wygodę użytkownikom.

LITERATURA

[1] T. Walsh, To żyje! Sztuczna inteligencja. Od logicznego fortepianu po zabójcze roboty., PWN, s.15

[2] J. Weizenbaum, ELIZAa computer program for the study of natural language communication between man and machine, „Communications of the ACM”, 1966, s. 36

[3] Chatbot History: The Parry Chatbot, www.yakbots.com/chatbot-history-the-parry-chatbot/ (dostęp: 22.12.2022)

[4] Experts Doubt Ethical AI Design Will Be Broadly Adopted as the Norm Within the Next Decade, www.pewresearch.org/internet/2021/06/16/experts-doubt-ethical-ai-design-will-be-broadly-adopted-as- the-norm-within-the-next-decade/ (dostęp: 22.12.2022)

[5] A. Castrounis, AI dla ludzi i firmy, O’Reilly, 2020, s. 23

[6] A.L.I.C.E., https://www.chatbots.org/chatbot/a.l.i.c.e/ (dostęp: 01.02.2023)

[7] Eleven outstanding world records in AI, https://indiaai.gov.in/article/eleven-outstanding-world-records-in-ai (dostęp: 11.03.2023)

[8] Sztuczna inteligencja w mainstreamie. ChatGPT bije rekordy popularności, podobnie jak Lensa AI, https://cyberdefence24.pl/technologie/sztuczna-inteligencja-w-mainstreamie-chatgpt-bije-rekordy-popularnosci-podobnie-jak-lensa-ai (dostęp: 11.03.2023)

[9] A. Kao, Natural language processing and text mining, Springer, 2007, s.1

[10] What Are Bots? How Do Chatbots Work?, https://botscrew.com/blog/what-are-bots/ (dostęp: 12.03.2023)

[11] We've already talked about it - Repetition in the Chatbot, https://liberty-chatbot.space/en/articles/598ac4fe97462d602e441fe8 (dostęp: 13.02.2023)

[12] AI Chatbots Are Getting Better. But an Interview With ChatGPT Reveals Their Limits, https://time.com/6238781/chatbot-chatgpt-ai-interview (dostęp: 22.02.2023)

[13] Co to jest DDoS? Na czym polega atak DDoS?, https://bezpiecznyinternet.edu.pl/co-to-jest-ddos/ (dostęp: 13.11.2022)

[14] Sports Industry: A Research Guide, https://guides.loc.gov/sports-industry/soccer (dostęp: 11.01.2023)

[15] Przepisy gry 2022/2023, IFAB, 2022, s. 9

[16] Przepisy gry 2022/2023, IFAB, 2022, s. 4

[17] S. Szczepłek, Moja historia futbolu, Rzeczpospolita, 2007, s. 15

[18] Dokumentacja Visual Studio, https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022 (dostęp: 24.01.2023)

[19] Dokumentacja języka C# ,https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp(dostęp: 10.01.2023)

[20] Dokumentacja Windows Forms, https://learn.microsoft.com/enus/dotnet/desktop/ winforms/overview/?view=netdesktop-6.0 (dostęp: 29.12.2022)

[21] Pandorabots Technology Deep Dive, https://pandorabots.com/static/html/About/techdeepdive.html (dostęp: 03.01.2023)

[22] Dokumentacja AIML, http://www.aiml.foundation/doc.html (dostęp: 27.03.2023)

[23] B. Witkowski, GIMP poznaj świat grafiki komputerowej, Helion, 2019, s.11

[24] AIML TUTORIAL, https://pemagrg.medium.com/aiml-tutorial-a8802830f2bf, (dostęp: 20.04.2023)

[25] Testy użyteczności, https://thestory.is/pl/proces/faza-designu/testy-uzytecznosci/ (dostęp: 28.05.2023)

SPIS RYSUNKÓW I TABEL

Spis rysunków

[Rysunek 1. Oś czasu z datami utworzenia najważniejszych chatbotów. 8](#_Toc137062984)

[Rysunek 2. Ogólny sposób działania chatbotów 12](#_Toc137062985)

[Rysunek 3. Spis artykułów znajdujących się w przepisach gry w piłkę nożną 16](#_Toc137062986)

[Rysunek 4. Struktura projektu Chatbot Refi 28](#_Toc137062987)

[Rysunek 5. Fragment kodu AIML pliku 15\_Wrzut.aiml 31](#_Toc137062988)

[Rysunek 6. Interfejs graficzny użytkownika chatbota ‘Refi’ 33](#_Toc137062989)

[Rysunek 7. Identyfikacja wizualna chatbota ‘Refi’ 34](#_Toc137062990)

[Rysunek 8. Kod metody generującej odpowiedź chatbota ‘Refi’ 35](#_Toc137062991)

[Rysunek 9. Kod metody sprawdzającej poprawność danych logowania 37](#_Toc137062992)

[Rysunek 10. Pytania i odpowiedzi dotyczące artykułu Czas trwania zawodów 39](#_Toc137062993)

[Rysunek 11. Pytania i odpowiedzi dotyczące artykułu Wrzut 40](#_Toc137062994)

Spis tabel

[Tabela 1. Najważniejsze elementy języka AIML 25](#_Toc137062023)

[Tabela 2. Wyniki testów wydajności chatbota 'Refi' 41](#_Toc137062024)

[Tabela 3. Wyniki testów użyteczności programu konwersacyjnego 'Refi' 42](#_Toc137062025)