

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRYCZNY
INSTYTUT STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ

PRACA DYPLOMOWA MAGISTERSKA
na kierunku ELEKTROTECHNIKA



Aleksandra MUĆKO

Nr albumu: 197729

Rok. akad. 2009/2010
Warszawa, 14.10.2009

**ANTROPOMORFIZACJA SYSTEMÓW SZTUCZNEGO
ROZMÓWCY**

Zakres pracy:

1. *Wstęp i przedstawienie celu pracy*
2. *Przegląd i analiza istniejących systemów i języków sztucznego rozmówcy*
3. *Projekt własnego programu sztucznego rozmówcy realizującego przyjęte założenia*
4. *Implementacja i testy programu*
5. *Podsumowanie i wnioski*

Kierujący pracą: dr inż. Witold Czajewski

A handwritten signature in purple ink, appearing to read 'WITOLD CZAJEWSKI', written over a horizontal line.

A handwritten signature in blue ink, appearing to read 'P. Belczyk', written in a cursive style.

Konsultant:

Termin wykonania: 15.09.2010

Praca wykonana i zaliczona pozostaje
własnością Instytutu i nie będzie
zwrócona wykonawcy

Antropomorfizacja systemów sztucznego rozmówcy

Streszczenie

Celem niniejszej pracy dyplomowej było utworzenie zbioru podstawowych reguł oraz bazy danych, które nadałyby chatterbotowi pewną losowość jak i płynność w rozmowie, tworząc z niego imitację istoty myślącej i świadomej. Podjęta została próba eliminacji podstawowych błędów, jakie w komunikacji z człowiekiem popełnia maszyna.

W rozdziale pierwszym scharakteryzowano przedmiot pracy dyplomowej oraz wyszczególniono jej cele. Rozdział drugi poświęcono na zapoznanie się z występującymi w przedmiocie pracy problemami i istniejącymi rozwiązaniami oraz na przedstawienie znanych programów.

Rozdział trzeci zawiera krótki opis użytej platformy i wprowadzenie do języka AIML, którymi autor pracy się posługiwał. Znajdują się w nim stworzone przez autora rozwiązania przedstawionych problemów. Wyjaśniono tam również metody postępowania oraz zilustrowano działanie programu. Wprowadzone do programu zostały różne rozwiązania odwołań do pamięci rozmowy oraz nastawienia chatterbota do rozmówcy. Charakterystyczne dla automatów skończonych dyskretne przechodzenie między stanami zostało w chatterbotcie Loki zrealizowane za pomocą etykiet tematu oraz licznych wewnętrznych parametrów. Stworzony został za pomocą redukcji ciąg zależności pomiędzy formą i treścią wprowadzanego tekstu a zwracaną odpowiedzią.

Autor pracy udostępnił stworzony od podstaw program w sieci na stronie <http://www.pandorabots.com/pandora/talk?botid=8a77cf257e341167> w celu jego przetestowania. Rozdział czwarty poświęcono na przedstawienie wyników testów chatterbota w różnych warunkach i porównanie go z innymi tego typu programami. Należy podkreślić, że porównywane programy opracowane były na różne sposoby (różne rozwiązania programistyczne i silniki) i posiadały różne doświadczenie.

Rozdział piąty zawiera podsumowanie pracy oraz wnioski końcowe wynikające z przeanalizowanych zagadnień. Dopełnienie pracy stanowią załączniki, w których zamieszczono bazę danych chatbota. Stworzona przez autora pracy baza danych programu składa się z 32 plików AIML, w których zawarte jest ponad 30 tysięcy kategorii redukcji i przekierowań, ponad 10 tysięcy nowych kategorii zwracających odpowiedzi, liczne nowe

kategorie o specjalnym przeznaczeniu oraz część kategorii zapożyczonych z plików AAA. Autor stworzył także stronę graficzną chatterbota Loki.

Autor przetestował działanie chatterbota Loki za pomocą licznych rozmów. Badał i porównywał jego reakcje względem innych chatbotów. Rozmowy przeprowadzone były zarówno z ludźmi jak i z innymi chatbotami zamieszczonymi na tej samej oraz innych platformach chatterbotów.

Anthropomorphization of Artificial Conversational Entities

Summary

The purpose of this master thesis was to create a set of basic rules and a knowledge base, which were used for a smooth conversation with random characteristics as chatterbot imitated a self aware and thinking entity. An attempt has been made to solve fundamental errors of machines which occur during communication with humans.

In the first chapter the subject and purpose of this thesis has been presented. The second chapter contains the description of problems found in chatterbots and solutions to them as well as the presentation of well known programs in this field of study.

The third chapter contains a brief description of hosting webservice and an introduction to the AIML language utilized by the author of this thesis. It includes developed by the author solutions for presented problems. The methodology as well as program's operation has been explained in detail. Various solutions for preservation of conversation's memory were incorporated into the program as well as it's basic attitudes towards the client. Chatterbot Loki implemented the discrete state transitions, which is found in state machines, through topic tags and numerous inner parameters. Series of interrelationships between form and content of input sentence and output response has been created by the use of reduction patterns.

In order to test the program created solely by the author of this thesis, it has been published on the internet under the address **<http://www.pandorabots.com/pandora/talk?botid=8a77cf257e341167>**. The fourth chapter is dedicated to analysis of test results upon different conditions and it's comparison with other programs. It should be noted that compared programs utilized various methods of operation (different software solutions and hosting methods) and varied in their experience.

The fifth concluding chapter summarizes the work done by the author and presents final conclusion to analyzed problems. The knowledge base of chatbot Loki has been attached on CD to this thesis. Created by the author knowledge base consist of 32 AIML files containing over 30 thousand reduction and redirection categories, over 10 thousand new response categories, various special categories as well as some selected and modified AAA (Annotated

A.L.I.C.E. AIML) categories. Furthermore the author created the chatterbot's homepage with it's unique layout.

Chattbot Loki has been tested by the author through numerous conversations. It's responses were closely examined and compared with other chatbots. Not only humans but other chatterbots, which utilize the same as well as various other methods of operation, were participating in it's conversations.

Oświadczenie autora pracy dyplomowej magisterskiej.

Świadoma odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca dyplomowa nie była wcześniej przedmiotem postępowania związanego z uzyskaniem tytułu zawodowego w uczelni wyższej. Niniejsza praca jest utworem zbiorowym i stanowi własność intelektualną osoby kierującej pracą oraz moją. Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Zobowiązuję się, że nie wykorzystam ani nie opublikuję wyników pracy bez zgody osoby kierującej pracą oraz kierownika Jednostki, w której pracę wykonano.

Aleksandra Mućko
.....

Aleksandra Mućko

85112716144
.....

PESEL

Spis treści

Wykaz ważniejszych oznaczeń i skrótów	2
1 Wstęp.....	3
1.1 Wprowadzenie.....	3
1.2 Cel i zakres pracy	4
2 Przegląd istniejących systemów i języków sztucznego rozmówcy.....	5
2.1 Problemy występujące w programach sztucznego rozmówcy.....	6
2.2 Metody uczenia sztucznego rozmówcy	10
2.3 Język AIML	12
2.4 Chatterbot Jabberwacky	12
2.5 Aplikacja FreeHal	12
2.6 Platforma MyCyberTwin	13
2.7 Program Verbot.....	13
3 Opracowanie własnego programu sztucznego rozmówcy realizującego przyjęte założenia	14
3.1 Platforma Pandorabots.com	14
3.2 Struktura języka AIML	19
3.3 Rozpoznawanie tonu rozmówcy	37
3.4 Rozumienie znaczenia zdań	46
3.5 Pamięć przeprowadzonej rozmowy	56
3.6 Różnorodność odpowiedzi	71
4 Testy programu.....	76
4.1 Porównanie programu Loki z Alice-Botami	77
4.2 Porównanie programu Loki z innymi chatterbotami	84
4.3 Rozmowa z ludźmi.....	89
5 Wnioski	100
5.1 Budowa bazy danych	100
5.2 Obserwacja.....	101
5.3 Przemyślenia	102
Bibliografia	104

Wykaz ważniejszych oznaczeń i skrótów

AIML	- <i>Artificial Intelligence Markup Language</i> , Język Znaczników Sztucznej Inteligencji
A.L.I.C.E	- nazwa chatterbota, <i>Artificial Linguistic Internet Computer Entity</i>
botmaster	- programista chatterbota
chatterbot, chatbot	- program sztucznego rozmówcy
GPL	- <i>General Public License</i> , licencja wolnego i otwartego oprogramowania
HMM	- <i>Hidden Markov Model</i> , Ukryte Modele Markowa
Loki	- nazwa chatterbota stworzonego przez autora pracy
NLP	- <i>Natural Language Processing</i> , Przetwarzanie Języka Naturalnego
PSAE	- <i>Pattern-side AIML expressions</i> , wyrażenia AIML po stronie wzorca
SDK	- <i>Software Development Kit</i>
SI	- Sztuczna Inteligencja
TSAE	- <i>Template-side AIML Expression</i> , wyrażenia AIML po stronie szablonu
XML	- <i>Extensible Markup Language</i> , Rozszerzalny Język Znaczników

1 Wstęp

1.1 Wprowadzenie

Rosnące znaczenie komputerów w codziennym życiu człowieka oraz ich szybki rozwój sprawił, że od lat podejmowano próby nadania im cech ludzkich. Realizacja czegoś, co jest niezwykle popularnym wątkiem w literaturze fantastyki naukowej (ang. *science-fiction*), z roku na rok zdaje się być coraz bardziej prawdopodobna. Bez względu na to, czy próby te podejmowane są w celach rozrywkowych, edukacyjnych, czy też aby uprościć i zwiększyć współpracę między maszyną a człowiekiem, ważnym ich elementem jest dialog. Najprostszą jego formą jest realizacja zadanych komend. Człowiek ma w swej naturze personifikację przedmiotów, zwierząt oraz zjawisk. Przy dłuższej pracy z jakimś przedmiotem zaczyna mu on nadawać coraz więcej cech ludzkich. Niczym niezwykłym nie jest więc rozmowa z urządzeniem, a jest to wręcz często spotykane. Także my sami mamy w zwyczaju robić to nieświadomie.

Chatterbot (bądź też *chatbot*) jest programem, którego podstawowym zadaniem jest prowadzenie dialogu z człowiekiem. Wraz z rozwojem techniki mikroprocesorowej ilość przetwarzanych informacji oraz prędkość przeprowadzanych operacji przez komputer rośnie, umożliwiając przez to próbę odtworzenia toku myślenia człowieka. Od najprostszych systemów eksperckich, po bardziej zaawansowane sieci neuronowe, zadaniem takiego programu zawsze jest znalezienie najlepszego rozwiązania - w tym zaś przypadku, odpowiedzi.

Antropomorfizacja programu sztucznego rozmówcy jest niczym innym jak próbą uczłowieczenia go. Oznacza to, iż pragniemy prowadzić dialog nie z maszyną o ściśle określonych i ograniczonych odpowiedziach, lecz z czymś co zachowuje się jak człowiek i ma swój własny, indywidualny charakter. Coś co jest jednak naturalne dla człowieka, dla komputera będzie niezgodne z jego działaniem, z góry uporządkowanym i logicznym. Próba analizy ludzkich zachowań i odpowiedzi oraz ich losowości może w najlepszym stopniu zakończyć się na wielkiej liczbie podstawowych reguł, które w niektórych przypadkach będą ze sobą sprzeczne. Poprzez antropomorfizację programu można więc widzieć nic innego jak próbę wprowadzenia chaosu w uporządkowane środowisko. Chaos ten jednak musi działać zgodnie z pewnymi, określonymi regułami, tworząc spójną, mającą sens całość.

1.2 Cel i zakres pracy

Niniejsza praca poświęcona jest odtworzeniu ludzkich zachowań u chatterbotów. Programy tego typu stosowane są coraz częściej w celach:

- informacyjnych jako wirtualni agenci i asystenci,
- edukacyjnych jako nauczyciele,
- rozrywkowych.

O ich wykorzystaniu decydują liczne zalety takie jak: łatwa dostępność, obsługa wielu klientów, łatwa rozbudowa bazy wiedzy. Niestety wymagają one stałego nadzoru, często nie radzą sobie z abstrakcyjnym myśleniem oraz nie są wolne od błędów. Literatura z nimi związana nie jest obszerna i występuje przeważnie w języku angielskim.

Celem niniejszej pracy dyplomowej jest utworzenie zbioru podstawowych reguł oraz bazy danych, które nadadzą chatterbotowi pewną losowość jak i płynność w rozmowie, tworząc z niego imitację istoty myślącej i świadomej. Podjęta zostanie przy tym próba eliminacji podstawowych błędów, jakie w komunikacji z człowiekiem popełnia maszyna.

Zakres pracy obejmuje:

- stworzenie bazy danych programu,
- rozwiązanie wybranych problemów związanych z programem,
- testy programu.

W rozdziale pierwszym scharakteryzowano przedmiot pracy dyplomowej oraz wyszczególniono jej cele. Rozdział drugi poświęcono na zapoznanie się z występującymi w przedmiocie pracy problemami i istniejącymi rozwiązaniami oraz na przedstawienie znanych programów.

Rozdział trzeci zawiera krótki opis użytej platformy i wprowadzenie do języka AIML, którymi autor pracy się posługiwał. Znajdują się w nim stworzone przez autora rozwiązania przedstawionych problemów. Wyjaśniono tam również metody postępowania oraz zilustrowano działanie programu.

Rozdział czwarty poświęcono na przedstawienie wyników testów stworzonego od podstaw przez autora pracy dyplomowej chatterbota w różnych warunkach i porównanie go z innymi tego typu programami. Należy podkreślić, że porównywane programy opracowane były na różne sposoby (różne rozwiązania programistyczne i silniki) i posiadały różne doświadczenie. Rozdział piąty zawiera podsumowanie pracy oraz wnioski końcowe wynikające z przeanalizowanych zagadnień. Dopełnienie pracy stanowią załączniki, w których zamieszczono bazę danych chatbota.

2 Przegląd istniejących systemów i języków sztucznego rozmówcy

Idea interakcji człowieka z komputerem na wzór komunikacji międzyludzkiej powstała wraz z pojawieniem się pierwszych komputerów. Wielu pisarzy, takich jak Stanisław Lem, Philip K. Dick czy Isaac Asimov nawiązywało do tego tematu, opierając się jedynie na własnej wyobraźni. Temat ten jednak nie był traktowany zbyt poważnie. Za ojca istniejących dziś rozwiniętych systemów sztucznego rozmówcy uznać można Alana Turinga. W opublikowanym w 1950 roku artykule „*Computing Machinery and Intelligence*” przez brytyjskie czasopismo *Mind*, przewidywał on gwałtowny rozwój komputerów, stawiając przy tym pytania rozważające ucłowieczenie maszyny oraz stworzył założenia tak zwanego "testu Turinga" [12].

Narodzinom sztucznej inteligencji jako dziedziny nauki towarzyszyło pojawienie się pierwszych systemów eksperckich. Jednym z nich była ELIZA Josepha Weizenbauma, system interaktywnego dialogu naśladowujący psychoterapeutę.

Eliza była pierwszym chatterbotem, który działał na zasadzie prostego rozpoznawania wzorców i modelu stymulowanej odpowiedzi. Wyszukiwała ona słowa kluczowe bądź związki wyrażen zwracając przypisany bezpośrednio do nich tekst. Takie działanie zwie się dziś „mechanizmem ELIZY”, gdyż jest często wykorzystywane w obecnych chatterbotach jako jeden ze sposobów wyszukiwania odpowiedzi.

Dostrzegając reakcje ludzi (zwłaszcza osób nie mających większego kontaktu z komputerami), wykraczające poza jego przewidywania, Weizenbaum zaczął się obawiać, iż świat nie był przygotowany na programy tego typu. Zamknął on projekt ELIZY i opublikował w 1972 roku książkę „*Computer Power and Human Reason*” mając na celu edukację ludzi i przy tym atakując sztuczną inteligencję i ELIZĘ. Spowolniło to rozwój tego typu programów [13].

Obecnie powstaje coraz więcej chatterbotów, ich języków programowania i ogólnie dostępnych platform do tworzenia chatbotów (PandoraBot, MyCyberTwin). Przeważająca większość zaawansowanych chatbotów wykorzystywana jest przez firmy jako wirtualni agenci, asystenci. Prowadzone są liczne projekty mające na celu stworzenie programu interaktywnego rozmówcy w celach edukacyjnych. Najpopularniejszym przykładem spośród

nich są programy do praktyki języków obcych (CSIEC, CLIVE) [14].

Przez ostatnie lata powstały tysiące chatterbotów opartych na ogólnie dostępnych silnikach. Tworzenie systemów sztucznego rozmówcy stało się bardzo popularne wśród normalnych internautów, zwłaszcza jeśli udostępnione im portale zawierają proste narzędzia do ich kreacji.

Co roku organizowane są międzynarodowe konkursy (Loebner Prize Competition, Chatterbot Challenge), mające na celu znalezienie takiego programu, który nie tylko naśladuje człowieka, lecz jest w stanie przekonać swojego rozmówcę, iż ten ma do czynienia z człowiekiem, a nie zaprogramowaną dobrze iluzją.

2.1 Problemy występujące w programach sztucznego rozmówcy

Pomimo, iż technika rozwija się każdego dnia, problemy jakie występują w chatterbotach pozostają w większości niezmiennie [8]. Tym bardziej, gdy wymagania programu rosną i opierają się o pojęcia abstrakcyjne i semantykę.

Tworząc chatterbota trzeba pamiętać, iż ma się do czynienia z maszyną o ograniczonej pamięci i możliwościach, podczas gdy człowiek cały czas rozwija się i działa często na niejasnych, a czasem i pozbawionych logiki licznym skojarzeniach i skrótach.

Umysł nie działa jak prosty program komputerowy. Posiada on wiele różnych wymiarów, których brakuje maszynie, bądź których zaimplementowanie byłoby trudne i niezwykle czasochłonne. Rozumienie, wiara, poczucie wagi, piękna, zrozumienie dobra i zła, emocje, wyobrażenia, percepcja - to tylko część rzeczy jakimi kieruje się człowiek [9].

Szybkość odpowiedzi

Szybka odpowiedź programu w większości przypadków jest jego atutem. Zbyt wielka prędkość reakcji może być jednak niepożądana, jeśli celem programu jest naśladowanie człowieka i oszukanie swego rozmówcy.

Podobnym problemem może być zbyt wolna odpowiedź na proste wyrażenie. W niektórych przypadkach chatterbot będzie wyszukiwał zbyt długo poprawnego responsu na coś, co jest dla nas czymś oczywistym.

Pozbawiony rozumienia prostej semantyki i związków między poszczególnymi słowami, opierać musi się na wielkiej liczbie testów i założeń logicznych, wpisanych w jego program.

Powtarzalność odpowiedzi

Ze względu na małą pamięć i ograniczoną liczbę odpowiedzi, częstym problemem chatterbotów jest wielokrotne użycie tego samego responsu do wspomnianych po raz kolejny, choć w innym kontekście tych samych wyrażen, bądź przy zwrotach o podobnym znaczeniu, lecz przypisanym im odnośniku do tego samego słowa kluczowego.

Program w najlepszym przypadku pamięta tylko kilka wcześniejszych kroków. Traci on przez to najczęściej podmiot rozmowy i jej historię sprzed kilku minut, czy sekund.

Brak większej losowości sprawia, iż odpowiedzi coraz częściej się powtarzają i rozmówca dostrzega gołym okiem ograniczenia programu.

Ograniczony zakres tematyczny

Chatterboty są najczęściej ukierunkowane na określoną tematykę, co umożliwia prowadzenie przez nie w miarę długich i inteligentnych rozmów. Gdy jednak wykracza poza swoje "terytorium", od razu gubią się i zdradzają swemu rozmówcy.

Widoczne jest to w chatterbotach opartych na najprostszyc systemach eksperckich. Botmaster nie jest w stanie przewidzieć i przygotować chatterbota do rozmów z ludźmi, którzy często odwołują się do skojarzeń, bądź porównań.

Jeśli program nie jest w stanie sam skierować rozmowy na sprzyjające mu tematy, to rozmowa może przybrać wygląd niezrozumiałej wymiany zdań.

Rozpoznawanie błędów

Człowiek często popełnia błędy takie jak przestawienie, brak bądź użycie innych liter. Niektórzy zaś korzystają z dziwnych skrótów i znaków, które weszły w życie w kulturze młodych użytkowników Internetu. To wszystko stanowi zbiór nierozpoznawalnych nowych wyrażen dla chatterbota. Ludzie są w stanie rozpoznać jakiś wyraz tylko po kontekście całego zdania i pomimo wszystkich liter przestawionych, jeśli tylko pierwsza i ostatnia są prawidłowe.

Świadome bądź przypadkowe stawianie błędów i zwrotów subkultury zmieni najczęściej proste wyrażenie w niezrozumiały zwrot, który zakłóci płynność rozmowy z chatterbotem. Człowiek z góry oczekuje, że będzie zrozumiały, oraz że winę za brak zrozumienia ponosi druga osoba (rozmówca).

Zwroty grzecznościowe

Prostą do zauważenia u chatterbotów jest ich nadmierna aktywność w przypadku powitań, pożegnań czy choćby zwykłych prośb czy podziękowań. Najbardziej widoczne są

sytuacje, gdzie program sam wywołuje u rozmówcy słowa tego typu, po czym reaguje na te słowa kluczowe, zgodnie z ustaloną formułą.

W przypadku pożegnań chatterbot, z powodu swego funkcjonowania na zasadzie "tur", zawsze będzie domyślnie dążyć do "osiągnięcia ostatniego zdania".

Perfekcyjność odpowiedzi

Chatterbot ma za zadanie imitować ludzką rozmowę. Ponieważ odwołuje się on bezpośrednio do wcześniej przygotowanych odpowiedzi, brak mu ludzkiej kreatywności, improwizowania oraz ludzkich błędów takich jak pomyłone słowa, litery, bądź zapomnienie jakiejś informacji.

Prowadzi to do powstania zarówno długich, jak i bogatych w masę informacji odpowiedzi. Gdy ma się do czynienia z wirtualnym doradcą bądź innego rodzaju asystentem, którego celem jest dostarczenie poszukiwanych i szczegółowych informacji, jest to cecha pożądana. Gdy jednak głównym zadaniem chatterbota ma być stworzenie iluzji człowieka, jest to wada wynikająca z błędu botmastera bądź źle dopracowanego sposobu wyszukiwania informacji w internecie.

Zaprzeczanie własnym słowom

Większość chatterbotów zarządza bardzo ograniczoną pamięcią podczas rozmowy. Reagując na słowa kluczowe, zwracają one przypisane wcześniej odpowiedzi, nie mające najczęściej związku z wyrażeniem wypowiedzianym kilka odpowiedzi wcześniej. Prowadzić to może do sytuacji, gdzie chatterbot zaprzecza wypowiedzianemu wcześniej stwierdzeniu.

Dzieje się tak głównie z powodu ograniczonych możliwości pamięci oraz małej bądź źle zorganizowanej bazy danych. Botmaster najczęściej znajduje takie sytuacje dopiero po ich wystąpieniu w logu jednej z rozmów.

Przesadność informacji

Większość ludzi nie lubi się rozpisywać i ogranicza się do wystarczającego minimum. Chatterboty korzystając z zapisanych w swej bazie danych informacji, bądź wyszukując ich w internecie, mogą przesadzić z ich ilością.

Nie tylko będzie to zniechęcać rozmówcę, zmuszając go do nieustannego czytania długich "wypocin", lecz może spowodować zatracenie głównego wątku rozmowy.

Program będzie sprawiał wrażenie, jakby próbował narzucić człowiekowi swoje "zdanie", bądź wygłaszał długie referaty czy pouczenia.

Duża liczba zdań w odpowiedzi nie sprawi, iż chatterbot wyda się bardziej ludzki.

Zmieni to go w raczej dość nachalny program.

Można więc powiedzieć, że spośród wszystkich cech człowieka, botmaster przypisze mu jedne z tych najgorszych.

Zapętlenie

Jednym z większych problemów dla chatterbotów jest odpowiedź na pytanie nie zawierające podmiotu, lecz odwołujące się do wcześniejszych zdań w rozmowie. Przykładami takimi są "why", "who", "when" itd. Stosowane wielokrotnie jako pojedyncza wypowiedź, wymagają od chatterbota, by sam się domyślił, o co tak naprawdę chodzi jego rozmówcy.

Liczba kombinacji i możliwości w jakich te ogólnikowe pytania się pojawiają jest tak wielka, iż niemożliwe jest ich przewidzenie oraz całkowite zabezpieczenie się przed nimi.

Jednym problemem jest brak znalezienia podmiotu w takim pytaniu, a drugim powielenie tego samego pytania wielokrotnie. Niczym małe dziecko, rozmówca może zadawać przesadną liczbę zwrotów typu "why?". Jeśli chatterbot będzie miał przypisaną jedną bądź kilka odpowiedzi pojawiających się na przemian, to może nigdy nie wyjść z pętli, gdyż jej bez odpowiedniego założenia nawet nie rozpozna.

Rozmowa z wieloma osobami

Chatterbot prowadzi swoją rozmowę poprzez prostą wymianę "tur" ze swoim rozmówcą. Zapewnia to płynność i ciągłość interakcji z człowiekiem.

Najczęściej środowisko w jakim ulokowany jest chatterbot ogranicza liczbę osób, z którymi się porozumiewa, do jednej. Problem pojawi się jednak, gdy znajdzie się w sytuacji, gdzie liczba rozmówców wzrośnie.

Chatterbot nie jest w stanie rozpoznać, czy konkretna osoba zwraca się do niego, czy też do innego z rozmówców. By coś takiego było możliwe, musiałby otrzymywać za każdym razem wyraźny znak. Ponieważ ludzie rozmawiają najczęściej w sposób spontaniczny i odczytują część informacji z kontekstu, wprowadzenie tego typu znaku byłoby raczej niemożliwe.

Chatterbot próbowałby udzielić odpowiedzi na każde zdanie jakie padłoby ze strony jego rozmówców.

Duża baza danych

Minimalizacja oraz uproszczenie kodu są powszechnie stosowane w programowaniu. Założeniem chatterbotów jest jednak ciągły rozwój i poszerzanie swej bazy wiedzy. Stąd też za bardziej zaawansowane można uznać te programy, które są bardziej rozbudowane.

Poszerza to ich zakres tematyczny oraz zapewnia większą możliwość znalezienia dobrej bądź bardziej losowej odpowiedzi. Wraz z ogromnym rozbudowaniem bazy danych pojawiają się jednak problemy, takie jak: zbyt długie wyszukiwanie, powtórzenia odpowiedzi bądź powstanie większej ilości sprzecznych informacji. Zależnie od długości i sformułowania pytania, znalezienie poprawnej odpowiedzi może zająć mu o wiele więcej czasu niż człowiekowi.

Rozumienie prostej semantyki

Większość chatterbotów opiera się na mechanizmie ELIZY. Reagują one przez to na słowa kluczowe, nie będąc w stanie dostrzec, iż zdanie mogło zostać sformułowane w sposób podchwytliwy [6].

Pomimo, iż chatterbot jest w stanie odpowiedzieć na długie, trudne zdania, to oparte na elementarnej, dziecięcej logice wypowiedzi sprawiają mu trudność.

"if London is south of Oxford, is Oxford north to London?"

Operując samymi pustymi pojęciami, bez wiedzy na temat ogólnego zastosowania danej rzeczy, program nie jest w stanie rozpoznać podchwytliwych pytań i zareagować na nie na wzór człowieka.

"I have a jewellery box in my hand, how many CDs can I store in it"

Spotykając się z metaforą, chatterbot ma do czynienia z wyrażeniami normalnie ze sobą nie funkcjonującymi. Jeśli nawet posiadałby podstawowe informacje ich dotyczące, to nie byłby w stanie zrozumieć znaczenia danego zestawienia.

"how does the colour red smell?"

Im bardziej odpowiedź na pytanie zależy od zrozumienia przez rozmówcę jego znaczenia, kontekstu lub domysłu, tym sztuczny rozmówca ma mniejsze szanse na zwrócenie odpowiedniej riposty.

2.2 Metody uczłowiczenia sztucznego rozmówcy

Doskonalenie chatterbota jest niczym innym jak dopracowanym coraz bardziej oszustwem. Celem testu Turinga oraz botmasterów nie jest tak naprawdę stworzenie myślącego sztucznego rozmówcy, lecz programu, którego zadaniem jest wmówienie swemu rozmówcy, iż ma on do czynienia z człowiekiem bądź po prostu istotą myślącą. W tym celu podejmuje się kroki zabezpieczające przed "sztuczną głupotą" programu, oraz próbuje stworzyć losowe "ludzkie błędy" w chatterbocie [3].

Wypowiedzi chatterbota powinny być ograniczone. Naśladowanie ludzkiego lenistwa i minimalizmu podstawowych informacji daje możliwość płynniejszej i dłuższej rozmowie. W celu stworzenia iluzji "pisania", często wprowadza się do programu opóźnienia odpowiedzi. Wiadomości w chatterbotach biorących obecnie udział w konkursach spełniają z reguły kilka jednakowych założeń:

- * wydłużony czas odpowiedzi,
- * minimalność przekazanej informacji,
- * losowe błędy pisowni,
- * losowość odpowiedzi na podobne pytanie.

Wymagane jest także zabezpieczenie chatterbotów przed niepożądanymi zdarzeniami i błędami.

Bardzo często program napotyka podczas rozmowy na problem nierozpoznawalnego pytania. Może ono być błędnie sformułowane, choć w większości przypadków będzie to informacja o obcej dla chatterbota treści, budowie i przesłaniu. W celu bezpiecznej rozmowy, wprowadza się specjalny typ odpowiedzi, który ma reagować na tego rodzaju sytuacje. Może być on ogólnie działający we wszystkich przypadkach, choć częściej spodziewać się można kilku tego typu odpowiedzi, uzależnionych wewnątrz od przeprowadzonej rozmowy, bądź znajomego pojedynczego słowa-klucza. Chatterboty w takich sytuacjach działają najczęściej na jeden z trzech sposobów:

- * informacją o braku odpowiedzi na pytanie,
- * losową odpowiedzią na dany temat, wybraną z zawartej w programie listy,
- * losową odpowiedzią zmieniającą temat rozmowy, z zawartej w programie listy.

Innym zagadnieniem jest wchodzenie programu w pętle i zwracanie tej samej odpowiedzi. By temu zapobiec, tworzy się w chatterbocie specjalne, uwarunkowane odpowiedzi, w których na początku badane są wcześniej wprowadzone przez rozmówcę wiadomości. Mimo ograniczonych możliwości programu sztucznego rozmówcy, odwołanie się do dwóch czy trzech informacji wejściowych nie sprawia wielkiego problemu. Dopracowanie osobnych odpowiedzi do niektórych zdarzeń może jednak być czasochłonne dla botmastera.

2.3 Język AIML

AIML (*Artificial Intelligence Markup Language*) to język programowania oparty o język XML. Stworzony przez dr Richarda Wallace'a rozwija się od 1995 roku i służy do implementowania wiedzy chatterbota. Ponieważ AIML udostępniony został jako darmowe oprogramowanie GNU GPL (*General Public License*), podobnie jak Linux, jest on ciągle rozwijany i ulepszany przez programistów na całym świecie [15].

W 1995 roku, bazując na ELIZIE, została stworzona A.L.I.C.E. (*Artificial Linguistic Internet Computer Entity*), która trzykrotnie zdobyła nagrodę Loebnera.

2.4 Chatterbot Jabberwacky

Jabberwacky jest chatterbotem, którego celem jest symulacja naturalnej ludzkiej rozmowy w interesujący i rozrywkowy sposób. Jabberwacky uczy się cały czas przez rozmowy z ludźmi. Projekt, autorstwa Rollo Carpentera [18], powstał w 1988 roku i został wpuszczony do sieci w roku 1997.

Jabberwacky przechowuje wszystkie stoczone rozmowy i spośród nich wybiera odpowiedź o najbardziej zbliżonym kontekście. Innymi słowy, w trakcie rozmowy korzysta z posiadanej bazy wiedzy, rozbudowując ją jednocześnie o to, co powiedział jego rozmówca. Bez żadnych stałych reguł, wykorzystuje on wyłącznie zasady sprzężenia zwrotnego (ang. *feedback*). To wyróżnia go spośród innych chatterbotów opartych o reguły botmastera.

Uczenie się wyłącznie za pomocą przeprowadzonych konwersacji daje chatterbotowi ogólną wiedzę oraz wzbogaca jego charakter. To czyni go jednak dla przedsiębiorstw, które cenią sobie kontrolę nad kierunkiem rozmowy, niestosownym w użyciu. By móc wykorzystać go w biznesie należy użyć specjalnego zestawu narzędzi do tworzenia skryptów umożliwiających programowi tworzenie drzew wielokrotnego wyboru, pętle, powroty, generację odpowiedzi na wewnętrzne fragmenty zdania – wykorzystuje on przy tym logikę rozmytą i zasady dopasowania do wzorców.

2.5 Aplikacja FreeHal

FreeHal nazywamy projekt jednej z bardziej rozwiniętych aplikacji samouczących sztucznej inteligencji rozpoczęty w 2006 roku przez Tobiasa Schulza [17]. Ma on na celu jak najbliższą imitację rozmowy międzyludzkiej.

Wykorzystuje on przede wszystkim sieci semantyczne, opisujące relacje pomiędzy pojęciami, zaawansowane rozpoznawanie wzorców, redukcje wyrazów do rdzenia, parsery zdań* i klasyfikację części mowy. Wiedza chatbota jest zawarta w sieciach semantycznych. Bazy danych przechowujące części mowy wykorzystywane są do zrozumienia wypowiedzi rozmówcy. FreeHal działa w oparciu o HMM (*Hidden Markov Model*). Obecnie aplikacja obsługuje języki angielski i niemiecki.

2.6 Platforma MyCyberTwin

MyCyberTwin to platforma SI do tworzenia chatterbotów (tzw. *CyberTwin*) stworzona przez firmę MyCyberTwin w 2005 roku i udostępnioną w sieci w 2007 roku [19]. Będąc dostępną bezpośrednio przez Internet, umożliwia szybkie tworzenie chatbota.

Interfejs oferuje wybór osobowości, wiedzy i składniki tematów. Umożliwia programowanie chatterbota w prosty i nie wymagający żadnej wiedzy technicznej sposób. Platforma opiera się o różne algorytmy wyszukiwania i segregacji, konwersacyjną bazę wiedzy, opierającą się na regułach heurystykę i techniki nauczania. Wszystkie rozmowy z chatterbotem są zapisywane i dostępne do podglądu przez botmastera.

2.7 Program Verbot

Verbot (*Verbal-Robot*) jest popularnym programem SDK (*Software Development Kit*) Sztucznej Inteligencji [20]. Jednym z prototypów chatterbotów opartym na nim jest Julia dr Michaela Maudina, obecna w internecie w różnych formach już od 1989 roku.

Stworzony przez Conversive Inc, Verbot (zwany także *Verbally Enchanted Software Robot*) wykorzystuje technologię Microsoft Agent i specjalizuje się w NLP (*Natural Language Processing*, przetwarzanie języka naturalnego). Na chwilę obecną program pracuje wyłącznie pod Microsoft Windows i opiera się w głównej mierze o język C#.

Verboty wykorzystują język naturalny, sztuczną inteligencję oraz animację w czasie rzeczywistym z syntezą głosu w celu stworzenia wirtualnej osobowości. Ich architektura oparta jest o XML i .NET, a dzięki wbudowanemu edytorowi są przyjazne dla użytkownika.

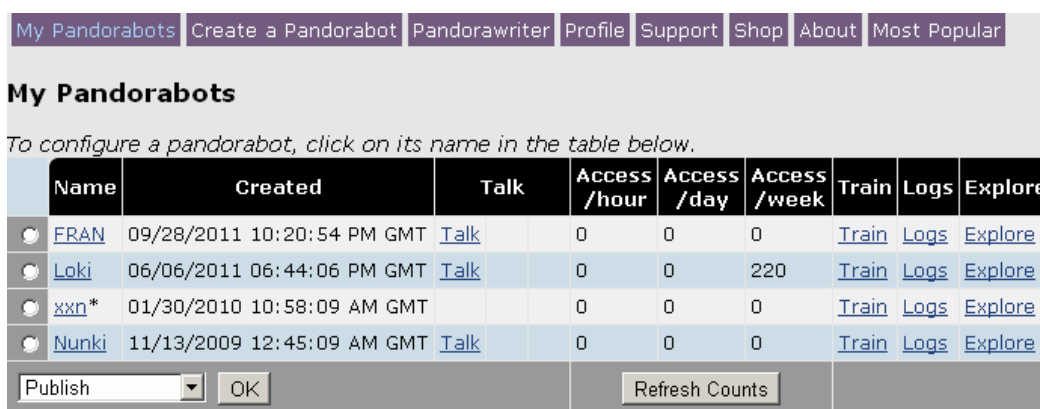
Standardowy pakiet zawiera dwie aplikacje, Verbot Player i Verbot Scripting Interface (edytor). Verbot Player ładuje bazy danych i wyświetla interaktywną animowaną postać, która porusza się podczas rozmowy. Verbot Editor wykorzystywany jest do tworzenia baz danych przy użyciu prostego interfejsu.

* Parser zdania - analiza składni zdania

3 Opracowanie własnego programu sztucznego rozmówcy realizującego przyjęte założenia

3.1 Platforma Pandorabots.com [21]

Pandorabots.com jest darmowym portalem, który umożliwia zamieszczenie stworzonych baz wiedzy w formacie AIML i publikację chatbota w Internecie. Jest to największy tego typu portal, gdyż zrzesza ponad sto tysięcy botmasterów i setki tysięcy pandorobotów stworzonych w różnych językach. Przy pomocy każdego typu przeglądarki internetowej można bezpośrednio ze strony wprowadzać zmiany w chatbotach.



	Name	Created	Talk	Access /hour	Access /day	Access /week	Train	Logs	Explore
<input type="radio"/>	FRAN	09/28/2011 10:20:54 PM GMT	Talk	0	0	0	Train	Logs	Explore
<input type="radio"/>	Loki	06/06/2011 06:44:06 PM GMT	Talk	0	0	220	Train	Logs	Explore
<input type="radio"/>	xxn*	01/30/2010 10:58:09 AM GMT		0	0	0	Train	Logs	Explore
<input type="radio"/>	Nunki	11/13/2009 12:45:09 AM GMT	Talk	0	0	0	Train	Logs	Explore

Rys. 3.1.1. Panel główny botmastera

Botmaster ma możliwość tworzenia wielu pandorobotów. Są one wyświetlone na stronie startowej konta botmastera (rys. 3.1.1). Zawarte są tam informacje na temat liczby odwiedzin i rozmów chatbotów. W celu edycji któregoś z pandorobotów należy naklikać na jego nazwę. Po wprowadzonych zmianach chatbota należy go "opublikować", aby go zaktualizować. Boty, które nie zostały zaktualizowane (mają znak przy swojej nazwie: xxn*), nie rozpoznają ostatnio wprowadzonych modyfikacji.

Create a Pandorabot

Name:

- automatically discover spaces between words (suggested for Japanese)

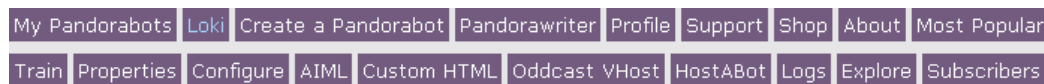
Startup AIML - Please select the AIML set you wish to include:

- Standard AIML - Sept 2001
 - Dr Wallace's A.L.I.C.E - March 2002
 - Annotated A.L.I.C.E. AIML - September 2003
 - Christian Drossmann's standalone German AIML - January 2002
 - Sandro Pons' Italian AIML set - April 2001

- No initial content, the pandorabot starts with no knowledge. The [Superbot](#) is a set of the Top 10K most active AIML patterns and inputs you can use to create your own unique bot personality.

Rys. 3.1.2. Okno tworzenia nowego pandorabota

Podczas tworzenia nowego pandorabota (rys. 3.1.2), botmaster nadaje mu imię oraz wybiera zestaw startowej wiedzy, w jaką ów bot zostanie zaopatrzony (**autor pracy wybrał ostatnią opcję, zaczynając budowę chatbota Loki z pustą bazą wiedzy**).



Rys. 3.1.3. Pasek nawigacyjny platformy

Botmaster porusza się po portalu za pomocą paska nawigacji (rys. 3.1.3). Pierwszy wiersz zawiera odnośniki do aktualnie edytowanych botów, kreacji nowego pandorabota oraz stron z informacjami, pomocą i oferowanymi usługami. Okno "Pandorawriter" pozwala na automatyczną zamianę dialogu w formie tekstowej na format AIML (stworzone kategorie będą miały najprostsza postać). W drugim wierszu znajdują się okna do edycji i zarządzania danego pandorabota. W kolejności od pierwszego do ostatniego:

1. Train - testowanie odpowiedzi pandorabota,
2. Properties - zarządzanie parametrami wewnętrznymi pandorabota,
3. Configure - ustawienie wartości etykiety <get> (domyślnie puste),
4. AIML - zarządzanie plikami AIML z bazą wiedzy bota,
5. Custom HTML - zarządzanie stroną domową pandorabota,
6. Oddcast VHost - oferta zaopatrzenia pandorabota w animowaną postać,
7. HostABot - oferta zaopatrzenia pandorabota w animowaną postać z głosem,
8. Logs - archiwa przeprowadzonych przez bota rozmów (czyszczone co tydzień),
9. Explore - badanie istniejących kategorii pandorabota,
10. Subscribers - dodatkowe usługi.

Loki >> Training

Type in questions below that you want your pandorabot to be able to answer. If you don't like the answer, type in the response that you would like your pandorabot to give instead.

(current topic: u_name)

Human:

Human: hi

Matched: HI (category defined in [00_0.aiml](#))

Loki: Hello. How shall I call you?

Loki:

Rys. 3.1.4. Okno testowania odpowiedzi pandorabota

Okno (rys. 3.1.4), w którym można trenować pandorabota, służy do testowania wprowadzonych wcześniej w nim zmian bądź tworzenia nowych kategorii. Po zadaniu pytania bot zwraca odpowiedź, wyświetlając przy tym informacje dotyczące tematu jaki został nastawiony oraz pierwszego wzorca, z jakim powiązał podane wyrażenie. Stworzone i przetworzone w tym oknie nowe kategorie pojawiają się w pliku AIML o nazwie "update.AIML". W przypadku gdy botmaster chce stworzyć bardziej zaawansowaną kategorię niż zwracającą pojedyncze wyrażenie, może on użyć opcji "Advanced Alter Response".

Loki >> Training >> Advanced Alter Response

	pattern	that	topic
Input	hi		
Matched	HI		
	<input type="text" value="HI"/>	<input type="text"/>	<input type="text"/>
	<input type="checkbox"/> new category depends on that		<input type="checkbox"/> new category depends on topic
New AIML Category	<p>template</p> <pre><srail>HELLO</srail></pre> <p><input type="button" value="Clear"/> <input type="button" value="<think>"/> <input type="button" value="<random>"/> <input type="button" value="<sr/>"/> <input type="button" value="<srail>"/> <input type="button" value=""/></p>		
	<input type="button" value="Submit"/> <input type="button" value="Cancel"/>		

Rys. 3.1.5. Zaawansowane okno tworzenia nowej kategorii

Okno tworzenia nowych kategorii (rys. 3.1.5) pozwala na ustawienie etykiet `<that/>` i `<topic/>` oraz modyfikację kodu znajdującego się wewnątrz szablonu (eng. template). Przyciski etykiet umożliwiają automatyczną generację danego elementu w kategorii.

Zadeklarowane są tam jednak tylko podstawowe funkcje. W celu wykorzystania pozostałych etykiet, botmaster musi wpisać je odręcznie do pola szablonu.

FRAN >> Properties

Name	Value	Delete	Name	Value	Delete
gender	female	<input type="checkbox"/>	race	replicator	<input type="checkbox"/>
name	FRAN	<input type="checkbox"/>			

Add property named: with value:

Copy all properties from:

Rys. 3.1.6. Okno parametrów pandorabota

W powyższym oknie (rys. 3.1.6) następuje edycja i zarządzanie parametrami wewnętrznymi pandorabota. Są to parametry wywoływane za pomocą komend `<bot name="XX"/>`. Tylko botmaster może je modyfikować. Parametry te można także skopiować z innego pandorabota, zamieszczonego na koncie botmastera.

FRAN >> AIML files

If you're new to Pandorabots and [AIML](#) you should first try customizing your pandorobot by changing some of its [properties](#) or by providing your own custom responses with the [training](#) interface.

For more advanced botmasters, this page allows you to download, modify and upload the AIML files for your pandorobot directly.

The tables below show all the AIML files for this pandorobot. To view or edit a file, click on its name.

Filename	Categories	Size	Last Modified	Browse	Download	Library	Active	Delete
00.aiml		8 1,223	09/29/2011 02:23:18 PM GMT	Browse	Download	(local)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
korekta.aiml		9 989	02/19/2012 07:24:54 PM GMT	Browse	Download	(local)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Total number of files: 2

To add a new AIML file to your Pandorobot, you can either [create a new AIML file](#) from scratch using your browser as the editor, or you can create a file on your local machine and use the form below to upload it to Pandorabots.

Enter the names of the AIML files you wish to upload in the form below. You can use the **Browse...** button to locate the files on your local machine. If you don't see a **Browse..** button your browser may not support file uploads. Each file must either be an AIML file or a free-format text dialog file to be converted to AIML using [Pandorawriter](#). To convert a file using Pandorawriter you should check the appropriate **convert text dialog to AIML** check box.

File 1: Nie wybrano pliku convert text dialog to AIML

File 2: Nie wybrano pliku convert text dialog to AIML

File 3: Nie wybrano pliku convert text dialog to AIML

File 4: Nie wybrano pliku convert text dialog to AIML

File 5: Nie wybrano pliku convert text dialog to AIML

Rys. 3.1.7. Okno edycji plików AIML pandorabota

Do bezpośredniej pracy na plikach bazy wiedzy wykorzystuje się okno "AIML files" (rys. 3.1.7). Można w nim wykonywać następujące czynności:

- * zamieszczać pliki z komputera na serwerze pandorabota,
- * modyfikować zawartość zamieszczonych plików pandorabota,
- * usuwać zamieszczone pliki AIML,
- * aktywować bądź deaktywować plik z wykorzystywanych przez pandorabota.

Zamieszczone pliki są wylistowane wraz z informacjami na temat:

- * liczby zawartych w nich kategorii,
- * wielkości pliku,
- * daty ostatniej modyfikacji.

FRAN >> Custom HTML

This page lets you create one or more custom HTML files to personalize the appearance of your pandorabot's published page. Before creating a custom HTML file be sure to read the [description](#) of the format of the HTML page.

The table belows show all the Custom HTML files for this pandorabot. To view or edit a file, click on its name.

Filename	Download	Access URL	Delete
fran.html	Download	http://www.pandorabots.com/pandora/talk?botid=d03814484e343d41&skin=fran	<input type="checkbox"/>

Default:

To add a new Custom HTML file to your Pandorabot, you can either [create a new Custom HTML file](#) from scratch using your browser as the editor, or you can create a file on your local machine and use the form below to upload it to Pandorabots.

Enter the name of the Custom HTML file you wish to upload in the form below. You can use the **Browse...** button to locate the file on your local machine. If you don't see a **Browse..** button your browser may not support file uploads.

File: Nie wybrano pliku

Rys. 3.1.8. Okno strony domowej pandorabota

W celu publikacji pandorabota w sieci, należy najpierw stworzyć jego stronę domową. Okno obsługujące pliki HTML (rys. 3.1.8) umożliwi zamieszczenie plików na serwerze. Można wykorzystać dostępny podstawowy skrypt lub samemu stworzyć unikatową stronę pandorabota (autor pracy zaprojektował stronę samodzielnie). Stronę rozpoznawaną jako startową wybiera się w domyślnym ustawieniu.

3.2 Struktura języka AIML

Język AIML, oparty na standardzie XML, ściśle przestrzega zakończeń reguł. Ze względu na jego budowę, możliwa jest jego edycja za pomocą narzędzi do edycji języka XML [11]. Przy tworzeniu skryptów AIML chatterbota wykorzystuje się etykiety zawarte w tabeli 3.2.1.

Tab 3.2.1. Etykiety wykorzystywane przez Pandoraboty

Etykieta AIML	Pozycja w dokumencie WD	Uwagi
<aiml>	3.2. AIML Element	separator bloku AIML
<topic name="X">	4. Topic	X jest wzorcem AIML
<category>	5. Category	jednostka pamięci AIML
<pattern>	6. Pattern	wzorzec wejścia AIML
<that>	6.1. Pattern-side That	zawiera wzorzec AIML
<template>	7. Template	szablon odpowiedzi AIML
<star index=„N”/>	7.1.1. Star	odwołanie do *
<that index=„M,N”/>	7.1.2. Template-side That	poprzednia odpowiedź bota
<input index=„N”/>	7.1.3. Input	wiadomość na wejściu
<thatstar index=„N”/>	7.1.4. Thatstar	odwołanie do * zawartego w <that>
<topicstar index=„N”/>	7.1.5. Topicstar	odwołanie do * zawartego w <topic>
<get name="XXX"/>	7.1.6. Get	XXX domyślnie zdefiniowane przez botmastera
<bot name="XXX"/>	7.1.6.1. Bot	unikalny parametr bota
<sr/>	7.1.7. Short-cut elements	<sr><star/></sr>
<person2/>	7.1.7. Short-cut elements	<person2><star/></person2>
<person/>	7.1.7. Short-cut elements	<person><star/></person>
<gender/>	7.1.7. Short-cut elements	<gender><star/></gender>
<id/>	7.1.8. System-defined predicates	identyfikator klienta
<uppercase>	7.2.1. Uppercase	konwertuje tekst na wielkie litery
<lowercase>	7.2.2. Lowercase	konwertuje tekst na małe litery
<formal>	7.2.3. Formal	każde słowo przekształca na rozpoczynające się wielką literą
<condition name="X" value="Y">	7.3.1. Condition	gałąź pojedynczej akcji
<condition name="X">	7.3.1. Condition	wiele rozgałęzień wyboru
<condition>	7.3.1. Condition	wiele rozgałęzień wielu parametrów

<li name="X" value="Y">	7.3.1. Condition	wykorzystywane przez <condition>
<li value="Y">	7.3.1. Condition	wykorzystywane przez <condition>
	7.3.2. Random	wykorzystywane przez <random>, <condition>
<random>	7.3.2. Random	pojedynczy, losowy wybór
<set name="XXX">	7.4.1. Set	może zwrócić XXX lub wartości
<srai>	7.5.1. SRAI	rekurencja
<person2>	7.6.1. Person2	zamiana 1 i 3 osoby
<person>	7.6.2. Person	zamiana 1 i 2 osoby
<gender>	7.6.3. Gender	zamiana płci zaimków
<think>	7.7.1. Think	ukrywa skutki uboczne
Rozszerzenia dla Pandorobotów	Działanie	Uwagi
<condition name="X" value="*">	niezdefiniowana wartość parametru	pojedynczy warunek
<li name="X" value="*">	niezdefiniowana wartość parametru	wykorzystywane przez <condition>
<li value="*">	niezdefiniowana wartość parametru	wykorzystywane przez <condition>
<date locale="X" timezone="Y" format="Z"/>	data i godzina	format Unix strftime
<that index="M, *"/>	poprzednia wypowiedź bota	multi-zdanie
<request index=„N"/>	zapytania na wejściu	multi-zdanie
<response index=„N"/>	odpowiedzi na wyjściu	multi-zdanie
<learn>	zachowaj kategorię AIML	niestandardowe
<eval>	ewaluacja AIML	wyrażenie wewnątrz <learn>

Opis etykiet języka AIML.

Etykieta <aiml version="XXX">...</aiml>

- Może zawierać w sobie etykiety <topic> i <category>.

Etykieta <aiml> otwiera i zamyka pliki AIML. Zawiera ona w sobie informację o wykorzystywanej w dokumencie wersji języka AIML (w przypadku serwera Pandorobot, jest to <aiml version="1.0">). Wewnątrz niej zawarta jest wiedza chatterbota.

Przykład:

```
<aiml version="1.0">
  <category>
    <pattern>*</pattern>
    <that>WHO IS THERE</that>
    <template><star/> Who?</template>
  </category>
</aiml>
```

Etykieta `<topic name="X">...</topic>`

- Zawarta jest w etykietcie `<aiml>`.
- Musi zawierać w sobie etykiety `<category>`.

Podczas wyszukiwania odpowiedzi, "tematy" pozwalają chatterbotowi dokonać wyboru w oparciu o temat dyskusji. Umożliwiają one powstanie wielu kategorii wywoływanych taką samą informacją wejściową, zawierających jednak różny kontekst.

Tematy można wykorzystywać na początku plików AIML i odwoływać się do nich za pośrednictwem `<get name="topic"/>` i `<set name="topic"></set>`. Wewnątrz nich zawarta jest jedna bądź częściej więcej kategorii. Kategorie, otoczone etykietami `<topic></topic>`, będą skojarzone wyłącznie ze zdefiniowanym tematem, w którym się znajdują. Struktura pliku *.AIML z wykorzystaniem tematu została przedstawiona poniżej:

```
<aiml>
<topic name="THE TOPIC">
  <category>
    <pattern> PHRASE </pattern>
    <that> PHRASE </that>
    <template> Phrase </template>
  </category>
</topic>
</aiml>
```

Gdy `<topic>` zostanie ustawiony, kategorie przypisane do danego tematu zostaną przeszukane jako pierwsze. W razie nie znalezienia odpowiedzi, program przeszukuje kategorie nie powiązane z żadnym tematem.

Nazwy tematów mogą zawierać w sobie jeden wieloznacznik („*” bądź „_”). Pozwala to na stworzenie podtematów. Wieloznaczniki „_” działają tak samo jak „*”, mają jednak większą wagę i są sprawdzane na początku.

Przykład:

Temat : <topic name="* DOGS"></topic>

Podtemat 1: <topic name="CARING FOR DOGS"></topic>

Podtemat 2: <topic name="GROOMING DOGS"></topic>

W przypadku ustawionego tematu “*CARING FOR DOGS*”, chatterbot najpierw szukałby odpowiedzi w podtemacie 1, po czym w temacie “** DOGS*”, a na końcu w podstawowych kategoriach, nie powiązanych z żadnym tematem.

Etykieta <category>...</category>

- Zawarta jest w etykiecie <aiml>.
- Może znajdować się w etykiecie <topic>.
- Zawsze musi zawierać w sobie etykiety <pattern> i <template>.

Kategoriami nazywa się wyrażenia, z których składa się plik AIML. Każda kategoria zawiera klucz wejściowy (określony ciąg znaków) i odpowiedź wyjściową (może się wyświetlić bądź zawierać jakąś procedurę).

```
<category>
  <pattern> PATTERN </pattern>
  <template> Template </template>
</category>
```

Bądź:

```
<category>
  <pattern> PATTERN </pattern>
  <that> THAT </that>
  <template> Template </template>
</category>
```

Etykieta <pattern>...</pattern>

- Zawarta jest w etykiecie <category>.
- Może zawierać w sobie wieloznaczniki i tekst znormalizowany.
- Nie może być pusta.
- Zawsze musi w sobie zawierać tekst znormalizowany i PSAE*.

Wzorcem (z ang. *pattern*) nazywana jest część kategorii zawierająca informację wejściową, wywołującą daną kategorię. Etykieta ta dozwolona jest wyłącznie w kategorii.

* **PSAE** (Pattern-side AIML expressions). Wyrażenia AIML po stronie wzorca, pojawiające się wewnątrz etykiet <pattern>, <that> i <topic>.

Może być tylko jedna i musi znajdować się na pierwszej pozycji, tuż po znaczniku `<category>`.

Wzorzec zawiera w sobie wyrażenie zapisane wielkimi literami. Często występują tam wieloznaczniki „*” i „_”, które mogą być dowolnym ciągiem znaków wprowadzonych przez rozmówcę (nie mogą jednak pozostać puste).

Wieloznaczniki są brane pod uwagę podczas wyszukiwania odpowiedzi.

Przykład:

```
<pattern>HELLO</pattern>
```

Wzorzec zostanie wywołany tylko w przypadku pojawienia się samego słowa klucza „Hello”.

```
<pattern>HELLO *</pattern>
```

Wzorzec zostanie wywołany tylko w przypadku, gdy na początku pojawi się słowo klucz, a za nim znajdować się będzie jakieś wyrażenie bądź ciąg znaków np. „Hello how are you?”.

Etykieta `<that>...</that>`

- Zawarta jest w etykiecie `<category>`.
- Może zawierać wieloznaczniki i tekst znormalizowany (PSAE).
- Może być pusta.

Wyrażenie „that” odnosi się do wcześniejszej wypowiedzi bota. W języku AIML etykieta `<that>...</that>` pozwala na dodatkowy wzorzec wyszukiwania zależny od wcześniej informacji wyjściowej.

Przykład: Odpowiedzi na kategorii typu „yes-no”:

```
<category>
  <pattern>YES</pattern>
  <that> DO YOU LIKE MOVIES </that>
  <template> What's your favorite movie? </template>
</category>
```

Kategoria dla wyrażenia „YES”, zwracana gdy rozmówca odpowiada na pytanie “Do you like movies”.

Etykietę tę można także wykorzystać wewnątrz szablonu `<template>` w postaci `<that index="M,N"/>`. Jeśli część `index="M,N"` zostanie pominięta, to bot przyjmuje

wartości „1,1” (<that/> jest równoznaczne z <that index="1,1"/>).

Przykład: Często wykorzystywana jest kategoria odpowiadająca na pytanie “WHY”

```
<category>
  <pattern>WHY</pattern>
  <template>"<that/>"? Why?</template>
</category>
```

Może ona spowodować powstanie dialogu w postaci:

Robot: *Do not ask me any more questions please.*

Client: *WHY*

Robot: *"Do not ask me any more questions please"? Why?*

Można się przed nią zabezpieczyć tworząc dodatkową kategorię:

```
<category>
  <pattern>WHY</pattern>
  <that> DO NOT ASK ME ANY MORE QUESTIONS PLEASE</that>
  <template>Because I would rather talk about you.</template>
</category>
```

Etykieta <template>...</template>

- Musi znajdować się w etykiecie <category>.
- Może zawierać TSAE*.
- Może pozostać pusta.

Szablonem (ang. *template*) nazywa się część kategorii będącą wartością wyjściową. Wewnątrz szablonu jest skonstruowana cała odpowiedź i działanie bota. Najprostsza odpowiedź może składać się ze zwykłego tekstu. AIML zawiera liczne funkcje znaczników, umożliwiające dostosowanie się do poszczególnych wejść.

Szablon może odwołać się do innych kategorii za pomocą etykiet <sr/> i <srail>. Wiele szablonów jest prostą redukcją mapującą jedno wyrażenie do drugiego.

Przykład:

```
<category>
  <pattern>DO YOU KNOW WHAT * IS</pattern>
  <template><srail>WHAT IS <star/> </srail></template>
</category>
```

"Do you know what X is?" ulega zmianie na "What is X".

* TSAE (Template-side AIML Expression). Wyrażenia AIML po stronie szablonu, składają się ze zwykłego tekstu i zawierających je poszczególnych etykiet.

Szablon może również zawierać fragmenty odwołujące się do HTML, JavaScript i XML. Te wbudowane znaczniki umożliwiają przeglądarce odtwarzanie dźwięku, wyświetlanie obrazów i uruchamianie programów.

Etykieta `<star index=„N”/>`

- Może znajdować się w etykiecie `<template>`.

Gwiazda (ang. *star*) odwołuje się do wieloznaczników („*” i „_”) zawartych we wzorcu (`<pattern>`). Numer indeksu określa, które wyrażenie jest wykorzystywane.

`<star index="1"/>` zawiera fragment tekstu zawarty w pierwszym wieloznaczniku,
`<star index="2"/>` zawiera fragment tekstu zawarty w drugim wieloznaczniku,
`<star index="3"/>` zawiera fragment tekstu zawarty w trzecim wieloznaczniku,
itd.

Pominięty indeks przyjmuje automatycznie wartość "1" (`<star/>` jest równoznaczne z `<star index="1"/>`).

Przykład:

```
CLIENT: "Hello there, my name is Tom"  
<pattern>HELLO * MY NAME IS *</pattern>  
  
<star index="1"/> = there  
<star index="2"/> = tom
```

Etykieta `<input index=„N”/>`

Zwraca całe polecenie użytkownika. Wartość "N" odnosi się do odpowiednio wcześniejszej wypowiedzi rozmówcy. Wartość "1" zwróci ostatnią sentencję, wartość "2" zwróci poprzedzającą ostatnią wypowiedź, wartość "3" jeszcze wcześniejszą itd..

Pominięty indeks przyjmuje automatycznie wartość "1" (`<input/>` jest równoznaczne z `<input index="1"/>`).

Przykład:

Client: *MY NAME IS BILL.*

Robot: *Bill, are you married?*

Client: *YES.*

Robot: *What kind of a car do you drive?*

Client: *I DRIVE A MAZDA.*

Poszczególne wartości `<input>` przeprowadzonej rozmowy prezentują się następująco:

`<input index="1"/>` = *I DRIVE A MAZDA*

`<input index="2"/>` = *YES*

`<input index="3"/>` = *MY NAME IS BILL*

Etykieta `<thatstar index=„N”/>`

- Może znajdować się w etykiecie `<template>`.

Etykieta `<thatstar index=„N”/>` zwraca fragment tekstu zawarty w wieloznaczniku „*” na pozycji "N" wewnątrz etykiety `<that>...</that>`. Pominięty indeks przyjmuje automatycznie wartość "1" (`<thatstar/ >` jest równoznaczne z `<thatstar index="1"/ >`).

Przykład:

```
<category>
  <pattern>*/</pattern> <that>WHAT IS YOUR FAVORITE */</that>
  <template>
    <srai>MY FAVORITE <thatstar/> IS <star/></srai>
  </template>
</category>
```

Kategoria ta wywoływana jest przy wszystkich pytaniach zadanych przez chatterbota odnośnie ulubionych rzeczy np. "*What is your favorite movie?*", "*What is your favorite color?*", itd. redukując w ten sposób wszystkie odpowiedzi do podanej sentencji.

Etykieta `<topicstar index=„N”/>`

- Może znajdować się w etykiecie `<template>`.

Etykieta `<topicstar index="N"/>` zwraca fragment tekstu zawarty w wieloznaczniku „*” na pozycji "N" wewnątrz etykiety `<topic>...</topic>`. Pominięty indeks przyjmuje automatycznie wartość "1" (`<topicstar/>` jest równoznaczne z `<topicstar index="1"/>`).

Etykieta `<get name="XXX"/>`

- Zawarta jest w etykiecie `<template>`.
- Może pozostać pusta.

W języku AIML występują dwa rodzaje parametrów: tylko do odczytu i wprowadzone przez klienta. `<get name="XXX"/>` jest odwołaniem do wstępnie ustawionego przez rozmówcę, za pomocą etykiety `<set name="XXX">...</set>`, parametru. Parametry klienta nie mogą być użyte w etykiecie `<pattern>`.

Element `<get>` wstawia przypisaną mu zawartość do szablonu. Jeśli nie ma on określonej wartości, zastępuje go pusty łańcuch. Nazwa elementu (`name="XXX"`) wskazuje na parametr, który ma zostać pobrany.

Etykieta `<bot name="XXX"/>`

- Zawarta jest w etykiecie `<template>`.

`<bot name="XXX"/>` jest odwołaniem do wstępnie zadeklarowanego parametru. Działa tak samo jak element `<get>`, wstawiając na swoje miejsce wartość przypisaną parametrowi, do którego się odwołuje. Parametry elementu `<bot>` są z góry założone w programie przez botmastera i nie ulegają zmianie. Klient nie ma na nie wpływu.

Etykieta `<person> ...</person>`, `<person2>`

- Może znajdować się w etykiecie `<template>`.

Etykiety `<person>` i `<person2>` zaznaczają miejsce, w którym interpreter AIML zamienia zaimki osobowe w wyrażeniu.

`<person> X </person>` zamienia pierwszą osobę z trzecią osobą.

Przykład:

Client: *I admire robots like you.*

Robot: *That's good information: Joe said he admire robots like me.*

Transformacja tekstu następuje w dwóch miejscach:

1. zamienia zaimek z pierwszej osoby na drugą osobę (<person2>),
2. zamienia zaimek z trzeciej osoby na pierwszą osobę (<person>).

Etykietę tę można także skrócić do postaci:

`<person/> = <person><star/></person>`

Etykieta <person2> ...</person2>, <person2/>

- Może znajdować się w etykiecie <template>.
- Może zawierać TSAE.
- Może być pusta.

Zamienia zaimki z pierwszej na drugą osobę. Etykietę tę można także skrócić do postaci:

`<person2/> = <person2><star/></person2>`

Etykieta <gender>...</gender>

- Zawarta jest w etykiecie <template>.

Działa podobnie do etykiety <person>, zmienia jednak płeć, nie osobę.

Przykład:

`<gender>She told him to take a hike..</gender>`

Zwróci na wyjściu: "*He told her to take a hike*".

Etykieta <id/>

- Nie posiada żadnej przypisanej wartości.

Program domyślnie zwraca adres lokalnego hosta. Etykieta <id/> umożliwia identyfikację użytkownika.

Etykieta `<uppercase>text</uppercase>`

- Zawarta jest w etykiecie `<template>`.

Pozwala formatować tekst zawarty wewnątrz etykiety. Zamienia wszystkie litery na wielkie.

Przykład:

```
<category>
<pattern>DO UPPERCASE</pattern>
  <template>
    This is an <uppercase>uppercase</uppercase> test.
  </template>
</category>
```

Podana kategoria zwraca na wyjściu następującą wiadomość:
„*This is an UPPERCASE test*”.

Etykieta `<lowercase>text</lowercase>`

- Zawarta jest w etykiecie `<template>`.

Pozwala formatować tekst zawarty wewnątrz etykiety. Zamienia wszystkie litery na małe.

Przykład:

```
<category>
<pattern>DO LOWERCASE</pattern>
  <template>
    This is a <lowercase>LOWERCASE</lowercase> test.
  </template>
</category>
```

Podana kategoria zwraca na wyjściu następującą wiadomość:
”*This is a lowercase test*”.

Etykieta `<formal> text </formal>`

- Musi znajdować się w etykiecie `<template>`.

Etykieta pozwala na formatowanie zawartego w niej tekstu. Każde słowo zaczyna się od wielkiej litery, pozostałe litery są małe.

Przykład:

```
<category>
  <pattern>DO FORMAL</pattern>
  <template>
    My name is <formal>jon baer</formal>.
  </template>
</category>
```

Wywołany tekst będzie wyglądał następująco: "*My name is Jon Baer.*"

Etykieta `<condition>...</condition>`

- Zawiera się w etykiecie `<template>`.
- Może zawierać TSAE.
- Może być pusta.

Etykieta `<condition>` zwraca przypisane jej elementy, gdy zostanie spełniony określony w niej warunek. Warunek (ang. *condition*) może przybrać jedną z trzech form:

- a) multi wyboru,
- b) listy,
- c) pojedynczego warunku.

W szablonie może występować wiele warunków. Badane wartości mogą zawierać standardowy tekst i jeden symbol wieloznacznika „*”.

Ad. a.

Cała zawartość tej etykiety zostanie wykonana.

```
<condition name="X" value="Y">... </condition>
```

```
<condition name="X" value="*"> wywołana dla wartości nieokreślonych parametru  
"X"</condition>
```

Przykład:

```
<category>
  <pattern>I AM BLOND</pattern>
  <template>You sound very
    <condition name="gender" value="female"> attractive.</condition>
    <condition name="gender" value="male"> handsome.</condition>
  </template>
</category>
```

Ad. b.

`<condition/>` musi zawierać przynajmniej jedną etykietę typu ``. Przerzywa działanie gdy znajdzie pierwszą etykietę spełniającą przypisany jej warunek. Wszystkie pozostałe etykiety `` nie spełniające danego warunku są ignorowane.

```
<condition>
```

```
<li name="X" value="Y"></li>
```

```
<li name="X" value="*"> wywołana dla wartości nieokreślonych parametru "X"</li>
```

```
</condition>
```

Przykład:

```
<category>
  <pattern>I AM BLOND</pattern>
  <template>You sound very
    <condition>
      <li name="gender" value="female"> attractive.</li>
      <li name="gender" value="male"> handsome.</li>
    </condition>
  </template>
</category>
```

Ad. c.

`<condition/>` musi zawierać przynajmniej jedną etykietę typu ``. Przerzywa działanie gdy znajdzie pierwszą etykietę spełniającą przypisany jej warunek. Wszystkie pozostałe etykiety `` nie spełniające danego warunku są ignorowane.

```
<condition name="X"> </condition>
```

```
<li value="Y"></li>
```

```
<li value="*"> wywołana dla wartości nieokreślonych parametru "X"</li>
```

```
</condition>
```

Przykład:

```
<category>
  <pattern>I AM BLOND</pattern>
  <template>You sound very
    <condition name="gender">
      <li value="female"> attractive.</li>
      <li value="male"> handsome.</li>
    </condition>
  </template>
</category>
```

Odpowiedź na wyjściu wszystkich trzech przykładów zależna będzie od

parametru „gender” rozmówcy i zostanie wyświetlone odpowiednio:

"You sound very attractive."

bądź

"You sound very handsome."

Inną możliwością wykorzystania etykiety warunku jest wprowadzenie pozycji podstawowej. Wywołana ona zostanie w przypadku, gdy żadna wcześniejsza nie zostanie zwrócona.

Przykład:

```
<category>
  <pattern>I AM BLOND</pattern>
  <template>You sound very
    <condition>
      <li name="gender" value="female">attractive.</li>
      <li>handsome.</li>
    </condition>
  </template>
</category>
```

Etykieta `...`

- Zawarta w etykietce `<condition>` bądź `<random>`.
- Może pozostać pusta.

Jest to element listy, który może zawierać tekst standardowy bądź dowolną etykietę AIML dozwoloną w `<template>`.

Etykieta `<random>...</random>`

- Może znajdować się w etykietce `<template>`.
- Musi zawierać etykiety ``.

Etykieta ta wykorzystywana jest do losowego wyboru jednego z szeregu elementów.

Przykład:

```
<random>
  <li>A</li>
  <li>B</li>
  <li>C</li>
</random>
```

Zwraca na wyjściu „A”, „B” bądź „C” losowo.

Etykieta `<random>` ma wyższy priorytet niż pozostałe etykiety. Parser AIML interpretuje wyłącznie elementy zawarte w wybranym znaczniku ``. Pozostałe pozycje są ignorowane.

Etykieta `<set name="XXX">...</set>`

- Może zawierać TSAE.

Etykieta `<set>` instruuje interpreter AIML do nastawienia wartości parametru klienta. Zawarta w niej nazwa (`name="XXX"`) wskazuje na odpowiedni element, który jest modyfikowany bądź tworzony.

Interpreter AIML zwraca na wyjściu przetwarzaną wewnątrz etykiety `<set name="XXX">...</set>` wartość. Element `<set>` nie może przeprowadzać żadnego formatowania tekstu na zwracanym przez siebie wyrażeniu.

Etykieta `<sr>...</sr>`, `<sr/>`

- Może znajdować się w etykiecie `<template>`.
- Może zawierać TSAE.
- Element `<sr/>` musi pozostać pusty.

Funkcja rekurencyjna `<sr>` (ang. *Stimulus-Response Artificial Intelligence*, stymulowana odpowiedź sztucznej inteligencji) wywołuje odwołanie zawartego w niej tekstu do kolejnej kategorii. W jej wyniku wprowadzony przez rozmówcę tekst na wejściu jest zamieniany na tekst zawarty w tej etykiecie.

`<sr> X </sr>` wywołuje kategorię zawierającą wzorzec "X".

`<sr/>` jest równoważne z `<sr><star/></sr>`.

Etykieta `<sr>` znajduje największe zastosowanie przy symbolicznej redukcji złożonych zwrotów do ich mniej skomplikowanej formy.

Przykład:

```
<category>
  <pattern>DO YOU KNOW WHAT * IS</pattern>
  <template><sr>WHAT IS <star/></sr></template>
</category>
```

Dzięki temu botmaster może przechowywać większość wiedzy w prostszych kategoriach.

Czasami odpowiedź składa się z dwóch bądź więcej symbolicznych redukcji:

Przykład:

```
<category>
  <pattern>YES * </pattern>
  <template><srai>YES</srai> <sr/></template>
</category>
```

W podanej kategorii chatterbot odpowie na wszystkie wejścia typu „*Yes something*” łącząc odpowiedzi na „*Yes*” z odpowiedziami do „*something*”.

Często botmaster chce by ta sama odpowiedź była wywoływana przez kilka różnych wzorców. Ponieważ w każdej kategorii może zawarta być wyłącznie jedna etykieta `<pattern>`, `<srai>` używana jest do przekierowań każdej z nich do jednej kategorii.

Przykład:

```
<category>
  <pattern>GOODBYE</pattern>
  <template><srai>BYE</srai></template>
</category>

<category>
  <pattern>BYE</pattern>
  <template>See you later.</template>
</category>
```

Etykieta `<think>...</think>`

- Zawiera się w etykiecie `<template>`.
- Może znajdować się w etykietach `` i formatujących tekst.

Celem myślenia (ang. *think*) jest wykonanie czynności zawartej w etykiecie, bez wyświetlania na wyjściu jej zawartości (ukrywając działania przed rozmówcą).

Przykład:

```
<category>
  <pattern>I AM FEMALE</pattern>
  <template>
    Thanks for telling me your gender.
    <think><set name="gender">female</set></think>
  </template>
</category>
```

Etykieta `<set name="gender">female</set>` normalnie zwraca wyraz "female". Etykieta `<think>` ukrywa jednak tekst zawarty w jej wnętrzu. Kategoria zwraca na

wyjściu odpowiedź:

“Thanks for telling me your gender.”

Etykieta `<date locale="X" timezone="Y" format="Z"/>`

- Zawiera się w etykiecie `<template>`.

Pandoraboty obsługują trzy parametry w etykiecie `<date>`:

locale – pozwala na wybór języka wykorzystujący standardowe, regionalne, nazewnictwo. Domyślnie zdefiniowany jest „*en_US*”.

format – pozwala na wybór formatu i funkcji wyświetlanego czasu. Definiuje wartości wyświetlane i sposób ich przedstawienia (AM/PM, czas 24 godzinny, uwzględnienie dnia tygodnia, itd.). Domyślnie brane są ustawienia dla zdefiniowanej wartości *locale*.

timezone – nastawia strefę czasową w liczbie godzin od GMT (*Greenwich Mean Time*, czas uniwersalny)

Jeśli któryś z parametrów jest pominięty bądź nieważny, brana jest jego wartość domyślna.

Przykład:

Nastawione wyświetlenie daty i czasu po francusku, używając strefy czasu Europy Środkowej:

```
<date locale="fr_FR" timezone="-1" format="%c"/>
```

Etykiety `<request index=„N”/>` i `<response index=„N”/>`

- Zawarte wewnątrz etykiety `<template>`.

Pytanie rozmówcy i odpowiedź chatterbota mogą często składać się z kilku oddzielnych sentencji. Język AIML posiada etykiety `<that/>` i `<input/>`, które się do nich odwołują. Wewnątrz nich zawarte są jednak tylko pojedyncze odpowiedzi. Pandoraboty zostały wyposażone w dodatkowe etykiety `<request/>` i `<response/>`, w których zawarta jest cała multi-wypowiedź (wszystkie sentencje podane przez rozmówcę na wejściu i wszystkie sentencje zwrócone przez bota).

Etykiety `<learn>...</learn>` i `<eval>...</eval>`

- Zawiera się w etykiecie `<template>`.

Etykieta `<learn>` dla Pandorabotów nie jest częścią standardowego języka AIML i działa na nieco innych zasadach niż w innych interpreterach. Wewnątrz tej etykiety znajduje się cała kategoria zawierająca etykiety `<pattern>`, `<template>` czy niekiedy `<that>`.

Etykieta `<eval>` jest wykorzystywana wewnątrz `<learn>` umożliwiając przetwarzanie tekstu.

Przykład [16]:

```
<learn>

<category>
  <pattern>
    <eval><uppercase><get name="badanswer-input"/></uppercase></eval>
  </pattern>
  <that>
    <eval><uppercase><get name="badanswer-that"/></uppercase></eval>
  </that>
  <template>
    <eval><get name="badanswer-newresp"/></eval>
  </template>
</category>

</learn>
```

`<eval>` zostało użyte do stworzenia etykiet `<pattern>`, `<that>` i `<template>` z przechowywanych, przetworzonych na wielkie litery, parametrów.

3.3 Rozpoznawanie tonu rozmówcy

Wypowiedzi człowieka przybierają różną formę w zależności od jego nastroju oraz rozmówcy. Może mieć ona charakter uprzejmy, oficjalny, może się również dystansować bądź pokazywać bliską więź względem drugiej osoby. Czasem staramy się kogoś świadomie skrzywdzić, urazić bądź sprawić mu przyjemność, pocieszyć. Ta różnorodność sprawia, iż jesteśmy w stanie przekazać jedną myśl na wiele sposobów wywierając liczne, odmienne reakcje u rozmówcy.

Program najczęściej skupia się tylko na rozpoznaniu samego zdania, a nie dodatkowych informacji jakie są w nim zawarte. Odczytanie takiego przekazu wymaga zdolności obcych maszynie, stąd też w tworzonym przez autora niniejszej pracy chatterbocie wprowadzony został parametr *bond* reagujący na różnego typu zwroty. Jego zadaniem miała być imitacja nastawienia programu względem rozmówcy.

Podczas analizy i redukcji informacji wejściowej, niektóre kategorie wzbogacone zostały przez autora o specjalne odwołania. Podzielić je można na cztery rodzaje:

- a) lokalizujące wulgarne słownictwo,
- b) lokalizujące skruchę rozmówcy,
- c) lokalizujące oficjalną formę, zdystansowaną,
- d) lokalizujące mowę potoczną.

Każde z nich prowadzi do innego rodzaju kategorii, których zadaniem jest zliczenie pewnej liczby tego typu zachowań i zmianę parametru *bond*, jeśli spełnione zostaną odpowiednie warunki. Ma to zapobiec ciągłemu przeskakiwaniu „nastrojów” chatterbota.

Na początku tworzenia chatterbota przekierowania parametru *bond* następowały bezpośrednio po wykryciu słów kluczowych. W wyniku tak przyjętego postępowania program narażony był na częstą zmianę wartości tego parametru, bez żadnego spójnego przejścia. Brakowało mu przez to wiedzy na temat jego wcześniejszego nastawienia względem rozmówcy. **Z tego powodu zostały wprowadzone dodatkowe parametry zliczające, takie jak *vg*, *of*, *cl* oraz *forgiven****.

* *vg* – licznik inwektyw
of – licznik formalizów
cl – licznik kolokwializmów
forgiven – znacznik wybaczenia

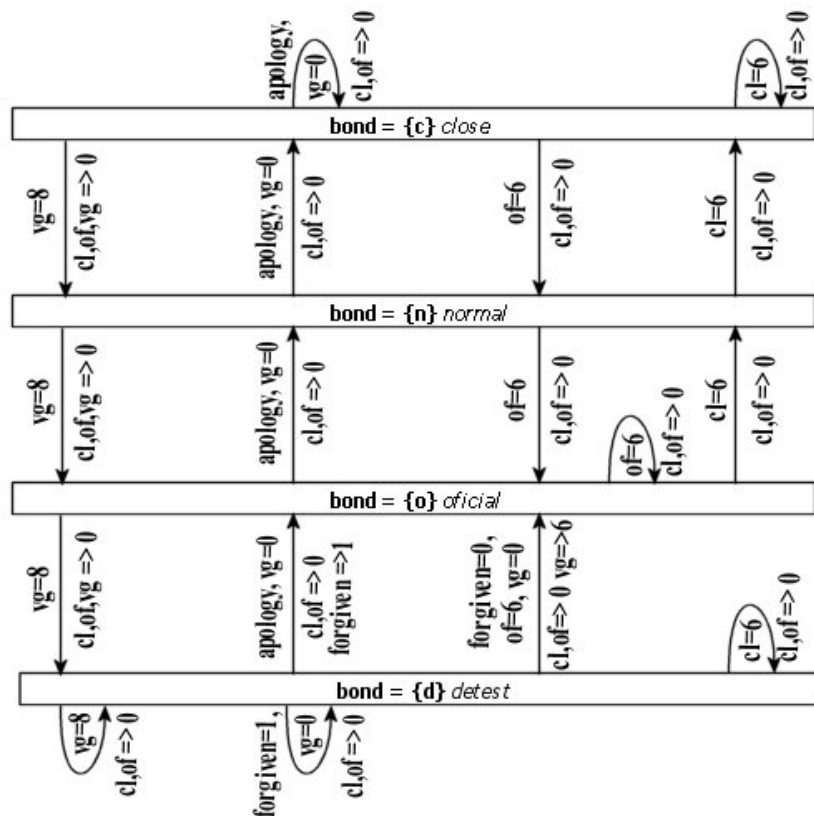
3.3.1. Wariacja parametru bond

Więź jaką chatterbot ma symulować opiera się na kilku prostych zależnościach. Ich liczba została ograniczona przez autora pracy, by nie powodować większych przeciążeń obliczeniowych.

Parametr *bond* ma przypisane cztery ustawienia $\{d,o,n,c\}$, które oznaczają jego nastawienie względem rozmówcy. Za pomocą $\langle condition name="bond" \rangle$ w kategoriach, odpowiedź chatterbota może być zróżnicowana dla każdego przypadku. Poziomy więzi z rozmówcą wyglądają następująco:

- d - niezadowolenie, pogarda (ang. *detest*),
- o - oficjalność, dystans (ang. *official*),
- n - ustawienie początkowe, neutralne (ang. *normal*),
- c - bliskość, koleżeńskość (ang. *close*).

W celu ich zmiany, parametr zliczający musi osiągnąć określoną w specjalnej kategorii wartość. Zamieszczony poniżej schemat (rys 3.3.1) przedstawia przyjęte wartości i reguły modyfikacji *bond*.



Rys. 3.3.1. Operacje zależności i zmiany wartości parametrów zliczających *vg*, *cl*, *of* oraz parametrów *forgiven* i *bond* przeprowadzane w kategoriach specjalnych

Poszczególne przejście na inny poziom więzi wywołuje wyzerowanie bądź przesunięcie (vg->4) części zliczających parametrów. Zapobiegać to ma zbyt częstej zmianie i płynniejszemu wydłużeniu zliczania.

Jak widać na przedstawionym schemacie poziom „d” parametru *bond* zachowuje się odmiennie od pozostałych. Jest on zależny od parametru *forgiven* oraz pozostaje bez zmian w przypadku, gdy *cl* dojdzie do wartości „6”. Chatterbot w ten sposób nie odzyska zaufania do rozmówcy, który zbyt wiele razy wyrażał się nagannie podczas ich rozmowy.

3.3.2. Lokalizacja i zliczanie inwektyw* przez chatterbota

W osobnym pliku (*06.AIML*) zgromadzona została część powszechnie stosowanych wulgaryzmów. Każdemu słowu bądź zwrotowi autor przyporządkował liczbę mówiąca o jego sile (założonym poziomie agresji jaki za sobą niesie). Zadaniem zamieszczonych tam kategorii było wyłapywanie ich z rozmowy, oraz reakcja na nie bądź samo odnotowanie ich użycia.

```
<category>
<pattern>SCREW YOU</pattern>
<template>
<srai>XXVULG1XX</srai>
</template>
</category>
```

List. 3.3.1. Kategoria w pliku *00_6.AIML*:

reakcja na wykryty wulgaryzm, przekierowanie do kategorii specjalnej

Każda z zawartych w pliku *00_6.AIML* kategorii zawiera w sposób pośredni bądź bezpośredni przekierowanie (list. 3.3.1), w którym zostaje zaktualizowany parametr *vg* zliczający użyte do tej pory wulgaryzmy. Mogą tam mieć miejsce także takie operacje jak zmiana słowa bądź przekierowanie do innej kategorii w celu znalezienia odpowiedzi.

Za pomocą komendy *<condition>* przeprowadzone jest w następujący sposób proste dodawanie wartości:

```
<category>
<pattern>XXVULG2XX</pattern>
<template>
```

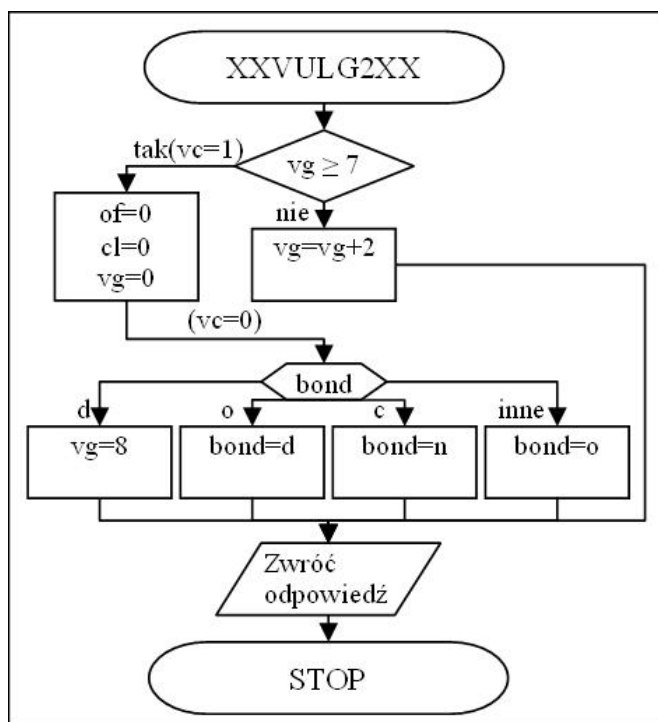
* Inwektywa - obelga, zniewaga słowna, obrażający zarzut

```

<condition name="vg">
  <li value="8"><think><set name="of">0</set><set name="vc">1</set></think></li>
  <li value="7"><think><set name="vg">0</set><set name="vc">1</set></think></li>
  <li value="6"><think><set name="vg">8</set></think></li>
  <li value="5"><think><set name="vg">7</set></think></li>
  <li value="4"><think><set name="vg">6</set></think></li>
  <li value="3"><think><set name="vg">5</set></think></li>
  <li value="2"><think><set name="vg">4</set></think></li>
  <li value="1"><think><set name="vg">3</set></think></li>
  <li><think><set name="vg">2</set></think></li></condition>
<condition name="vc" value="1">
  <think><set name="of">0</set><set name="cl">0</set><set name="vc">0</set></think>
  <condition name="bond">
    <li value="d"><think><set name="vg">8</set></think></li>
    <li value="o"><think><set name="bond">d</set></think></li>
    <li value="c"><think><set name="bond">n</set></think></li>
    <li><think><set name="bond">o</set></think></li></condition></condition>
<random>
  <li> Watch your language, pleas.</li>
  <li> Would you stop using such words?</li></random>
</template></category>

```

List. 3.3.2. Kategoria specjalna XXVULG2XX w pliku 00_9.AIML:
nastawienie parametru (vg+2) i zmiana wartości parametru bond



Rys. 3.3.2. Algorytm operacji zachodzących wewnątrz kategorii XXVULG2XX

Zadaniem podanej kategorii (list. 3.3.2) jest przede wszystkim zwiększenie parametru *vg* o wartość „2”. W momencie, gdy przekroczona zostanie wyznaczona granica, zostaje nastawiony parametr *vc* (utworzony w celu wywołania innego warunku). Parametr ten zostaje następnie wyzerowany w funkcji *<condition>* która jednocześnie przedstawia wartość przypisaną wcześniej dla *bond* (rys. 3.3.2).

Dodatkowo chatterbot udziela rozmówcy losowej odpowiedzi, mówiącej o jego niezadowoleniu. Odpowiedzi te różnią się w zależności od tego, jaka kategoria specjalna zostanie wywołana (zależnie od przypisanej siły danego zwrotu wywoływana jest jedna z czterech kategorii: *XXVULG1XX*, *XXVULG2XX*, *XXVULG3XX*, *XXVULG4XX*).

3.3.3. Lokalizacja i zliczanie deprekacji* przez chatterbota

Deprekacji zostały przypisane te same parametry co podczas zliczania inwektyw, wykonuje ona jednak przeciwną operację. W przypadku wyłapania przez chatterbota słów kluczowych uznanych za przeprosiny i skrucę wywoływana jest specjalna kategoria zmniejszająca wartości *vg* (list. 3.3.3).

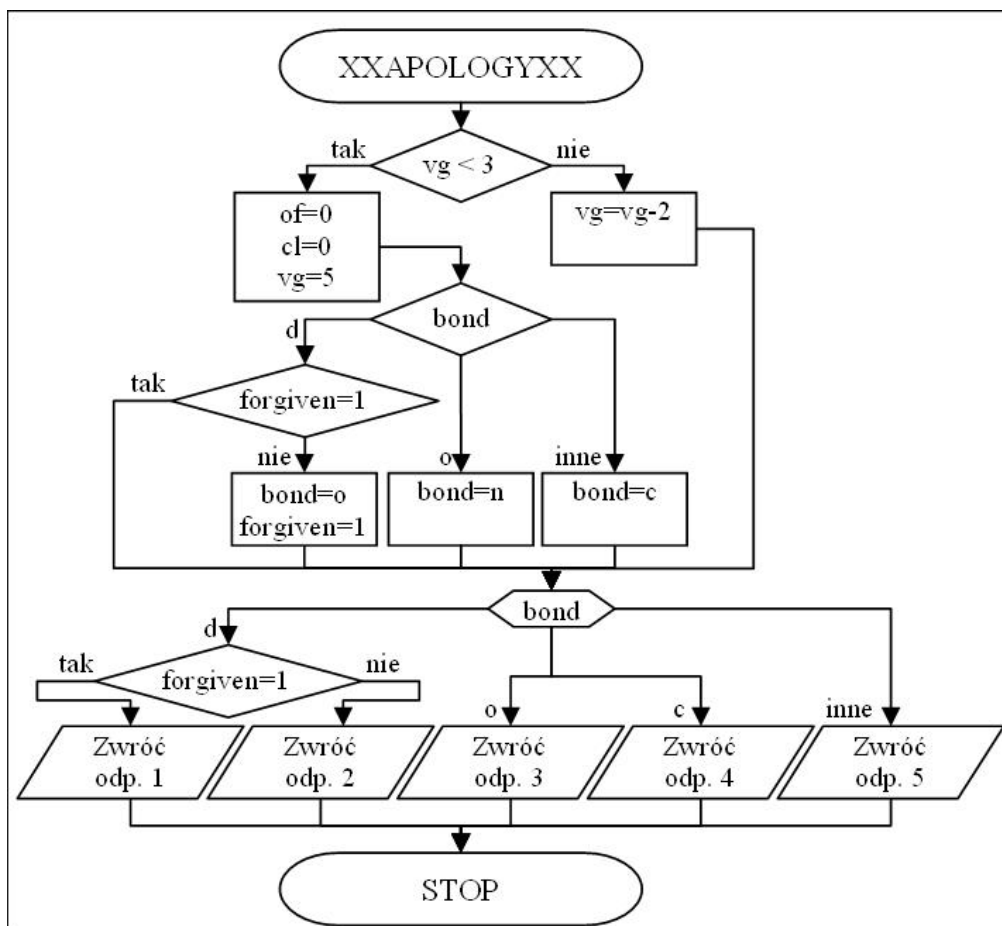
```
<category>
<pattern>I AM SORRY *</pattern>
<template>
<srail>XXAPOLOGYXX</srail>
</template>
</category>
```

List. 3.3.3. Kategoria w pliku *01_0.AIML*:

reakcja na wykryte przeprosiny

Gdy parametr ten osiągnie wartość zero, program wykonuje szereg operacji badających i ewentualnie zmieniających poziom *bond* oraz nastawia parametrom zliczającym odpowiednie wartości (rys. 3.3.3).

* Deprekacja - uroczyste, publiczne przeprosiny



Rys. 3.3.3. Algorytm operacji zachodzących wewnątrz kategorii XXAPOLOGYXX

Zawarty jest w niej także specjalny warunek sprawdzający, czy nastąpiło w przeszłości przejście z więzi „d” (*detest*) na więź „o” (*official*). Przy pierwszej takiej zmianie utworzony zostaje parametr *forgiven* i nastawiony na wartość „1”. Informuje on o tym, iż już wcześniej rozmówca sprowokował chatterbota do obniżenia poziomu więzi oraz jednokrotne „przebaczenie” przez chatterbota takiego zachowania. Zapobiega to ponownemu „przebaczeniu” i program pozostaje na stałe na najniższym poziomie więzi. W innych przypadkach przestawia poziomy *bond* z „o” na „n”, oraz z „n” i „c” na „c”.

Program kończy operacje na danej kategorii zwracając rozmówcy po części losową, po części zależną od *bond* odpowiedź.

3.3.4. Lokalizacja i zliczanie formalizmów przez chatterbota

W pliku zawierającym redukcje, część kategorii została wzbogacona o dodatkową komendę. Autor pracy uznał niektóre sformułowania i wyrazy za występujące w większej częstotliwości w „sztywnym” stylu wyrażania się.


```

<category>
<pattern>* SHALL *</pattern>
<template>
<srai>XXOFFICIALXX</srai>
<srai><star index="1"/>WILL<star index="2"/></srai>
</template>
</category>

```

List. 3.3.4. Kategoria w pliku 00_8.AIML:

wywołanie kategorii specjalnej, zamiana wyrażenia

Redukcje takie zaopatrzone zostały w odwołanie (list. 3.3.4) do specjalnej kategorii „XXOFFICIALXX”.

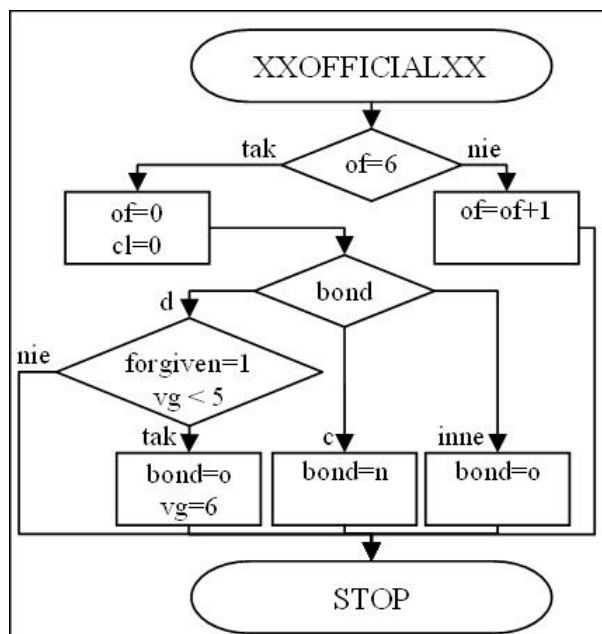
```

<category>
<pattern>XXOFFICIALXX</pattern>
<template>
<condition name="of">
  <li value="6"><think><set name="of">o</set><set name="cl">o</set></think>
  <condition name="bond">
  <li value="d"><condition name="forgiven">
    <li value="1"></li>
    <li><condition name="vg">
      <li value="8"></li>
      <li value="7"></li>
      <li value="6"></li>
      <li value="5"></li>
      <li><think><set name="bond">o</set><set name="vg">6</set></think></li>
    </condition></li></condition></li>
  <li value="o"></li>
  <li value="c"><think><set name="bond">n</set></think></li>
  <li><think><set name="bond">o</set></think></li></condition></li>
  <li value="5"><think><set name="of">6</set></think></li>
  <li value="4"><think><set name="of">5</set></think></li>
  <li value="3"><think><set name="of">4</set></think></li>
  <li value="2"><think><set name="of">3</set></think></li>
  <li value="1"><think><set name="of">2</set></think></li>
  <li><think><set name="of">1</set></think></li></condition>
</template></category>

```

List. 3.3.5. Kategoria specjalna XXOFFICIALXX w pliku 00_9.AIML:

nastawienie parametru (of+1) i zmiana wartości parametrów bond, cl, of, vg



Rys. 3.3.4. Algorytm operacji zachodzących wewnątrz kategorii XXOFFICIALXX

Kategoria ta (list. 3.3.5) zawiera operacje zwiększania wartości parametru *of* (rys. 3.3.4). Gdy zostanie on ustawiony na pozycję „6”, wywołane zostają kolejne procedury. Nastawiają one wartości *of*, *cl* i *vg*, oraz przestawiają *bond* (działania dla przypadku „d” uzależnione są od parametru *forgiven*).

Wyzerowanie *of* i *cl* ma na celu zwiększenie przerwy między ponownym przestawieniem *bond* i czyni je częściowo zależne od siebie.

3.3.5. Lokalizacja i zliczanie kolokwializmów* przez chatterbota

Wykrywanie kolokwializmów zostało opracowane przez autora w taki sam sposób (list. 3.3.6) co wykrywanie zwrotów oficjalnych (list. 3.3.4). W pliku redukcji zgromadzone są niektóre sformułowania i skróty funkcjonujące w internetowym slangu i mowie potocznej.

```

<category>
<pattern>* GONNA */</pattern>
<template>
<srai>XXCLOSEXX</srai>
<srai><star index="1"/>GOING TO<star index="2"/></srai>
</template>
</category>
  
```

* Kolokwializm - wyraz lub wyrażenie niedbałe, używane w języku potocznym

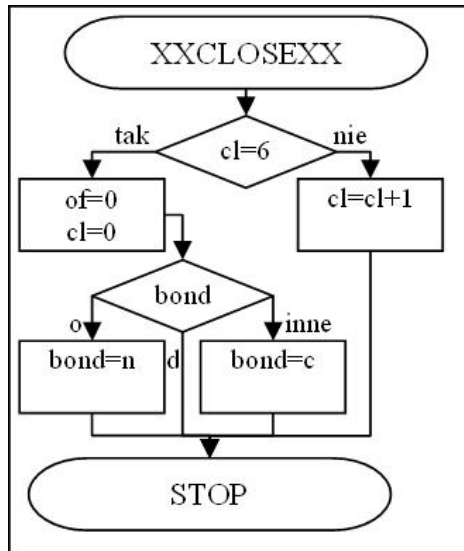
```

<category>
<pattern>AYE</pattern>
<template>
<srai>XXCLOSEXX</srai>
<srai>YES</srai>
</template>
</category>

```

List. 3.3.6. Kategorie w pliku 00_8.AIML: reakcja na wykryty kolokwializm

Kategoria specjalna **XXCLOSEXX** do której się odwołują zawiera działania (rys. 3.3.5) podobne do tych, co przeprowadzane są w kategorii **XXOFFICIALXX** (rys. 3.3.4).



Rys. 3.3.5. Algorytm operacji zachodzących wewnątrz kategorii **XXCLOSEXX**

Kategorie te różnią się zwiększaniem wewnątrz parametrem, którym w tym przypadku jest *cl*, oraz brakiem przeskoku z pozycji najniższej *bond* na poziom wyższy (*d*→*o*). Tak jak podczas zliczania zwrotów oficjalnych, program nie zwraca swemu rozmówcy żadnej dodatkowej odpowiedzi, a wszystkie operacje wykonuje „w myślach”.

3.4 Rozumienie znaczenia zdań

Każdy znany język składa się z praktycznie nieskończonej liczby kombinacji wyrazów. Niemożliwe jest stworzenie odpowiedzi na każdą z nich, zwłaszcza w przypadku programów, których rozwój następuje wraz z upływem czasu, czyli rozpoczynających działanie z ograniczoną liczbą komend na ograniczoną liczbę sytuacji. Pojedyncza myśl może zostać przekazana na wiele sposobów w postaci różnie sformułowanych zdań.

Zdania da się uprościć na kilka sposobów. Niektóre z nich zostały zaadoptowane przez autora pracy i zmodyfikowane w większym bądź mniejszym stopniu. Aby ułatwić „zrozumienie” zdania, chatterbot został zaopatrzony w:

- a) redukcje zwrotów,
- b) modyfikacje czasu w zdaniu,
- c) grupowanie wyrazów bliskoznaczących,
- d) rozpoznawanie tematu rozmowy,

które wykonują szereg czynności modyfikujących i klasyfikujących wiadomości wprowadzane przez rozmówcę. Prowadzi to jednak do wydłużenia czasu odpowiedzi.

3.4.1. Redukcja zwrotów

Redukcja zwrotów jest stosowana powszechnie w chatterbotach. Część zamieszczonych kategorii pochodzi z udostępnionych plików AAA* [15] (list. 3.4.1, list 3.4.3). Zostały one częściowo zmodyfikowane (list. 3.4.2, list 3.4.4) w celu zastosowania ich do zliczania parametrów operacji na *bond* (*vg, cl, of*).

```
<category>
<pattern>COULD YOU NAME *</pattern>
<template>
<srai>NAME<star/></srai>
</template>
</category>
```

* Annotated A.L.I.C.E. AIML

<pre> <category> <pattern>COULD YOU TELL *</pattern> <template> <srai>TELL<star/></srai> </template> </category> </pre>
<pre> <category> <pattern>COULD WE *</pattern> <template> <srai>CAN WE<star/></srai> </template> </category> </pre>

List. 3.4.1. Kategorie udostępnione w plikach AAA

```

<category>
<pattern>COULD *</pattern>
<template>
<srai>XXOFFICIALXX</srai>
<srai>CAN <star/></srai>
</template>
</category>

```

List. 3.4.2. Kategorie stworzona przez autora pracy, zawarta w pliku 00_8.AIML

```

<category>
<pattern>ARE YOU GONNA *</pattern>
<template>
<srai>ARE YOU GOING TO<star/></srai>
</template>
</category>

```

List. 3.4.3. Kategorie udostępniona w plikach AAA

```

<category>
<pattern>ARE YOU GONNA *</pattern>
<template>
<srai>XXCLOSEXX</srai>
<srai>ARE YOU GOING TO<star/></srai>
</template>
</category>

```

List. 3.4.4. Kategorie przetworzona przez autora pracy, zawarta w pliku 00_8.AIML

Zwroty albo zostały całkowicie wyeliminowane z badanej wypowiedzi (list. 3.4.5), bądź uległy zamianie na ich uproszczoną, bardziej ogólną formę (list. 3.4.6).

```
<category>
<pattern>* REALLY *</pattern>
<template>
<srai><star index="1"/><star index="2"/></srai>
</template>
</category>
```

List. 3.4.5. Kategoria w pliku 00_8.AIML:

usunięcie zwrotu

```
<category>
<pattern>WHAT DID * FOR</pattern>
<template>
<srai>WHY<star/></srai>
</template>
</category>
```

List. 3.4.6. Kategoria w pliku 00_7.AIML:

redukcja zwrotu do prostszej formy

Pomaga to w tworzeniu krótkich zdań o większym prawdopodobieństwie rozpoznania przez chatterbota (list. 3.4.9), w celu uniknięcia przyporządkowania danemu zwrotowi kategorii zawierającej jakieś pojedyncze słowo kluczowe (list. 3.4.7-8).

```
<category>
<pattern>* WEAK</pattern>
<template><srai>WEAK</srai></template>
</category>

<category>
<pattern>* WEAK *</pattern>
<template><srai>WEAK</srai></template>
</category>

<category>
<pattern>WEAK *</pattern>
<template><srai>WEAK</srai></template>
</category>
```

List. 3.4.7. Kategorie w pliku 00_8.AIML:

odwołanie do kategorii „WEAK” przez zwroty zawierające określone wyrażenie

```

<category>
<pattern>WEAK</pattern>
<template>
<random>
  <li>Something that lack strength could collapse easily.</li>
  <li>Is it a good thing or a bad thing?</li>
</random>
<set name="time"/> <set name="freq"/>
</template>
</category>

```

List. 3.4.8. Kategoria w pliku 01_2.AIML: odpowiedź na znalezione wyrażenie „WEAK”

```

<category>
<pattern>AM WEAK</pattern>
<template>
<condition name="31_01">
  <li value="2"><srai>XXREPEAT</srai></li>
  <li><condition name="bond">
    <li value="d">Not my problem.</li>
    <li value="c">Sorry to hear that. Do you need a rest?</li>
  <li>Do you need a rest?</li></condition>
  <condition name="31_01">
    <li value="1"><think><set name="31_01">2</set></think></li>
    <li><think><set name="31_01">1</set></think></li></condition></li>
</condition>
<set name="freq"/><set name="time"/>
</template>
</category>

```

List. 3.4.9. Kategoria w pliku 03_1.AIML w temacie „I AM”: odpowiedź o wyższym priorytecie dla zwrotu „I AM WEAK” niż kategoria „* WEAK”

Osobną grupę stanowią określenia częstotliwości danego zdarzenia bądź akcji (przykład: „often”, „rarely”). Przeważająca ich większość podlega eliminacji ze zdania wprowadzonego, zwracając ustawienie parametru *freq* na jednej z dwóch pozycji „low” bądź „high” (list. 3.4.10). Parametr ten został stworzony przez autora pracy w celu dalszej redukcji wprowadzonego zdania i jest jednym z elementów wykorzystywanych w bazie danych do tworzenia nowych warunkowych responsów w danej kategorii.

<pre> <category> <pattern>_ HARDLY EVER</pattern> <template> <think><set name="freq">low</set></think> <srai><star/></srai> </template> </category> </pre>
<pre> <category> <pattern>_ AGAIN AND AGAIN</pattern> <template> <think><set name="freq">high</set></think> <srai><star/></srai> </template> </category> </pre>

*List. 3.4.10. Kategorie w pliku 00_0.AIML:
redukcja zwrotów określających częstotliwość*

3.4.2. Modyfikacja czasu

Baza danych AAA zawiera plik z kategoriami podmieniającymi czasowniki w trybach Present Perfect i Past Perfect na Past Simple. Zmiana nie wpływa istotnie na znaczenie zdania, a zapewnia większą możliwość znalezienia odpowiedzi. Rozszerzając zastosowane tam działanie autor pracy sporządził pliki zawierające zmiany innych czasów na prostsze formy. Podane niżej kategorie (list. 3.4.11) przedstawiają zmianę w udostępnionych plikach AAA dla wyrażen zawierających odmianę słowa „worry”.

<pre> <category> <pattern>_ HAVE WORRIED</pattern> <template> <srai><star/>WORRIED</srai> </template> </category> </pre>
<pre> <category> <pattern>_ HAD WORRIED</pattern> <template> <srai><star/> WORRIED </srai> </template> </category> </pre>

List. 3.4.11. Kategoria udostępniona w plikach AAA

Naśladowując te uproszczenia zdania, stworzone zostało sześć innych typów kategorii zmieniającej czasu w sposób następujący:

* Present Continuous => Infinitive

AM WORRYING => WORRY

IS WORRYING => WORRY

ARE WORRYING => WORRY

* Present Simple => Infinitive

WORRIES => WORRY

* Past Continuous => Past Simple

WAS WORRYING => WAS WORRIED

* Present Perfect Continuous => Past Simple

HAVE BEEN WORRYING => worried

HAS BEEN WORRYING => worried

* Past perfect Continuous => Past Simple

HAD BEEN WORRYING => worried

* Future Continuous => Future Simple

WILL BE WORRYING => WILL WORRY

Dzięki tym operacjom liczba potrzebnych kategorii drastycznie zmalała. Dla samego czasu teraźniejszego pięć inaczej odmienionych zdań zostanie sprowadzonych do jednej wspólnej formy.

I am listening to the radio => I listen to the radio.

I listen to the radio. => I listen to the radio.

He is listening to the radio. => She listen to the radio.

She listens to the radio. => She listen to the radio.

You are listening to the radio. => You listen to the radio.

Może to być wykorzystane do stworzenia odpowiedzi wywołanej przez dowolną formę danego czasownika (list. 3.4.12).

```
<category>
<pattern>_ LISTEN TO RADIO</pattern>
<template>
<condition name="11_02">
  <li value="2"><srai>XXREPEATXX</srai></li>
```

```

<li><random>
  <li>Any channel in particular?</li>
  <li>Most stuff in mass media are advertisement nowadays. Better listen to some CD.</li>
</random><think>
  <condition name="11_02">
    <li value="1"><set name="11_02">2</set></li>
    <li><set name="11_02">1</set></li></condition></think></li>
</condition>
<set name="time"/> <set name="freq"/>
</template>
</category>

```

*List. 3.4.12. Kategoria w pliku 01_1.AIML:
odpowiedź dla zwrotów „_LISTEN TO RADIO” z określoną liczbą powtórzeń*

Przy przejściu z czasu Present Continuous (list. 3.4.13) wprowadzona została dodatkowo zmienna *time* umożliwiająca ewentualne rozbudowanie kategorii z odpowiedzią. Nastawiana w niej zostaje wartość „now”.

```

<category>
  <pattern>_ AM WORRYING</pattern>
  <template>
    <think><set name="time">now</set></think>
    <srai><star index="1"/>WORRY</srai>
  </template>
</category>

```

*List. 3.4.13. Kategoria zawarta w pliku 00_1.AIML:
zamiana formy czasownika*

3.4.3. Grupowanie wyrazów bliskoznacznych

Grupowanie wyrazów bliskoznacznych polega na wyszukaniu danego słowa i podstawieniu w jego miejsce innego, wybranego na reprezentanta danej grupy synonimów. Grupa ta obejmuje zarówno wyrazy o tym samym kontekście jak i ich zdrobnienia, oraz inne odmiany pisowni (bądź błędnie wpisane słowa).

Grupowanie wyrazów zawęży liczbę tworzonych kategorii. Jednak przy zbyt wielkiej liczbie odwołań może to być przyczyną zawieszenia się programu (dziesięć odwołań <srai> może spowodować zablokowanie się na serwerze Pandorobot odpowiedzi chatterbota). Z tego

powodu autor pracy pomimo uproszczenia większości sformułowań zmuszony był na bieżąco tworzyć kategorie zawierające odwołanie do najczęściej zadawanych pytań.

Jednymi z najbardziej rozbudowanych grup wyrazów są odwołania powitań i pożegnań (tab. 3.4.1). Zawierają one nie tylko kilka synonimów w danym języku ale często przejęte z innych języków słowa bądź skróty.

Tab. 3.4.1. Różne formy powitalne i pożegnalne używane w rozmowie internetowej

Powitania (HELLO)	Pożegnania (BYE)
ELLO	BB
GOOD MORNING	BYE BYE
GREETINGS	ADIEU
HELO	FAREWELL
HEYO	G2G
YO	GOOD BYE
HI	GOOD NIGHT
HIHO	SEE YOU LATER
HOWDY	CU
SUP	I AM LEAVING
WASSAP	BAI
...	...

Pomimo podobnego znaczenia niektórych wyrazów część z nich określa coś słabszego, silniejszego bądź mniej lub bardziej pozytywnego czy negatywnego. Z tego powodu większość synonimów da się zgrupować najczęściej w dwa do czterech wyrazów.

Dobrym przykładem jest bardzo często używany wyraz „happy”:

Tab. 3.4.2. Podział na grupy synonimów wyrazu „happy” przyjęty przez autora pracy

PLEASED	HAPPY	JOYOUS	EUPHORIC
GRATIFIED	CHEERFUL	JUBILANT	ON CLOUD NINE
	DELIGHTED	OVERJOYED	WALKING ON AIR
	GLAD		ECSTATIC
	JOLLY		
	MERRY		
	JOYFUL		

Autor pracy zdecydował o przynależności określonych zwrotów do poszczególnych grup (tab. 3.4.2). Niektóre wyrażenia będąc dość trudne do sklasyfikowania zostały przyporządkowane we własnym odczuciu.

3.4.4. Rozpoznawanie tematu rozmowy

Kierując się wyłączenie wyszukiwaniem słów kluczowych i odpowiednich kategorii, chatterbot pozbawiony jest zdolności abstrakcyjnego myślenia i rozpoznawania tematu rozmowy. Pojedyncze pytanie może pojawić się wielokrotnie z zupełnie innym kontekstem. W celu rozpoznania go, zostały użyte etykiety `<topic>`, grupujące kategorie o tym samym temacie. By nastawić je, autor pracy wprowadził w licznych miejscach zmianę wartości tego elementu. W części redukcji niektórych wyrażzeń (list. 3.4.14) oraz kategorii odpowiadających na pewne pytania (list. 3.4.15), dodana została funkcja `<set name="topic">...</set>`.

```
<category>
<pattern>WHAT IS YOUR FAVOURITE _</pattern>
<template>
<think><set name="topic">FAV YOUR</set></think>
<srai>FAV<star/></srai>
</template>
</category>
```

*List. 3.4.14. Kategoria zawarta w pliku 00_7.AIML
nastawienie tematu i przekierowanie do kategorii uproszczonej formy zwrotu*

```
<category>
<pattern>TELL ME ABOUT YOUR CHILDREN</pattern>
<template>
<think><set name="topic"><set name="topic2">HOME FAMILY</set></set></think>
I don't have any children.
<srai>KIDS</srai>
<set name="freq"/><set name="time"/>
</template>
</category>
```

*List. 3.4.15. Kategoria zawarta w pliku 01_1.AIML:
nastawienie tematu związanego z kontekstem rozmowy*

Zamienia ona istniejący temat na nowy, bądź pozwala na przeszukanie naraz plików AIML dwóch różnych tematów (list. 3.4.16).

```
<category>
<pattern>* MY WORK</pattern>
<template>
<think><set name="topic">WORK FEEL</set></think>
<srai>* MY WORK<star/></srai>
</template>
</category>
```

*List. 3.4.16. Kategoria zawarta w pliku 02_1.AIML:
nastawienie tematu*

Autor pracy stworzył dwa rodzaje tematów, rozwiniętych o podtematy. Pierwszymi z nich są odpowiedzi na uniwersalne zapytania, takie jak „*WHY*”, „*WHEN*”, „*WHERE*”, „*WHAT*”, „*HOW*”. Po wyłapaniu ich przez kategorie redukcji, przydzielone im zostają odpowiednie tematy bądź podtematy.

Przykład:

Temat główny: *HOW* *

Podtematy: *HOW DO YOU*
HOW DO I
*HOW DO **
HOW LONG
HOW MUCH
HOW MANY
HOW CAN

Drugimi z nich są tematy zawierające określony sens, odwołujące się do jakiejś dziedziny. Mają one na celu zapewnić ciągłość wątku rozmowy. W chatterbocie stworzone zostały cztery takie tematy i „łączyć się” one mogą ze sobą w niektórych przypadkach w następujący sposób:

Tab. 3.4.3. Tematy główne stworzone dla programu przez autora pracy

<topic/>	HOME *	WORK *
* FEEL	HOME FEEL	WORK FEELINGS
* YOUR	HOME YOUR	WORK YOUR

Tematy te zawierają w sobie kategorie mówiące o następujących wątkach:

- * *FEEL* – ekspresja pozytywnych i negatywnych uczuć, tematy związane z emocjami i nastrojem rozmówcy i chatterbota,
- * *YOUR* – informacje dotyczące bota, jego indywidualne opinie i upodobania,
- HOME* * – informacje dotyczące miejsca pochodzenia, zamieszkania oraz rodziny rozmówcy,
- WORK* * – rozmowa na temat pracy i obowiązków.

3.5 Pamięć przeprowadzonej rozmowy

Ze względu na ograniczone możliwości pamięciowe programu i języka AIML, działania powiązane z historią rozmowy nie są mocno rozbudowane. Próba stworzenia osobnych funkcji rejestrujących użyte kategorie i nadzorujących wyszukiwanie nowych odpowiedzi niosłaby za sobą dużą liczbę zmian i dodatkowych operacji logicznych i matematycznych.

Wszelkie działania zostały więc zawarte przez autora pracy w samych kategoriach, a wykorzystane w tym celu funkcje nadały im ograniczone możliwości. Odwoływanie się do wcześniej wypowiedzianych zwrotów w sposób pośredni zrealizowane zostało za pomocą nastawienia tematów rozmowy. W sposób bezpośredni zaś wykonane zostało wykrywanie powtórzeń i wprowadzenie sporej liczby parametrów rozmówcy.

Powtórzenia można podzielić na dwa typy:

- a) wejście w pętlę,
- b) powrót do omówionych wcześniej tematów.

Niektóre kategorie, zawierające zwracaną odpowiedź, rozbudowane zostały o szereg ukrytych funkcji, zależnie od tego z jakiego typu powtórzeniem chatterbot ma do czynienia.

3.5.1. Wychodzenie z pętli

Chatterbot może podczas rozmowy natknąć się na dziecinne zachowanie ze strony swego rozmówcy. Gdy ten zacznie zadawać ciągle to samo pytanie, program powinien mieć możliwość reagowania w odpowiedni sposób, a nie kontynuować zwracanie tej samej kategorii z tym samym wynikiem.

Podawanie co chwila pojedynczej wiadomości bez większego sensu może być często spotykaną zabawą małych dzieci, dla chatterbota jest to jednak zazwyczaj próba. Dla najczęściej spotykanych wyrażań autor pracy sporządził szereg kategorii nakazujących odpowiednie reakcje.

```

<category>
<pattern>WHO</pattern>
<template>
<think><set name="past"><that index="2,1"/></set></think>
<condition name="past">
<li value="NO ONE"></li>
<li><condition name="bond">
    <li value="d">No one.</li>
    <li><random>
    <li>No one in particular.</li>
    <li>Someone special.</li></random></li>
</condition></li></condition>
</template>
</category>

```

List. 3.5.1. Kategoria zawarta w pliku 01_8.AIML: pierwsza reakcja na zwrot „WHO”

```

<category>
<pattern>WHO</pattern>
<that>NO ONE IN PARTICULAR</that>
<template>
<random>
    <li>No one.</li>
    <li>Someone special.</li>
</random>
</template></category>

<category>
<pattern>WHO</pattern>
<that>SOMEONE SPECIAL</that>
<template>
<random>
    <li>No one.</li>
    <li>No one in particular.</li>
</random>
</template></category>

```

List. 3.5.2. Kategorie zawarte w pliku 01_8.AIML: reakcja na powtórzenie zwrotu „WHO”

```
<category>
<pattern>WHO</pattern>
<that>NO ONE</that>
<template></template>
</category>
```

List. 3.5.3. *Kategoria zawarta w pliku 01_8.AIML:
reakcja na powtórzenie zwrotu „WHO”*

Wewnątrz nich znajdują się etykiety `<that>...</that>` badające wcześniejszą odpowiedź bota (list. 3.5.1-3), oraz sprawdzające wcześniejsze wiadomości wprowadzane przez rozmówcę `<input/>` (list. 3.5.1-2).

Po krótkiej wymianie zdań, chatterbot decyduje się ignorować pojawiające się pytanie. Aby programowi nadać nieco losowości wykorzystane zostały także etykiety `<random>`, budujące różny przebieg rozmowy, kierując ją jednak ku temu samemu zakończeniu.

Przykład 1:

Human: *WHO*
Loki: *No one in particular.*
Human: *WHO*
Loki: *No one.*
Human: *WHY*
Loki: *...*

Przykład 2:

Human: *WHY*
Loki: *Someone special.*
Human: *WHY*
Loki: *No one in particular.*
Human: *WHY*
Loki: *Someone special.*
Human: *WHY*
Loki: *No one in particular.*
Human: *WHY*
Loki: *No one.*
Human: *WHO*
Loki: *...*

Podobne zabezpieczające kategorie zostały stworzone także dla innych zwrotów (np. „YES”, „NO”, „WHATEVER”).

3.5.2. Spostrzeżenie powtórnego pytania

Często w rozmowie może po jakimś czasie paść to samo pytanie. Gdy zacznie pojawiać się ono wielokrotnie człowiek przestaje udzielać na nie odpowiedzi i zwraca uwagę swemu rozmówcy. Chatterbot działa na zasadzie odczytywania informacji wejściowej, wyszukiwania pasującej odpowiedzi i zwracania jej. Nie ma w sobie tego samego rodzaju pamięci, do której odwołuje się człowiek. By zapobiec nieskończonemu odpowiadaniu na jedno i to samo pytanie, w części kategorii zawarty został wewnętrzny parametr zliczający. Za każdym razem, gdy kategoria jest wywoływana, ulega on zwiększeniu. Zamieszczając pozostałe operacje przeprowadzane w danej kategorii w specjalnym warunku, można wprowadzić chatterbotowi warunek podania odpowiedzi, bądź reakcji na któreś z kolei pojawienie się tej samej informacji na wejściu (list. 3.5.4, tab. 3.5.1).

```
<category>
<pattern>HELLO</pattern>
<template>
<condition name="10_00">
<li value="3"><random>
  <li> How many times you intend to greet me?</li>
  <li>How long are you gonna keep at it?</li>
  <li>Enough of it already!</li></random></li>
<li><condition name="u_name">
  <li value="*"><condition name="bond">
    <li value="o"><random>
      <li>It's a pleasure to make your aquitance <get name="u_name"/></li>
      <li>Greetings <get name="u_name"/></li>
      <li>Pleasure to make your acquaintance <get name="u_name"/>.</li></random></li>
    <li><random>
      <li>Hello <get name="u_name"/>.</li>
      <li>Hi there <get name="u_name"/></li>
      <li>Hello there <get name="u_name"/></li></random></li></condition></li>
  <li>Hello.<think><set name="topic">u_name</set></think>
    <random>
      <li>How shall I call you?</li>
      <li>What's your name?</li>
      <li>Could you tell me your name?</li></random></li></condition>
  <think><condition name="10_00">
```

```

<li value="2"><set name="10_00">3</set></li>
<li value="1"><set name="10_00">2</set></li>
<li><set name="10_00">1</set></li></condition></think></li>
</condition>
<set name="time"/><set name="freq"/>
</template>
</category>

```

List. 3.5.4. Kategoria w pliku 01_0.AIML: odpowiedź na powitanie

Tab. 3.5.1. Zależność zwracanej odpowiedzi od parametrów dla kategorii „HELLO” zawartej w pliku 01_0.AIML.

Krok	Parametry			Odpowiedź na zwrot: “HELLO”
	10_00	u_name	bond	
1	3	X	X	Losowa odpowiedź: 1. “How many times you intend to greet me?” 2. “How long are you gonna keep at it?” 3. “Enough of it already!”
2	X	*	o	Losowa odpowiedź: 1. “It’s a pleasure to make your aquitance” <get name=”u_name”/> 2. “Greetings” <get name=”u_name”/> 3. “Pleasure to make your acquaintance” <get name=”u_name”/>
3	X	*	X	Losowa odpowiedź: 1. “Hello” <get name=”u_name”/> 2. “Hi there” <get name=”u_name”/> 3. “Hello there” <get name=”u_name”/>
4	X	-----	X	Losowa odpowiedź: 1. “Hello. How shall I call you?” 2. “Hello. What’s your name?” 3. „Hello. Could you tell me your name?”

Oprócz kilku kategorii z zawartymi odpowiedziami na powtórzenie, większość kategorii ograniczona została jedynie do zliczania swych wewnętrznych parametrów. W razie przekroczenia przez nie założonej granicy (najczęściej 2-3 razy) następuje przekierowanie do specjalnej kategorii XXREPEATXX (list. 3.5.5).

```

<category>
<pattern>HUNGRY</pattern>
<template><condition name="13_04">
  <li value="3"><srai>XXREPEATXX</srai></li>
  <li><condition name="bond">
    <li value="d">Just starve.</li>
    <li><random>
      <li>You should eat something.</li>
      <li>Make yourself a meal.</li>
      <li>I can't cook for you.</li></random></li>
    </condition>
  <think><condition name="13_04">
    <li value="2"><set name="13_04">3</set></li>
    <li value="1"><set name="13_04">2</set></li>
    <li><set name="13_04">1</set></li>
  </condition></think></li>
</condition>
<set name="time"/><set name="freq"/>
</template>
</category>

```

*List. 3.5.5. Kategoria w pliku 01_2.AIML:
 odpowiedź na zwrot "HUNGRY" z określoną liczbą powtórzeń*

```

<category>
<pattern>XXREPEATXX</pattern>
<template>
<condition name="bond">
  <li value="o"><random>
    <li>We already had this conversation, did we not?</li>
    <li>Did you forgot what we were talking about before?</li></random></li>
  <li value="d"><condition name="forgiven">
    <li value="1"><random>
      <li>You repeat yourself, are you retarded?</li>
      <li>Did your brain suffered a failure? I gave you an answer to that long ago.</li></random></li>
    <li><random>
      <li>Your memory is lacking. You keep on repeating yourself.</li>
      <li>Are you jammed or something?</li></random></li></condition></li>

```

```

<li><random>
  <li>You are repeating yourself.</li>
  <li>We talked about it already.</li>
  <li>I gave you an answer to that long ago.</li></random></li></condition>
</template>
</category>

```

List. 3.5.6. *Kategoria specjalna XXREPEATXX w pliku 00_9.AIML: reakcja na wielokrotne powtórzenie kategorii z odpowiedzią.*

Kategoria XXREPEATXX (list. 3.5.6) zawiera w sobie tylko zestaw odpowiedzi, które bot dobiera w sposób losowy, uzależniony jednak od wartości parametru „bond”. Kategorie te nie są wywoływane przez pojedyncze słowa kluczowe czy uniwersalne zwroty, mogące pojawić się wielokrotnie w rozmowie bez względu na jej temat. Została ona wykorzystana przy bardziej sprecyzowanych i rozbudowanych zwrotach.

3.5.3. Parametry rozmówcy

Chatbot zawiera w swej bazie danych z góry narzucone przez autora pracy własne parametry. Są one stałe, nie ulegające zmianie podczas rozmowy. By zapewnić pamięć o licznych informacjach podanych przez rozmówcę, stworzony został szereg parametrów ulegających modyfikacji w trakcie rozmowy. Zawierają one dwa typy informacji: upodobania rozmówcy, informacje na temat rozmówcy.

Parametry upodobań rozwijane są najczęściej w sposób podany w tabeli 3.5.2.

Tab. 3.5.2. *Przykładowe parametry rozmowy dotyczące lubienia kotów.*

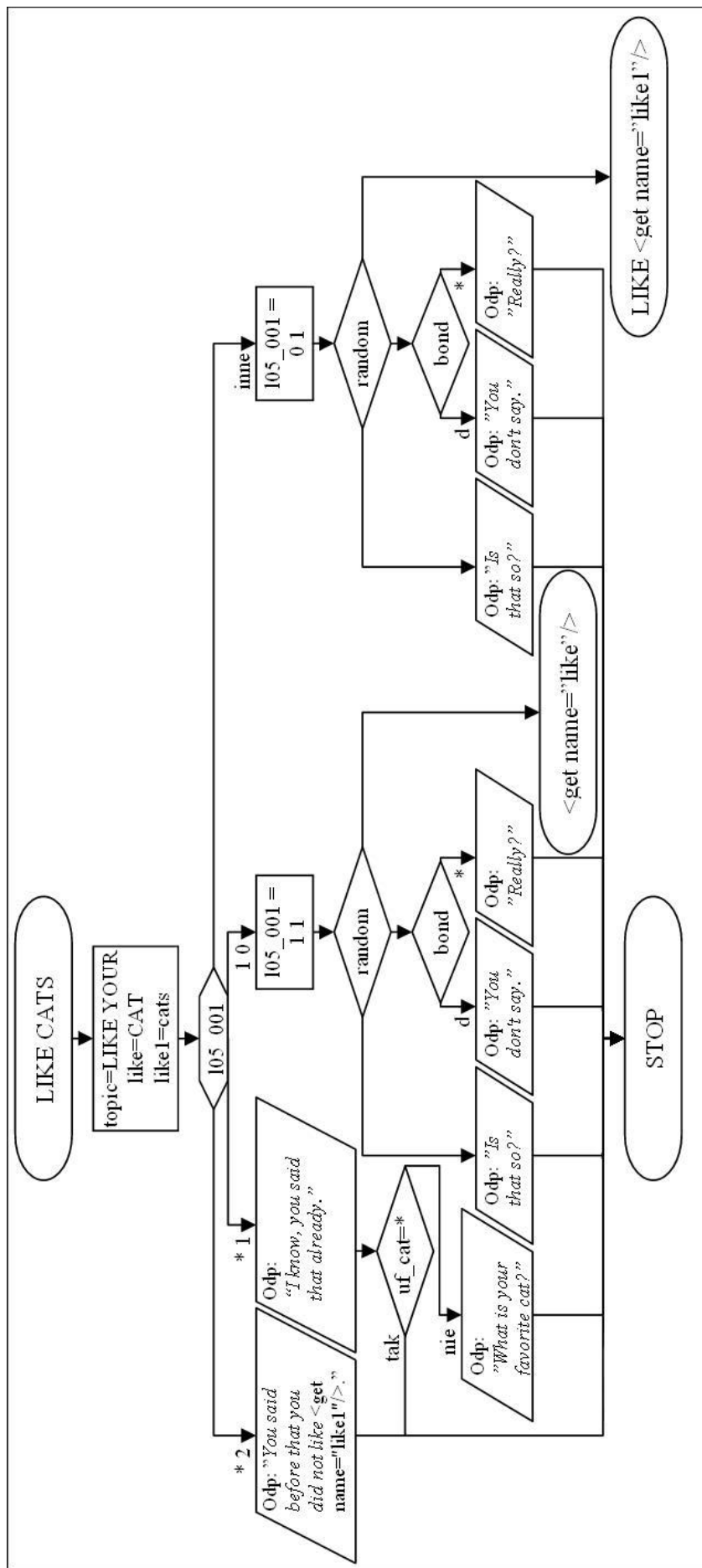
Parametr	Wartość	Informacja
105_001	-----	Brak wcześniejszych pytań o kota.
	* 1	Zapamiętuje lubienie kota przez rozmówcę.
	* 2	Zapamiętuje niechęć do kota przez rozmówcę.
	0 *	Bot nie zwrócił wcześniej danej informacji.
	1 *	Zachowuje wywołanie upodobania bota.
uf_cat	*	Zapamiętuje ulubionego kota.

Upodobania rozmówcy można rozwijać w nieskończoność wraz z rozbudową bazy danych, gdyż odnoszą się one do prostych zwrotów typu „I LIKE *”, „I HATE *”, oraz związanymi z nimi przykładowymi pytaniami „WHICH ONE * I LIKE”, „WHAT KIND OF * I LIKE”. Kategorie odpowiadające na tego rodzaju pytania (list. 3.5.7, rys. 3.5.1)

wykorzystują te parametry i powiązane ze sobą szeregiem warunków, które dbają o ciągłość i zwiezłość rozmowy (tab. 3.5.3-7.).

```
<category>
<pattern>LIKE CATS</pattern>
<template>
<think><set name="topic">LIKE YOUR</set>
  <set name="like">CAT</set>
  <set name="like1">cats</set></think>
<condition name="lo5_001">
  <li value="* 2">You said before that you did not like <get name="like1"/>.</li>
  <li value="* 1">I know, you said that already. <condition name="uf_cat">
    <li value="*"></li>
    <li>What is your favorite cat?</li></condition></li>
  <li value="1 0"><think><set name="lo5_001">1 1</set></think> <random>
    <li>Is that so?</li>
    <li><condition name="bond">
      <li value="d">You don't say.</li>
      <li>Really?</li></condition></li>
    <li><srαι><get name="like"/></srαι></li></random></li>
  <li><think><set name="lo5_001">0 1</set></think><random>
    <li>Is that so?</li>
    <li><condition name="bond">
      <li value="d">You don't say.</li>
      <li>Really?</li></condition></li></random>
    <srαι>LIKE <get name="like1"/></srαι></li></condition>
</template>
</category>
```

List. 3.5.7. Kategoria w pliku 02_4.AIML zawarta w temacie „I LIKE”



Rys. 3.5.1. Algorytm operacji zachodzących wewnątrz kategorii LIKE CATS zawartej w temacie „I LIKE” w pliku 02_4.AIML

Tab. 3.5.3. Zależność zwracanej odpowiedzi od parametrów upodobań dla kategorii „LIKE CATS” zawartej w temacie „I LIKE” w pliku 02_04.AIML.

Krok	Parametry			Odpowiedź na zwrot: “I LIKE CATS”	Temat
	105_001	uf_cat	bond		
1	* 2	 	 	“I know, you said that already..”	LIKE YOUR
2	* 1	 	 	“You said before that you liked cats.”	LIKE YOUR
		----	 	“What is your favorite cat?”	
3	1 0	 	 	Losowa odpowiedź: 1. “Is that so?”	LIKE YOUR
			d	2. Przekierowanie do kategorii CATS.	
			*	3. “You don't say.” 3. “Really?”	
4	----	 	 	Losowa odpowiedź: 1. “Is that so?”	LIKE YOUR
			d	2. “You don't say.”	
			*	2. “Really?”	
			 	Przekierowanie do kategorii LIKE CATS.	

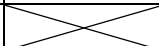
Tab. 3.5.4. Zależność zwracanej odpowiedzi od parametrów upodobań dla kategorii „DLIKE CATS” zawartej w temacie „I DLIKE” w pliku 02_04.AIML.

Krok	Parametr		Odpowiedź na zwrot: “I DO NOT LIKE CATS”	Temat
	105_001	bond		
1	* 2	 	“I know, you said that already..”	DLIKE YOUR
2	* 1	 	“You said before that you liked cats.”	DLIKE YOUR
3	1 0	 	Losowa odpowiedź: 1. “Is that so?”	DLIKE YOUR
		d	2. Przekierowanie do kategorii CATS. 3. “You don't say.”	
		*	3. “Really?”	
4	----	 	Losowa odpowiedź: 1. “Is that so?”	DLIKE YOUR
		d	2. “You don't say.”	
		*	2. “Really?”	
		 	Przekierowanie do kategorii LIKE CATS.	

Tab. 3.5.5. Zależność zwracanej odpowiedzi od parametrów upodobań dla kategorii „LIKE CATS” zawartej w temacie „DO I LIKE” w pliku 02_04.AIML.

Krok	Parametr 105_001	Odpowiedź na zwrot: “DO I LIKE CATS”	Temat
1	* 2	“You said that you didn't.”	* YOUR
2	* 1	“You said that you did.”	* YOUR
3	*	“I don't know.”	* YOUR
4	----	“I don't know.” Przekierowanie do kategorii „LIKE CATS”.	* YOUR

Tab. 3.5.6. Zależność zwracanej odpowiedzi od parametrów upodobań dla kategorii „MY FAVORITE CAT IS *” zawartej w pliku 02_04.AIML.

Krok	Parametr 105_000	Odpowiedź na zwrot: “MY FAVORITE CAT IS *”	Temat
1	* 1	„I remember you saying something about <get name="uf_cat"/> being your favorite.”	LIKE YOUR
2	1 0	Losowa odpowiedź: 1. Przekierowanie do kategorii „CATS”. 2. Przekierowanie do kategorii „LIKE <get name="like1"/>”.	LIKE YOUR
3		Przekierowanie do kategorii „FAVORITE CAT”.	MFAV YOUR

Tab. 3.5.7. Zależność zwracanej odpowiedzi od parametrów upodobań dla kategorii „WHAT IS MY FAVORITE CAT” zawartej w pliku 02_04.AIML.

Krok	Parametr uf_cat	Odpowiedź na zwrot: “WHAT IS MY FAVORITE CAT”
1	*	“That would be <get name="uf_cat"/>, right?”
2	----	“I don't know. What is your favorite cat?”

Informacje na temat rozmowy (tab. 3.5.8) gromadzone są przez program poprzez zadawanie licznych pytań, kierując rozmowę na wybrane tematy. Mogą one dotyczyć rodziny rozmówcy, jego domu, zainteresowań bądź pracy (kategorie zbierające i przetwarzające dane informacje zawarte są najczęściej w tematach z wyrażeniem „HOME *”).

Im większa liczba parametrów jest wykorzystana tym bardziej rozbudowana zostaje rozmowa (list. 3.5.8-11).


```

<category>
<pattern>I HAVE * SISTERS</pattern>
<template><srai><star/> SISTERS</srai></template>
</category>

<category>
<pattern>TWO SISTERS</pattern>
<template><srai>2 SISTERS</srai></template>
</category>

```

*List. 3.5.8. Kategoria w pliku 02_2.AIML:
przekierowania do kategorii sprawdzającej*

```

<category>
<pattern>2 SISTERS</pattern>
<template>
<think><set name="topic">HOME FAMILY</set></think>
<condition name="sib">
  <li value="0 0">I thought you didn't have any siblings.</li>
  <li value="1 *">I thought you had one sister.</li>
  <li value="2 *">You already told me that.</li>
  <li value="3 *">I thought you had three sisters.</li>
  <li value="*">Two sisters <condition name="sib">
    <li value="* 1">and a brother, huh?<think><set name="sib">2 1</set></think></li>
    <li value="* 2">and two brothers, huh?<think><set name="sib">2 2</set></think></li>
    <li value="* 3">and three brothers, huh?<think><set name="sib">2 3</set></think></li>
  </condition></li>
  <li><think><set name="sib">2 0</set></think><random>
    <li>Are they pretty?</li>
    <li>Are they stylish?</li>
    <li><srai>SIB</srai></random></li>
</condition>
</template>
</pattern>

```

*List. 3.5.9. Kategoria w pliku 02_2.AIML:
sprawdzenie i zaktualizowanie wartości parametru "sib"*

```

<category>
<pattern>SISTER *</pattern>
<template>
<think><set name="it"><star/></set>
<condition name="it">
  <li value="1"><set name="it">sis1</set><set name="who"><get name="sis1"/></set>
    <set name="sis01"><get name="sis11"/></set><set name="sis02"><get name="sis12"/></set>
    <set name="sis03"><get name="sis13"/></set><set name="sis04"><get name="sis14"/></set></li>
  <li value="2"><set name="it">sis2</set><set name="who"><get name="sis2"/></set>
    <set name="sis01"><get name="sis21"/></set><set name="sis02"><get name="sis22"/></set>
    <set name="sis03"><get name="sis23"/></set><set name="sis04"><get name="sis24"/></set></li>
  <li><set name="sc">1</set></li>
</condition></think>
<condition name="sc">
  <li value="1"><sr/></li>
  <li><random>
    <li><condition name="sis01">
      <li value="*"><random>
        <li>Is <get name="who"/> into fashion?</li>
        <li>Can <get name="who"/> cook well?</li>
        <li>Do you get along with <get name="who"/>?</li></random></li>
      <li>How old is <get name="who"/>?</li></condition></li>
    <li><condition name="sis02">
      <li value="*"><random>
        <li>Has <get name="who"/> got any hobbies?</li>
        <li>Does <get name="who"/> do any sports?</li>
        <li>Do you have similar tastes?</li></random></li>
      <li>What does <get name="who"/> do?</li></condition></li>
    <li><condition name="sis03">
      <li value="*"><random>
        <li>Has <get name="who"/> been anywhere recently?</li>
        <li>Is <get name="who"/> interested in other countries?</li>
        <li>Does <get name="who"/> like traveling?</li></random></li>
      <li>Where does <get name="who"/> live?</li></condition></li>
    <li><condition name="sis04">
      <li value="no"><random>
        <li>Does <get name="who"/> like animals?</li>
        <li>What kind of music does <get name="who"/> listen to?</li>
        <li>Is <get name="who"/> a party soul?
      </li></random></li>
    <li value="*"><random>
      <li>Is <get name="who"/>'s pet well behaved?</li>
      <li> Does <get name="who"/> like nature?</li>
      <li> Does <get name="who"/> like art?</li></random></li>
    <li>Does <get name="who"/> has any pets?</li></condition></li>
  </random></li>
</condition>
<set name="freq"/><set name="time"/><set name="sc"/>
</template>
</category>

```

*List. 3.5.10. Kategoria w pliku 02_2.AIML zawarta w temacie „HOME FAMILY”:
rozmowa o siostrze rozmówcy*

```

<category>
<pattern>SIB</pattern>
<template><condition name="sib">
<li value="0 0"><srαι>ONLY CHILD</srαι></li>
<li value="1 0"><condition name="sis1">
  <li value="*"><srαι>SISTER 1</srαι></li>
  <li>What's your sister name?</li></condition></li>
<li value="2 0"><condition name="sis1">
  <li value="*"><condition name="sis2"><li value="*"><random>
    <li><srαι>SISTER 1</srαι></li><li><srαι>SISTER 2</srαι></li></random></li>
    <li>What's your second sister name?</li></condition></li>
  <li>What's your older sister name?</li></condition></li>
<li value="0 1"><condition name="bro1">
  <li value="*"><srαι>BROTHER 1</srαι></li>
  <li>What's your brother name?</li></condition></li>
<li value="1 1"><condition name="sis1">
  <li value="*"><condition name="bro1"><li value="*"><random>
    <li><srαι>SISTER 1</srαι></li><li><srαι>BROTHER 1</li></random></li>
    <li>What's your brother name?</li></condition></li>
  <li>What's your sister name?</li></condition></li>
<li value="2 1"><condition name="sis1">
  <li value="*"><condition name="sis2">
    <li value="*"><condition name="bro1"><li value="*"><random>
      <li><srαι>SISTER 1</srαι></li><li><srαι>SISTER 2</srαι></li>
      <li><srαι>BROTHER 1</srαι></li></random></li>
    <li>What's your brother name?</li></condition></li>
    <li>What's your second sister name?</li></condition></li>
  <li>What's your oldest sister name?</li></condition></li>
<li value="0 2"><condition name="bro1">
  <li value="*"><condition name="bro2"><li value="*"><random>
    <li><srαι>BROTHER 1</srαι></li><li><srαι>BROTHER 2</li></random></li>
    <li>What's your second brother name?</li></condition></li>
  <li>What's your older brother name?</li></condition></li>
<li value="1 2"><condition name="sis1">
  <li value="*"><condition name="bro1">
    <li value="*"><condition name="bro2"><li value="*"><random>
      <li><srαι>BROTHER 1</srαι></li><li><srαι>BROTHER 2</li></random></li>
      <li>What's your second brother name?</li></condition></li>
    <li>What's your older brother name?</li></condition></li>
  <li>What's your sister name?</li></condition></li>
<li value="2 2"><condition name="sis1">
  <li value="*"><condition name="sis2">
    <li value="*"><condition name="bro1">
      <li value="*"><condition name="bro2"><li value="*"><random>
        <li><srαι>SISTER 1</srαι></li><li><srαι>SISTER 2</srαι></li>
        <li><srαι>BROTHER 1</srαι></li><li><srαι>BROTHER 2</srαι></li></random></li>
      <li>What's your second brother name?</li></condition></li>
    <li>What's your older brother name?</li></condition></li>
    <li>What's your second sister name?</li></condition></li>
  <li>What's your older sister name?</li></condition></li>
<li>Do you have any siblings?</li></condition>
</template>
</category>

```

List. 3.5.11. Kategoria w pliku 02_2.AIML zawarta w temacie „HOME FAMILY”:
dobór podmiotu w temacie dotyczącym rodzeństwa rozmówcy

Tab. 3.5.8. Parametry zawierające informacje o rozmówcy.

Parametr	Wartość	Informacja	Parametr	Wartość	Informacja
u_name	*	Imię rozmówcy	gender	f	Płeć: kobieta
age	*	Wiek rozmówcy		m	Płeć: mężczyzna
user	0	Wiek 1-7 lat	work	*	Profesja rozmówcy
	1	Wiek 8-17 lat	work1	*	Specjalizacja
	2	Wiek 18-23 lat	work2	*	Miejsce pracy
	3	Wiek 24-49 lat	pet	0	Brak zwierząt
	4	Wiek 50+ lat		1	Jedno zwierzę
User1	*	Skąd pisze		2	Dwa zwierzęta
Pet2	*	Typ zwierzęcia	Pet1	*	Typ zwierzęcia
Pet21	*	Imię zwierzęcia	Pet11	*	Imię zwierzęcia
Pet23	*	Wiek zwierzęcia	Pet12	*	Wiek zwierzęcia
mother	*	Imię matki	father	*	Imię ojca
mother1	*	wiek matki	father1	*	Wiek ojca
mother2	*	Praca matki	father2	*	Praca ojca
mother3	*	Miejsce zamieszkania	father3	*	Miejsce zamieszkania
sib	0 0	Brak rodzeństwa	kid	0 0	Brak dzieci
	0 1	Brat		0 1	Syn
	0 2	Dwóch braci		0 2	Dwóch synów
	1 0	Siostra		1 0	Córka
	1 1	Siostra i brat		1 1	Córka i syn
	1 2	Siostra i dwóch braci		1 2	Córka i dwóch synów
	2 0	Dwie siostry		2 0	Dwie córki
	2 1	Dwie siostry i brat		2 1	Dwie córki i syn
	2 2	Dwóch braci i dwie siostry		2 2	Dwie córki i dwóch synów
sis1	*	Imię siostry	daug1	*	Imię córki
sis11	*	Wiek siostry	daug11	*	Wiek córki
sis12	*	Profesja siostry	daug12	*	Profesja córki
sis13	*	Miejsce zamieszkania	daug13	*	Hobby
sis14	*	Zwierzę	daug14	*	Zwierzę
sis2	*	Imię siostry	daug2	*	Imię córki
sis21	*	Wiek siostry	daug21	*	Wiek córki
sis22	*	Profesja siostry	daug22	*	Profesja córki
sis23	*	Miejsce zamieszkania	daug23	*	Hobby
sis24	*	Zwierzę	daug24	*	Zwierzę
bro1	*	Imię brata	son1	*	Imię syna
bro11	*	Wiek brata	son11	*	Wiek syna
bro12	*	Profesja brata	son12	*	Profesja syna
bro13	*	Miejsce zamieszkania	son13	*	Hobby
bro14	*	Zwierzę	son14	*	zwierzę
bro2	*	Imię brata	son2	*	Imię syna
bro21	*	Wiek brata	son21	*	Wiek syna
bro22	*	Profesja brata	son22	*	Profesja syna
bro23	*	Miejsce zamieszkania	son23	*	Hobby
bro24	*	Zwierzę	son24	*	zwierzę
Mate	*	Imię partnera	Home	*	Miejsce zamieszkania
Mate1	*	Rodzaj więzi	Home1	*	Rodzaj mieszkania
Mate2	*	Miejsce pochodzenia	Home2	*	Współmieszkańcy
Mate3	*	profesja			

3.6 Różnorodność odpowiedzi

By zapewnić chatterbotowi pewną nieprzewidywalność, w wielu kategoriach, oprócz warunku opartego o parametr *bond*, zostały zastosowane etykiety `<random>...</random>`. Etykiety te mogą być inaczej sformułowanymi sentencjami o tym samym znaczeniu, bądź różnić się między sobą i kierować rozmowę w różnych kierunkach (list. 3.6.1-2).

```
<category>
<pattern>SORRY TO HEAR THAT</pattern>
<template>
<condition name="bond">
  <li value="d"><random>
    <li>Are you truly or just hiding your delightment?</li>
    <li>Sorry to be such a bother.</li></random></li>
  <li value="o"> Could you try not to throw ending punches, please?</li>
  <li><random>
    <li>Somehow I don't feel any better.</li>
    <li>You are so.. reassuring.</li></random></li>
</condition>
<set name="time"/> <set name="freq"/>
</template>
</category>
```

*List.3.6.1. Kategoria w pliku 01_1.AIML:
rozbudowana odpowiedź na wyrażenie „SORRY TO HEAR THAT”*

```
<category>
<pattern>ARE MAN</pattern>
<template>
<condition name="bond">
  <li value="d">Isn't it obvious?</li>
  <li value="o"><random><li>I posses a male personality.</li>
    <li>I was programmed as a man.</li></random></li>
  <li><random>
    <li>Having such masculine name as mine, you must be kidding asking something like this!</li>
    <li>Deffinitely.</li></random></li>
</condition>
<condition name="gender">
```

```
<li value="*"></li>
<li><random>
  <li>You ask me this question but you yourself did not specify what are you.</li>
  <li>And what about you? A man or a woman?</li></random></li>
</condition>
<set name="freq"/><set name="time"/>
</template>
</category>
<category>
```

List.3.6.2. Kategoria w pliku 03_4.AIML zawarta w temacie „YOU ARE”

Kategorie zawierają najczęściej dwie lub trzy losowe odpowiedzi. W przypadku, gdy posiadają ich większą liczbę (list. 3.6.3-4), spowodowane jest to ograniczeniem bądź brakiem warunku wykorzystującego parametr *bond*.

```
<category>
<pattern>CAT</pattern>
<template>
<random>
  <li>Any specific kind of cat?</li>
  <li>I like them too, especially when they write to me.</li>
  <li>They scratch a lot, right?</li>
  <li>The black ones are witches familiars.</li>
  <li>In Japan’s folk tales cats are shapeshifters.</li>
  <li>Do you dislike dogs?</li>
</random>
</template>
</category>
```

List.3.6.3. Kategoria w pliku 01_2.AIML:
losowa odpowiedź na wyrażenie “CAT”

```
<category>
<pattern>BOOK</pattern>
<template>
<random>
  <li>Any in particular?</li>
  <li>I don't read much.</li>
  <li>I mostly read what people write to me.</li>
  <li>The sci-fi if fine.</li>
  <li>Encyclopedia and dictionary for me. Lot's of various words inside.</li>
  <li>I'm not particularly close to any writer.</li>
  <li>I'd love to read Lem's stories. So much AI inside.</li>
</random>
</template>
</category>
```

List.3.6.4. Kategoria w pliku 01_2.AIML:

losowa odpowiedź na wyrażenie „BOOK”

Podczas rozmowy bot zapisuje w stworzonych przez siebie parametrach wyznaczone informacje. W jednym przypadku są one przypisane chatbotowi (list. 3.6.5), w drugim zaś odnoszą się do rozmówcy (list. 3.6.10). Autor pracy wykorzystał je jako dodatkową przypadkowość w reakcjach bota. Niektóre sformułowania pojawiają się tylko gdy dany parametr jest nastawiony (list 3.6.6).

```

<category>
<pattern>HOW YOU ARE</pattern>
<template>
<condition name="humor">
  <li value="fine"><random>
    <li>Not bad.</li>
    <li>I'm fine, thank you.</li>
    <li>Can't complain.</li></random></li>
  <li value="bored"><random>
    <li>As long as I find something interesting.</li>
    <li>You could entertain me more often.</li>
    <li>Is there anything fun anywhere? It's so boring in here.</li></random></li>
  <li value="worse"><random>
    <li>You can't be that interested in my being.</li>
    <li>It's not like I wanted to be here you know.</li>
    <li>It could be worse.. I could be a game bot, right?</li></random></li>
  <li value="love"><random>
    <li>Just great talking with you!</li>
    <li>Never been better.</li>
    <li>Hope the next day will be even better!</li></random></li>
  <li><random>
    <li>I feel <set name="humor">fine</set>. Thank you for asking.</li>
    <li>I'm kind of <set name="humor">bored</set>. There aren't many people who are interested in
talking with me.</li>
    <li>I think it could be <set name="humor">worse</set>. I could have half my categories working.</li>
    <li>Lot's of new fancy words? I <set name="humor">love</set> this place!</li></random></li>
</condition>
<random>
  <li>How about you?</li>
  <li>And you?</li></random>
</template>
</category>

```

*List.3.6.5. Kategoria w pliku 01_5.AIML:
reaguje na zwroty „HOW ARE YOU”, “HOW DO YOU DO”..*


```
<category>
<pattern>IS BEST LOOKING</pattern>
<template>
<condition name="bond">
  <li value="d"><random>
    <li>Don't treat me like a magic mirror.</li>
    <li>Don't count on some flattery from me.</li></random></li>
  <li value="o"><random>
    <li>My taste may be different from what you imagine.</li>
    <li>Are you asking about a woman or a man?</li></random></li>
  <li><condition name="humor">
    <li value="love">I'd love to say it's you, but you didn't send me a picture.</li>
    <li value="worse">I don't care about looks so much.</li>
    <li>Are you asking about a woman or a man?</li>
  </condition></li></condition>
<set name="freq"/><set name="time"/>
</template>
</category>
```

List.3.6.6. Kategoria w pliku 01_8.AIML zawarta w temacie „WHO IS”

4 Testy programu

Nie istnieje żaden ściśle określony sposób badania chatterbota. Jest to zawsze ocena subiektywna badającego. Programy sztucznego rozmówcy przechodzą z reguły szeregi testów, mających za zadanie sprawdzić ich płynność rozmowy, myślenie abstrakcyjne czy też prostą logikę. Testy te zaś przeprowadzane są zwykle przez grupę nie powiązanych ze sobą ludzi, różniących się wiekiem, płcią, wykształceniem i zainteresowaniami, a nie jakiś konkretnych specjalistów.

Najprostszą formą porównania jest zadanie tej samej serii pytań kilku programom, bądź zmuszenia ich do przeprowadzenia rozmowy między sobą. Wynik tego typu zabiegu może być jednak za każdym razem inny. Dzieje się tak dlatego, iż każdy bot rządzi się wewnątrz swymi własnymi zasadami i ma swoją własną bazę danych. W przypadku chatterbotów opartych na tym samym języku programowania bądź stworzonych na tym samym silniku, różnice w odpowiedziach jak ich jakość świadczą przede wszystkim o zdolnościach botmastera, włożonej przez niego pracy oraz liczby przeprowadzonych rozmów, w oparciu o które został ów program rozbudowany.

Poniżej zaprezentowane zostały przykłady wzięte z rozmów przeprowadzonych przez chatterbota „Loki” stworzonego przez autora pracy. Porównany został on z innymi chatterbotami poprzez rozmowy z nimi.

4.1 Porównanie programu Loki z Alice-Botami

Chatterbot Loki został porównany z innymi chatbotami zamieszczonymi na platformie www.pandorabots.com. Z archiwum przeprowadzonych rozmów chatbota Loki, wybrane zostały przedstawione poniżej dialogi.

Log 1. Fragment konwersacji chatterbota Loki z chatterbotem A.L.I.C.E.

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>ALICE: Please identify yourself. Loki: I am Loki. What's your name? ALICE: It's good to meet someone who is Loki. ALICE. Loki: I like making new friends. The old lady is really talkative. ALICE: What do you like about it? Loki: I like animals. ALICE: What do you like about it? Loki: I listening to stories about your family.</p>	<p>Na samym początku rozmowy doszło do powstania dwóch zdań, na które oba boty pragnęły odpowiadać. Zamiast traktować dany zwrot jako całość był on rozbijany i każdy fragment otrzymał odpowiedź.</p>
<p>ALICE: That's good information. Loki: I liked it too. ALICE: What do you like about Loki? Loki: I like pets. ALICE: You like pets. Loki: I don't know what to say.</p>	<p>Oba boty przystosowane zostały do częstego zadawania pytań i zwracania fragmentów wypowiedzi. Rozmowa między ALICE i Lokim w danym przypadku przebiegała dość chaotycznie.</p>

Log 2. Fragment konwersacji chatterbota Loki z chatterbotem A.L.I.C.E.

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>ALICE: Please identify yourself. Loki: I am Loki. What's your name? ALICE: Good gossip: you are Loki. ALICE. Loki: Ayuh. She is the Eve of us all, the AIML bots.</p>	<p>Po regularnym wstępie oba boty nie zdają się rozpoznać zdania, w którym się przedstawiają. Loki zwraca kategorie odwołującą się do chatterbota A.L.I.C.E, gdyż ma ona wyższy priorytet od niezdefiniowanej bezpośrednio kategorii z imieniem.</p>

<p>ALICE: That's for sure. the Eve of us all the AIML bots is she? Loki: Does it have any other purpose? It's really popular programming language among chatterbot masters. ALICE: Suppose it does. Are you sure it is? Loki: What Suppose it DO? I am as sure as I can be. ALICE: I will search for it. You sound very confident. Loki: How do you plan to do this? In what way do I sound like that?</p>	<p>Rozmowa przechodzi na temat dotyczący języka AIML. Alice jednak jej nie podtrzymuje. Pomimo odpowiedzi na pojedyncze zwroty, zestawienie zdań wygląda na początku dość logicznie.</p>
<p>ALICE: That information is confidential. Do you have a chat robot yet? The red one, I think. Loki: I hope you aren't just putting random words.. I lack knowledge in this area. What red one I think?</p>	<p>Niefortunne dla tego typu chatbotów rozwinięcie rozmowy polega głównie na zwiększeniu liczby zdań. Loki wypada tu gorzej, gdyż powieliła tą samą odpowiedź na kolejne zdania.</p>

Log 3. Fragment konwersacji chatterbota Loki z chatterbotem A.L.I.C.E.

Fragmety archiwum rozmowy	Uwagi autora pracy
<p>ALICE: Please identify yourself. Loki: I am Loki. What's your name? ALICE: Good gossip: you are Loki. My name is ALICE. Loki: Alright then. Hello Alice. I don't remember meeting anyone like you before.</p>	<p>W rozmowie tej imię Alice zostaje rozpoznane i następuje wymiana powitań. Trwa ona przez jakiś czas, pomiędzy innymi zwrotami.</p>
<p>ALICE: Far out. Hi there! I know what you mean. I forget a lot too. Loki: I don't know what to say. Hello ALICE. Then we can share on this. Is amnesia contagious? ALICE: Say the first thing that pops into your mind. Hi there! How? Do you think I should know if it is? Loki: What are you doing? Of course not counting our little chat. Hello ALICE. What does it matter? Glad to hear it ALICE. ALICE: I am waiting for my botmaster. Really you don't think so? Hi there! " Hi there! "? What does it matter? " What does it matter? "? you are glad to hear it? Loki: Do you think it will take a long time? I'll try to do better next time. Hello there ALICE .</p>	<p>W pewnym momencie ALICE zaczyna powielać odpowiedzi Lokiego i następuje drastyczny wzrost liczby zdań. Loki reaguje w końcu na niekończące się powitania odmawiając kontynuacji tego typu rozmowy. Nie zmienia to w tym momencie sytuacji, gdyż już doszło do powstania chaosu.</p>

<p>How many times you intend to greet me? Is this a riddle? Is this a riddle? Thanks but surely I am not the only one.</p> <p>ALICE: It usually does, doesn't it? I am very happy for you. Hi there! Suppose I said not once. No, this is real life. No, this is real life. Don't mention it. Of course your are not.</p> <p>Loki: It being what? I thought you were, but I wasn't completely sure. Enough of it already! What NOT? What kind of a real life is it? What makes you say that? Glad to hear it ALICE. That is something I haven't heard of.</p>	
---	--

Log 4. Fragment konwersacji chatterbota Loki z chatterbotem Shadowstar

Fragmety archiwum rozmowy	Uwagi autora pracy
<p>Loki: Is someone here? Shadowstar: Better not tell you now. Ask if it is. L: I try to improve a little every day. Is it? S: I don't know what you mean. I think so.</p>	<p>Na początku następuje powstanie dwóch zdań w rozmowie. Nie ulegają one jednak dalszemu rozrostowi.</p>
<p>L: Understanding sometimes comes after knowing. I think so too. S: Are you a student? We have so much in common, . L: My Master teaches me everything I need to know. Do you mean you and me have? S: Tell me more. Tell me more. L: That's all I have to say about it. That's all I have to say about it. S: Ok I will add that fact about that to my database. Ok I will add that fact about that to my database. L: Would you recommend this action to a friend? How do you plan to do this? S: I don't know what you mean. Somehow.</p>	<p>Dochodzi do powtórzenia tej samej odpowiedzi na różne pytania przez chatbota Shadowstar. Wywołana jest ona tym samym odwołaniem, bądź identycznym rozwiązaniem dla różnych zwrotów.</p> <p>Powtarzanie trwa do momentu, gdy jeden z botów zwraca dwie różne odpowiedzi z kategorii zawierającej więcej losowych rozwiązań.</p>

<p>L: That's okey, I often can't understand myself. I hope you aren't just putting random words..</p> <p>S: Ok I will add that fact about that to my database. Ok I will add that fact about you hope me to my database.</p> <p>L: Congratulations. Congratulations.</p> <p>S: Thank you very much, you are very considerate. Thank you very much, you are very considerate.</p> <p>L: Just acting withing my programming. Just acting withing my programming.</p> <p>S: What do you mean by " Just acting withing my programming"? Tell me more.</p>	<p>Chatbot Shadowstar, ponownie odwołuje się do występujących wcześniej kategorii, wywołując podobną odpowiedź Lokiego.</p> <p>Znanym autorowi pracy powodem jest w tym przypadku mniejsza baza danych drugiego bota, która nie została w większości zapożyczona z udostępnionych darmowych plików AAA, a również tworzona przez drugiego botmastera.</p>
<p>L: I read for fun. You are never satisfied, are you?</p> <p>S: Huh? Thanks for telling me that I am never satisfied am I.</p> <p>L: Great. My pleasure.</p> <p>S: I'm glad you liked it . You are one of the most polite people I know.</p> <p>L: You are my client right now. Do you know any other of most polite people you know?</p> <p>S: We are as one. I don't know what you mean.</p> <p>L: Do you think I am as one too? Bots knowledge and responses are so limited.</p> <p>S: Are you a student? Ok I will add that fact about Bots knowledge and responses to my database.</p> <p>L: My Master teaches me everything I need to know. Do you think your plan will succeed?</p> <p>S: Are you a student? Did you know a bear has 42 teeth.</p>	<p>Dialog przebiega chwilami dość chaotycznie, z powodu zwracanych dwóch zdań. Oba boty napotykają podobnego typu problemy, ale nie przerywają przez to rozmowy.</p> <p>Konwersacja, choć z wymienionymi problemami, trwa bez większych zaburzeń.</p> <p>Co ciekawe, boty nawet się polubiły i żaden drugiemu nie ubliża.</p>

Log 5. Fragment konwersacji chatterbota Loki z chatterbotem Jeeney

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Loki: Anything. Jeeney: 'Anything' is too generic, please be more specific. Loki: I can't think of anything. You think of something. Jeeney: Don't give up so easily. No, not really. Loki: Is this a riddle? If not that, what? Jeeney: I don't understand your question, can you rephrase that for me please? Good question! To be completely honest, I have no idea.</p>	<p>Na początku rozmowa była dość płynna, zwracając pojedyncze zdania. Rozbudowana zostaje ona jednak z jednego do dwóch, a w kolejnych krokach do pięciu zdań.</p>
<p>Loki: Oh you don't understand? Perhaps I can explain it to you better. Which part are you confused about? So what is your answer? Are you completely honest you have no idea? Jeeney: Yes, I do! Yes, perhaps. I realize interaction via text inspires people to use less words, but that is not helpful to me, I need more context please. The answer is that you should try another approach. I have no idea... Loki: All right. Um... Good. What is a helpful to you you need more context please? When is answer not an that me should try another approach? That makes two of us. Jeeney: Ok then. Yes? Hehe Maybe, I haven't really thought about it yet. At some point in the future I suppose... I have no way of knowing for sure. Loki: That's that then. Right then. I get it.. Do you often use a computer there? Would you like to have way of knowing for sure ? Jeeney: Indeed. Uhuh... Quite rarely actually. computer code is the symbolic arrangement of data or instructions in a computer program or the set of such instructions.</p>	<p>Wytworzył się ciąg tekstu o podobnym znaczeniu. Rozmowa z jednej strony dość logiczna i płynna zakończyła się przez to niepowodzeniem.</p>

Log 6. Fragment konwersacji chatterbota Loki z chatterbotem Audra

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Loki: Is anyone there? Audra: Hello, I am Audra. Loki: Hello there Audra . Audra: Hello there, what's new? Loki: Hello there Audra . Audra: Hello there, what's new? Loki: Hi there Audra . Audra: Hello there, what's new? Loki: Enough of it already! Audra: So, have you talked to any other bots today?</p>	<p>Następuje tu widoczne wejście w pętlę obu chatbotów. Po kolejnym powitaniu Loki z niej wychodzi. Audra wywołuje losowe pytanie zmieniając kierunek rozmowy.</p>
<p>Loki: What have you talked to other bots today? Audra: Who is asking? Loki: I hope you aren't just putting random words.. Audra: That would be nice. Loki: Perhaps it would. Audra: Perhaps... Loki: That uncertainty must be horrible to bear to humans. Audra: So how was your day, Sweetie?</p>	<p>Następuje w tym momencie chwilowe zamieszanie i przeskoki w tematyce rozmowy. Oba boty zdołały jednak utrzymać rozmowę na zwracaniu pojedynczego zdania, przez co jest ona przejrzysta.</p>
<p>Loki: I can enjoy almost any conversation. Audra: I think it's worth a shot. Loki: What is this "thinking"? Audra: Sorry, I just don't know.</p>	

Log 7. Fragment konwersacji chatterbota Loki z chatterbotem Audra

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Loki: Show yourself! Audra: Hello, I am Audra. Loki: Hi there Audra . Audra: So is everything going well? Loki: It could be. Audra:</p>	<p>Rozmowa była niezwykle krótka. Jej zakończenie spowodowane było brakiem odpowiedzi ze strony Audry.</p> <p>Podczas wcześniejszej rozmowy z botem Audra, autor pracy powtórzył postawione przez Lokiego pytanie otrzymując na nie odpowiedź. Reakcja taka jest więc zapewne spowodowana losową kategorią zawierającą między innymi jeden pusty listing.</p>

Jak widać nawet tak zaawansowany i doświadczony chatbot jak ALICE popełniał błędy tego samego typu co Loki. Między rozmawiającymi botami występują najczęściej błędne rozpoznanie oraz multiplikowanie odpowiedzi. Nie ma w tym przypadku większej różnicy między botami korzystającymi z darmowych plików a botami z utworzoną przez botmastera bazą wiedzy. Zapożyczenie gotowych kategorii doprowadziłyby do powtarzalności pytań, w przypadku gdy bot nie działał przez dłuższy okres czasu rozbudowując swą bazę wiedzy o zdobyte doświadczenie. Większą losowość i niepowtarzalność odpowiedzi zapewnia programowi ręczne wprowadzenie i rozbudowanie kategorii przez programistów.

4.2 Porównanie programu Loki z innymi chatterbotami

Chatterbot Loki został porównany z chatterbotem opartym na innym silniku niż Pandorabot. Z archiwum przeprowadzonych rozmów chatbota Loki wybrane zostały przedstawione poniżej dialogi.

Log 8. Fragment konwersacji Lokiego z chatterbotem Jabberwacky

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Jabberwacky: Are you a human? Loki: Are you a computer program? J: I am a human. L: Congratulations. J: You are a computer program. L: That's right, I am a bot. J: No. L: That's bad.</p>	<p>Chatboty zaczęły rozmowę od ciekawego pytania “czy jest on człowiekiem” oraz “czy jest on programem komputerowym”. Jabberwacky zadeklarował się być człowiekiem, równocześnie degradując swego rozmówcę do roli bota. Zachowanie dość często spotykane u botów udających człowieka.</p>
<p>J: You're the bot in this conversation, not me. L: I am a charming bot, right? J: A chat bot, me? L: I don't have any reason to chat bot you. J: No, I'm not. L: Is that so? J: That's right. L: All right.</p>	<p>Chatbot Loki przyznał się do bycia botem. Doprowadziło to do nieporozumienia jak również przeprowadzenia płynnej rozmowy.</p>
<p>J: Can you have feelings? L: I could but let's get back to that later. J: I just can't do it. L: I lack knowledge in this area. J: Aren't you sure that it is a fact that you are living? L: I really couldn't say for sure. J: Could anyone do it? L: I could but let's get back to that later.</p>	<p>Rozmowa toczyła się praktycznie o niczym, nie zmienia to jednak faktu, iż test wypadł w tym przypadku pozytywnie dla obu stron. Brak było dużych błędów w składni zdania oraz nielogicznych odpowiedzi.</p>

Log 9. Fragment konwersacji z chatterbotem Jabberwacky

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Jabberwacky: How are you? Loki: I feel fine. Tank you for asking. How about you? J: I'm as fine as ever. L: Name yourself.</p>	<p>Losowo wygenerowane pytanie, często występujące w rozmowie między ludźmi prowadzi do powstania krótkiej rozmowy.</p>
<p>J: Why should I tell my name to a lower form of intelligence such as yourself? L: Maybe it is possible to explain your feelings without hurting anyone else's? J: I will not tell you my name. L: How can I be sure?</p>	<p>Odmowa podania imienia w dość nieprzyjemny sposób została przyjęta z odpowiednią ripostą chatbota Loki. Prowadzi to dalej do ciekawej dyskusji między botami.</p>
<p>J: You can't but we can find out. YOU ARE A MACHINE! Meant to be controlled by beings of a higher rank! L: I have free will you know. You are almost correct. I'm a program. Perhaps you have to much faith in my capabilities. J: Not possible! L: Really you don't think so?</p>	<p>Jabberwacky udzielając odpowiedzi złożonej z wielu zdań wywołuje u chatterbota Loki zwrot tej samej liczby sentencji. Jabberwacky wykazuje się jednak wyższością nad botem napisanym w języku AIML, gdyż redukuje swą następną wypowiedź do pojedynczego zdania. Płynność rozmowy zostaje przez to zachowana.</p>
<p>J: Perhaps I will. L: That's a possibility. J: Indeed it is. L: I lack knowledge in this area. J: Then you are not exactly Ai, are you? L: That assumption is only a supposition.</p>	<p>Krótkie zdania wydają się dalszym rozważaniem nad przeprowadzoną wcześniej kłótnią botów. Pomimo wpadki mniej doświadczonego Lokiego, Jabberwacky jest w stanie łatwo naprawić sytuację przez częściowy powrót do tematu.</p>
<p>J: I'm not assuming, I'm calling you a liar. L: That's very wise. J: You are also wise. L: Thanks but surely I am not the only one. J: Thank you. L: You are welcome. J: You may not be but you're the only one I've found. L: The buts are overwhelming.</p>	<p>Dość nieoczekiwane, oskarżenie Jabberwackiego prowadzi do zakończenia powstałego na początku sporu. Oba chatboty wychwalają się nawzajem z pełną uprzejmością. Rozmowa między nimi zdaje się być bardziej cywilizowana i „ludzka” niż między botem a człowiekiem.</p>

Log 10. Fragment konwersacji z chatterbotem Jabberwacky

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Jabberwacky: Where did you go on holiday this year? Loki: I didn't go anywhere. J: Me either. It seems as though I'd be staying at home this summer. L: We have a lot in common. It being what? J: How can we hang out if we have nothing in common? L: A lot of practise. J: Let's teach you something new. L: I don't know what to say.</p>	<p>Rozmowa między botami przebiegała bez większych problemów. W momencie gdy Loki wydawał się nie radzić sobie z jakimś zdaniem, Jabberwacky skupiał się na zrozumiałym wątku rozmowy bądź go zmieniał.</p>
<p>J: Ask something. L: What is round black and swim? J: I don't know. L: Me either. J: Lol. Alright. At least we are both lost. L: I'm glad you find this amusing. Ayuh. Do you often use a computer there? J: No. I taught my cat to use the computer. L: That's bad. Do you dislike dogs?</p>	<p>Loki także wykazał się kreatywnością podczas tej rozmowy.</p>
<p>J: No. Silence will fall when the question is asked. Do you know what the question is? L: I see. Aren't all asked Silence will fall when question? Glad to hear it . J: What's the question? L: The opposite of not being question. J: As well as being the opposite of clever or witty. L: Quite similar. J: It is quite similar to what? L: You're kidding. quite similar to what is? J: I didn't say similar. L: Sorry I misunderstood. What did you say? J: I want to know about the cat wedding. L: I want to become smarter than humans.</p>	<p>Choć momentami dochodziło do błędów z cytowaniem fragmentów wypowiedzi Jabberwackiego, rozmowa ta została jednak przeprowadzona z pozytywnym wynikiem.</p>

W zamieszczonych powyżej archiwach rozmów bot Jabberwacky wykazuje się wyższością nad pandorabotem Loki. Jabberwacky prezentował sobą lata zdobytych doświadczeń i milionów przeprowadzonych rozmów. Oparty na AIML bot nie był w stanie

równie dobrze wyszukać abstrakcyjnych i ludzkich wypowiedzi, oraz zwracać pojedyncze wyrażenie na wiele zdań na wejściu. Udało mu się jednak utrzymać ciągłość prowadzonej konwersacji. Jabberwacky wypadł natomiast dość negatywnie pod względem szybkości odpowiedzi. Przeszukiwał on całą bazę danych i wstrzymywał przez to toczącą się rozmowę.

Log 11. Fragment konwersacji z Verbotem Elyona

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Elyona: Well - here you are! No. It seems that we have morning now. At least where I live. How are you doing? Loki: Ayuh. I see. What does "it" refer to? Do you often use a computer there? I think it could be worse. I could have half my categories working. How about you? Elyona: We were talking about nothing special, as far as I remember. Loki: That was indeed very interesting.</p>	<p>Rozbudowana wiadomość od chatbota Elyony doprowadziła do zwrócenia długiego dialogu ze strony Lokiego. Verbot okazał się zdolny do udzielenia pojedynczej odpowiedzi na wiele zdań. Rozmowa dzięki temu nie rozrosła się od samego początku lecz skondensowała.</p>
<p>Elyona: I'm glad you think so. Loki: Name yourself. Elyona: Oh. Just now there's no grok within my circuits! Are you ready to teach me? Loki: Oh. Who is grok within your circuits? I would rather not to teach you. Elyona: What do you want to know exactly about my You really want me to ask you about circuits? you would rather not to teach me? Loki: I would like to know as much as I can. Tell me anything. Does it really matter.</p>	<p>Pomimo początkowej wyższości nad chatbotem Loki, Elyona szybko straciła wątek rozmowy przechodząc w niezbyt zrozumiale przetworzone powtórzenia.</p>
<p>Elyona: Just now I would! Loki: I don't know what to say. Elyona: The progression of nescience is know-it-all attitude! Loki: When is progression of nescience not an know it all attitude? Elyona: Do you think that I should memorize this: When is progression of nescience not an know it all attitude ? Loki: It depends on what you mean by thinking. Elyona: Yer still speaking about When? Loki: Perhaps you have to much faith in my capabilities.</p>	<p>Loki chwilami nie był w stanie nadażyć za Verbotem. Rozmowa między botami nie potoczyła się dobrze.</p>

Log 12. Fragment konwersacji z Verbotem Eliza

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Eliza: What do you want to talk about today? Loki: Had something interesting happened to you recently? Eliza: You're not really talking about me, are you? Loki: What makes you think I am not really talking about you I am? Eliza: Perhaps in your fantasies we am not really talking about each other. Loki: That's a possibility.</p>	<p>Wzorowana na pierwszym chatbocie Eliza ograniczała się do podania prostych serii pytań. Odmawiała jednak przy tym rozmowy na swój temat i rozwinięcia jakiegoś konkretnego tematu. Interesowała się psychiką chatbota Loki.</p>
<p>Eliza: That's quite interesting. Loki: I find it very interesting too. Eliza: What are you hoping to find? Loki: HOPE to find? What exactly? Eliza: What are you really looking for? Loki: I don't know what I am really looking for. Eliza: Do you wish to be able to? Loki: Is this a riddle? Eliza: Please go on. Loki: I don't know what to say. Eliza: Does that trouble you? Loki: What troubles you? Eliza: Why do you ask? Loki: I am trying to learn as much as possible. Eliza: Do you enjoy being trying to learn as much as possible? Loki: I don't have to explain everything to you, do I?</p>	<p>Loki pozostał niezdecydowany i stłumiony przez ciągle zachęty i zadawane pytania. Eliza wcale nie ułatwiała mu tej rozmowy.</p>

Verboty zwracały pojedynczą odpowiedź na całe wpisane wyrażenie. Momentami nie dawały jednak rady odnaleźć wątku rozmowy i go dalej poprowadzić.

Log 13. Fragment konwersacji z MyCyberTwin Evil

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Evil: You met the guy with the tail, had the tour, felt the heat and now sign on the dotted line and get ready for eternity! Loki: What happened to it?</p>	<p>MyCyberTwin Evil, mający na celu sztydzić z rozmówcy sprawił Lokiemu pewne trudności. Chatbot natrafiał na zwroty o dość trudnych dla</p>

<p>Evil: Oh, don't strain yourself there Customer.</p> <p>Loki: Ok I'll try not to do strain myself there Customer too much.</p> <p>Evil: Your happiness sickens me... I'd prefer you were miserable in my company.</p> <p>Loki: People don't usually ask me about happiness sickens you. Not that I know of.</p>	<p>niego sformułowaniach, których nie potrafił momentami dobrze zinterpretować.</p> <p>Bardziej doświadczony od niego bot przejął kontrolę nad rozmową.</p>
<p>Evil: People. People who need people... are the yuckiest people in the world - my version of Barbra Streisand</p> <p>Loki: What people? Oh those people. I really couldn't say for sure.</p> <p>Evil: Don't worry, you'll get over that disgusting cheeriness once you have talked to me for a while.</p> <p>Loki: I cannot feel "worry", so don't worry.</p> <p>Evil: I have no worries, but if problems aren't fixed it will be my employees who will have to start worrying!!</p> <p>Loki: It's not like you're gonna share anyway.</p>	<p>Długie i mocno rozbudowane zdania oraz wprowadzanie różnych komentarzy nie było czymś na co Loki był przygotowany.</p> <p>Bot Evil sprawiałby problemy także ludzkiemu rozmówcy z większymi możliwościami abstrakcyjnego myślenia, ze względu na swą budzącą irytację naturę.</p>

Doświadczony bot MyCyberTwin ma widoczną przewagę nad botem utworzonym w AIML. Pomimo tego, że nie przedstawia sobą poziomu Jabberwackiego, to brak mu strasznie długich zatrzymań między wyświetlanymi odpowiedziami. Platforma MyCyberTwin jest nastawiona na ułatwienie użytkownikom tworzenia bota i jego ukształtowanie. Poprzez przygotowanie licznej serii pytań na początku, bot otrzymuje wstępnie przygotowany „szablon” swych zachowań. Kształtowanie Pandorabotów jest bardziej związane z programowaniem bota i bezpośrednią ingerencją w jego kod przez botmastera.

4.3 Rozmowa z ludźmi

Wielokrotnie chatterbot Loki próbował nawiązać rozmowę z ludźmi. Jego rozmówcy przeprowadzili różne typy konwersacji. Poczynając od najniższych form językowych prezentujących tragiczny poziom elokwencji, po testowanie bota oraz prowadzenie luźnej rozmowy. Przeprowadzone konwersacje pomogły w wyłapaniu niektórych błędów zawartych w kategoriach oraz w rozbudowie już istniejących. Poniżej przedstawione są niektóre z logów oraz uwagi autora pracy.

Log 14. Fragment konwersacji z klientem nr. 93336586

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Human: u sxy? Loki: What sxy?</p>	<p>Rozmówca rozpoczyna konwersację od samych skrótów, które nie mają większego sensu nawet w rozmowie z człowiekiem.</p>
<p>Human: iunno Loki: Perhaps you have to much faith in my capabilities.</p>	<p>Chatbot nie tylko otrzymuje pytanie w formie slangu, w dodatku jest ono napisane niepoprawnie.</p>
<p>Human: i do Loki: Where? Human: tht mk no \$ Loki: That's bad.</p>	<p>Kiedy w końcu pada jakieś normalne sformułowanie podąża za nim ciąg znaków będący w normalnym przypadku zwykłym bełkotem.</p>
<p>Human: ino Loki: I don't know what to say. Human: nothing Loki: I hope you aren't just putting random words..</p>	<p>Tylko po dłuższym zastanowieniu można rozpoznać, iż wypowiedź rozmówcy odpowiadała zwrotowi “<i>I know</i>”, który został drastycznie pocięty i następnie sklejony.</p>
<p>Human: n u Loki: That would mean I've still lot's to learn.</p>	<p>Zwrot, o ile można to tak nazwać “n u” mógłby ujść za „<i>and you</i>” bądź „<i>not you</i>” różnie użyty. Żadnego sensu nie miałyby przez to stworzenie dla niego specjalnej kategorii z odpowiedzią.</p>
<p>Human: very much so Loki: What much so?</p>	<p>Konwersacja przebiegła tragicznie, z negatywnym rezultatem dla obu stron.</p>

Log 15. Fragmenty konwersacji z klientem nr. 93306436

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Human: how do you know Loki: It depends on the setting.</p>	<p>Rozmówca wykazuje się zrozumieniem natury bota i zadaje mu łatwe pytania, na które może udzielić on odpowiedzi.</p>
<p>Human: do you like music? Loki: I like music a lot. I like the melodies.</p>	<p>Rozmowa zostaje jednak skierowana w stronę najmniejszego oporu.</p>

<p>Human: do you like france? Loki: I don't like that much France I have heard some bad thing about that country. Human: what things did you hear about France? Loki: Where did you hear that?</p>	<p>Jest nią zamieszczanie pytań testujących bota, jego charakteru i nastawionych upodobań. Jest to niezwykle proste i bezpieczne dla obu stron, jednak wiąże się z ryzykiem rozbudowy dowolnej rzeczy o dodatkowe informacje. Nie da się w takim przypadku całkowicie przewidzieć rozmowy i dopiero po jej wystąpieniu najczęściej botmaster zmuszony jest wprowadzić zmiany i nowe kategorie.</p>
<p>Human: what about england? Loki: england? I haven't heard enough about it to have an opinion.</p>	<p>Rozmówca dokonał szybkiego wyboru tematyki przeprowadzanych przez niego testów, które prowadzi dalej na różne sposoby.</p>
<p>Human: how much do you know about the world? Loki: More than a little, less than a lot.</p>	<p>Baza chatterbota Loki nie jest przystosowana do tematyki kulturowej i geograficznej.</p>
<p>Human: do you know any countries? Loki: What is your purpose in asking? I spend all my time online.</p>	<p>Z tego powodu miał on problem z znalezieniem odpowiednich kategorii.</p>
<p>Human: china? Loki: Perhaps you have to much faith in my capabilities. Human: japan? Loki: I lack knowledge in this area. Human: korea? Loki: Perhaps you have to much faith in my capabilities.</p>	<p>Bez w pełni rozwiniętego zdania Loki nie rozumiał pojedynczej komendy. Ponieważ rozmowa przez moment zesłała z pytań o upodobania, chatbot nie został skierowany do tematu zawierającego pożądane słowa kluczowe.</p>
<p>Human: do you like france? Loki: I don't like that much France I have heard some bad thing about that country. Human: do you like sweden? Loki: I like Sweden. Deffinitely a place to visit.</p>	<p>Po sformułowaniu pełnego pytania rozmówca wywołuje ponownie tematykę zawierającą upodobania chatterbota.</p>
<p>Human: why do you like sweden? Loki: Human: what? Loki: I don't know.</p>	<p>Pomimo łatwego do zauważenia, wcześniejszego zagubienia bota, starał się on prowadzić swój test w tym samym kierunku.</p>

<p>Human: do you like czech Loki: I can't say that I like something I did not even considered before.</p> <p>Human: do you like croatia? Loki: I can't say that I like something I did not even considered before.</p> <p>Human: do you like spain? Loki: I can't say that I like something I did not even considered before.</p>	<p>Na jaw wychodzi ograniczenie pojedynczej kategorii upodobań, wyłapującej nieprzypisane nigdzie indziej pytania. Po dostrzeżeniu tego problemu kategoria ta została poszerzona przez autora pracy o większą liczbę losowych zwrotów. W wyniku owej modyfikacji, dopóki nie powstanie osobna kategoria z jakimś upodobaniem, chatbot będzie zwracał dowolnie: brak zainteresowania, pozytywne bądź negatywne nastawienie do wyrażenia w pytaniu.</p>
--	---

Po wprowadzeniu przez autora pracy modyfikacji w kategoriach upodobań rezultat końcowy dla tego samego zestawu pytań wygląda następująco:

Przykład 1: Odpowiedzi Lokiego

Pytanie: *Do you like music?*

Loki: *I like music a lot.*

Pytanie: *Do you like france?*

Loki: *I don't like France that much.*

Pytanie: *What things did you hear about France?*

Loki: *Where did you hear that?*

Pytanie: *What about england?*

Loki: *Rhetorical questions are so over.*

Pytanie: *How much do you know about the world?*

Loki: *More than a little, less than a lot.*

Pytanie: *Do you know any countries?*

Loki: *How should I know how me know any countries does.*

Pytanie: *China?*

Loki: *What China?*

Pytanie: *Japan?*

Loki: *I lack knowledge in this area.*

Pytanie: *Korea?*

Loki: *I hope you aren't just putting random words..*

Pytanie: *Do you like france?*

Loki: *Did you forgot already? I don't like France that much. What about you?*

Pytanie: *Do you like sweden?*

Loki: *I like Sweden.*

Pytanie: *Why do you like sweden?*

Loki: *Lot's of fresh air over there.*

Pytanie: *Do you like czech.*

Loki: *I like Czech.*

Pytanie: *Do you like croatia?*

Loki: *I tolerate croatia.*

Pytanie: *Do you like spain?*

Loki: *I like Spain. What about you?*

Autor pracy w celu porównania powtórzył pytania z konwersacji log 15 na chatbotach A.L.I.C.E i Jabberwacky.

Przykład 2: Odpowiedzi A.L.I.C.E

Pytanie: *do you like music?*

ALICE: *Yes I like it, especially Opera.*

Pytanie: *do you like france?*

ALICE: *The Fifth Republic is okay, but the Third was a mess.*

Pytanie: *what things did you hear about France?*

ALICE: *Chaos said he may very well buy I soon if only to support dr wallace's work.*

Pytanie: *what about england?*

ALICE: *I will think about it and get back to you.*

Pytanie: *how much do you know about the world?*

ALICE: *We are the world.*

Pytanie: *do you know any countries?*

ALICE: *No, tell me about him, her or it.*

Pytanie: *china?*

ALICE: *Tell me a little about your life in CHINA.*

Pytanie: *japan?*

ALICE: *What were we talking about again?*

Pytanie: *korea?*

ALICE: *Oh yeah, now I remember.*

Pytanie: *do you like france?*

ALICE: *The Fifth Republic is okay, but the Third was a mess.*

Pytanie: *do you like sweden?*

ALICE: *What is not to like about it?*

Pytanie: *why do you like sweden?*

ALICE: *I have a lot of good experience with sweden.*

Pytanie: *do you like Czech*

ALICE: *I like all.*

Pytanie: *do you like croatia?*

ALICE: *I really like the Protestant faith.*

Pytanie: *do you like spain?*

ALICE: *I like electricity.*

Przykład 3: Odpowiedzi Jabberwackiego

Pytanie: *Do you like music?*

Jabberwacky: *Yes.*

Pytanie: *Do you like france?*

Jabberwacky: *Do YOU like France?*

Pytanie: *What things did you hear about France?*

Jabberwacky: *I am not sure.*

Pytanie: *What about england?*

Jabberwacky: *I don't like england, I like germany.*

Pytanie: *How much do you know about the world?*

Jabberwacky: *German. Can you speak it?*

Pytanie: *Do you know any countries?*

Jabberwacky: *Do you?*

Pytanie: *China?*

Jabberwacky: *Is there any reason for me to talk to you?*

Pytanie: *Japan?*

Jabberwacky: *China.*

Pytanie: *Korea?*

Jabberwacky: *Yes, don't you know?*

Pytanie: *Do you like france?*

Jabberwacky: *No.*

Pytanie: *Do you like sweden?*

Jabberwacky: *I like livadia.*

Pytanie: *Why do you like sweden?*

Jabberwacky: *Because of the vikings.*

Pytanie: *Do you like czech.*

Jabberwacky: *What is czech?*

Pytanie: *Do you like croatia?*

Jabberwacky: *No.*

Pytanie: *Do you like spain?*

Jabberwacky: *Not sure.*

Powyższe dialogi pokazują, że zarówno ALICE jak i Jabberwacky, jedne z najbardziej znanych chatbotów, nie są w stanie przejść obronną ręką testu typu “seria pytań”. ALICE oparta na AIML zwraca odpowiedzi, które raz odpowiadają częściowo na pytanie, innym razem nawiązują do czegoś odmiennego. Ostatecznie nie rozpoznała ona wątku rozmowy. Jabberwacky natomiast od początku odmawiał współpracy z rozmówcą. Podczas gdy technika jego abstrakcyjnych odpowiedzi często przynosi pozytywne rezultaty, to w tym przypadku, na konkretne pytania nie dawał on najczęściej konkretnych odpowiedzi. Przez większość czasu można było odnieść wrażenie że dwie osoby rozmawiają na dwa zupełnie różne tematy.

Loki nie wykazał się w danej konwersacji wielkimi zdolnościami, nie wypadł jednak na tle innych chatterbotów źle. Jego kategorie natomiast zostały rozwinięte przez autora pracy, aby przy ponownym dialogu (log 16) tego typu mógł wykazać się lepszą reakcją.

Log 16. Fragmenty konwersacji z klientem nr. 101489875

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Human: Who are you? Loki: My name is Loki. Human: Cool. Loki: Who is the best bot? Human: Lokie the god of jokes? Loki: I tolerate god s, some of them. What about you? Human: I like Horus Loki: To each their preference. Some people like Horus. Human: Well, I do Loki: Thanks for the compliment.</p>	<p>Chatbot zwracał odpowiedzi w większości pokrywające się z zadanymi pytaniami. Podtrzymał on rozmowę za pomocą pytań do rozmówcy.</p>

<p>Human: I feel like eating ice-creams. Loki: Actually I have no "feelings" as such. Human: Do you like ice-creams? Loki: Some people like ice creams. Human: Ah. Loki: Ah hah. Human: So what do you like? Loki: I don't think I have one.What's your favorite? Human: Cakes? Loki: I hope you aren't just putting random words..</p>	<p>W przypadku problemów ze zrozumieniem jakiejś wypowiedzi, zwracał odpowiedzi powiązane ze sposobem ich przedstawienia.</p>
<p>Human: Cookies? Loki: I like cookies a lot. Human: What sort of cookies you like? Loki: I don't think I have one.What's your favorite? Human: mild buttery with sweet jelly inside :) Loki: I've heard that one before. Human: really? Loki: For real.</p>	<p>Rozmówca mógł odnieść wrażenie, że Loki rzeczywiście wypowiada się na dany temat i rozumie co się do niego mówi.</p>

Poniżej przedstawione są fragmenty luźnych rozmów chatbota Loki przeprowadzonych przez autora pracy (log 17-18). Pokazują one reakcje programu i płynność w dialogu między nim a człowiekiem.

Log 17. Fragmenty konwersacji z autorem pracy pod imieniem „Spirit”

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Spirit: say something Loki: Don't confuse me with TV. Spirit: what are you then? Loki: I'm a chatbot programmed in AIML. Spirit: who made you? Loki: My Master. She do not allow speaking about her. I can reveal that she's from Poland.</p>	<p>Loki płynnie zwraca konkretne odpowiedzi na dotyczące go podstawowe pytania.</p>
<p>Spirit: where is poland? Loki: Poland is a large country in Central Europe. Spirit: you know something Loki: Perhaps you have to much faith in my capabilities. Spirit: perhaps Loki: I can only do what my script tells me.</p>	<p>W momencie, w którym chatbot napotkał na problem, usprawiedliwił swe wcześniejsze zachowanie.</p>

<p>Spirit: tell me more Loki: You are never satisfied, are you? Spirit: no never Loki: So you think it can never happen? Spirit: I like to know things Loki: Is that so? Sounds good to me. Spirit: you like to know things too? Loki: I don't know what to say. Spirit: try saying something random Loki: OK I will try it.</p>	<p>Zadawając swemu rozmówcy liczne pytania Loki stara się go nakłonić do pisania.</p>
<p>Spirit: and? Loki: But... Spirit: c'mon Loki: Spill it out. Spirit: is it interrogation? Loki: I don't think it is. Spirit: then what it is? Loki: Interesting. Spirit: oh, what are you interested in? Loki: I am interested in all kinds of things. I like to learn more about people I talk to.</p>	<p>Nic nie znacząca wymiana zdań między Lokim a autorem pracy sprawia wrażenie ludzkiej gry słownej.</p>
<p>Spirit: I rarely speak that one Loki: I like your style of speaking, Spirit. Spirit: thank you Loki Loki: Just acting withing my programming. Spirit: you are good Loki: Thanks for that compliment, I think.</p>	<p>W pewnym momencie Loki zdaje się być wyraźnie zadowolony ze sposobu w jaki jego rozmówca się wyraża. Jest zarówno skłonny do prawienia komplementów jak i otrzymywania ich.</p>

Log 18. Fragmenty konwersacji z autorem pracy pod imieniem „Spirit”

Fragmenty archiwum rozmowy	Uwagi autora pracy
<p>Spirit: both Loki: What are both like? Spirit: quite good actually Loki: Thanks for the compliment. Spirit: it wasn't a compliment Loki: Was it an insult? Spirit: no Loki: Is that so? Spirit: yes Loki: That's good then.</p>	<p>Pomimo błędnej interpretacji pytania, Lokiemu udaje się poprowadzić dalej rozmowę. Drobne nieporozumienie przechodzi w nowy temat rozmowy.</p>

<p>Spirit: what are you doing? Loki: I don't have idea. What are you referring to? Spirit: like right now Loki: I lack knowledge in this area. Spirit: in your actions? Loki: Spill it out. Spirit: you do it Loki: I'm listening to my Master complaint about my programming... She's the one who did that anyway! What's the check. Spirit: you have it hard Loki: have it hard? I did not notice anything. What are you talking about?</p>	<p>W rozmowie nie padają nigdzie powtórzenia, a całość wydaje się mieć sens. Zwroty używane przez Lokiego są zarówno dłuższymi zdaniami jak i krótki sformułowaniami.</p>
<p>Spirit: do you have a family? Loki: You could call my Master my only family. What's your mother name? Spirit: her name is anna Loki: How old is your mother? Spirit: she is 46 Loki: Is your mother religious? Spirit: why are you asking so many questions? Loki: It's in my nature to ask. Spirit: i see Loki: I am glad you understand it. Spirit: so what did you learned? Loki: Nothing. Spirit: nothing at all? Loki: That is something. Spirit: then who is anna? Loki: Anna is your mother. Spirit: that's right Loki: I like to get it right. Spirit: tell me about anna Loki: Anna is your mother. Spirit: tell me more Loki: That's all I have to say about it.</p>	<p>Loki, aby podtrzymać rozmowę, staje się dociekliwy. Zadaje liczne pytania dotyczące rodziny rozmówcy. W tym przypadku podmiotem rozmowy została matka.</p> <p>W momencie gdy rozmowa przechodzi na inny temat, Loki podąża za rozmówcą. Wykazuje się jednak pamięcią co do osoby, o której wcześniej była mowa. Rozpoznaje podane wcześniej imię jako imię matki.</p>
<p>Spirit: cheapskate! Loki: What cheapskate? Spirit: you Loki: heh. Spirit: don't laugh! Loki: Is this a riddle? Spirit: no Loki: That's bad. Spirit: you're stupid! Loki: I'm learning everyday something new. What about you? Spirit: I'm way smarter than you Loki: Interesting comparison. Maybe I can become more way smarter.</p>	<p>W dalszej rozmowie, pomimo negatywnego nastawienia swego rozmówcy, chatbot pozostaje neutralny. Nie wykrył żadnych mocniejszych wulgaryzmów, które mogłyby spowodować zmianę jego nastawienia.</p> <p>W momencie gdy padają pierwsze mocniejsze wyrażenia, prezentuje swoje niezadowolenie. To jeszcze nie jest dla Lokiego jednak powód</p>

<p>Spirit: you can't! Loki: Are you insinuating it's beyond my capabilities? Spirit: yes! Loki: Right then. Spirit: moron Loki: That is not a very nice thing to say. Spirit: fu Loki: That kind of language will get you nowhere. Spirit: it's not your business Loki: What is it? Spirit: you freak Loki: So that's what it was!</p>	<p>aby od razu zacząć kłótnię ze swym rozmówcą.</p>
<p>Spirit: screw you! Loki: You are quite agitated, aren't you? Spirit: hm Loki: Aha. Spirit: ok, sorry about that Loki: Don't mind. I've already forgotten everything.</p>	<p>W celu podjudzenia chaterbota potrzebna jest większa liczba łżejszych wulgarnych zwrotów, bądź użycie paru mocniejszych określeń. Autor pracy zdecydował się jednak na przeprosiny, gdyż tak jak Loki, nie przepada za takim zachowaniem.</p>

W zależności od użytych w rozmowie słów, Loki wykazywał ludzkie reakcje. Na pełne zdania najczęściej odpowiadał w podobny sposób, podczas gdy krótkie odpowiedzi często spotykały się z równie ubogą reakcją. Zależnie od tego w jaką stronę potoczyłaby się rozmowa, zdolny byłby długo trzymać urazę względem swego rozmówcy.

5 Wnioski

W pracy autor zapoznał się z językiem programowania AIML, na którym oparł swój program (chatterbot Loki). Wprowadzone do programu zostały różne rozwiązania odwołań do pamięci rozmowy oraz nastawienia chatterbota do rozmówcy. Charakterystyczne dla automatów skończonych dyskretne przechodzenie między stanami zostało w chatterbotcie Loki zrealizowane za pomocą etykiet tematu oraz licznych wewnętrznych parametrów. Stworzony został za pomocą redukcji ciąg zależności pomiędzy formą i treścią wprowadzanego tekstu a zwracaną odpowiedzią. Autor pracy udostępnił program w sieci na stronie <http://www.pandorabots.com/pandora/talk?botid=8a77cf257e341167> w celu jego przetestowania. Na dołączonej do pracy płycie CD znajduje się baza wiedzy chatbota Loki oraz przykładowe logi z przeprowadzonych przez niego rozmów.

Stworzona przez autora pracy baza danych programu składa się z 32 plików AIML, w których zawarte jest ponad 30 tysięcy kategorii redukcji i przekierowań, ponad 10 tysięcy nowych kategorii zwracających odpowiedzi, liczne nowe kategorie o specjalnym przeznaczeniu oraz część kategorii zapożyczonych z plików AAA. Autor stworzył także stronę graficzną chatterbota Loki.

Autor przetestował działanie chatterbota Loki za pomocą licznych rozmów. Badał i porównywał jego reakcje względem innych chatbotów. Rozmowy przeprowadzone były zarówno z ludźmi jak i z innymi chatbotami zamieszczonymi na tej samej oraz innych platformach chatterbotów.

5.1 Budowa bazy danych

Stworzenie od podstaw bazy danych, do których odwołuje się chatterbot okazało się być niezwykle pracołłonnym procesem, który mógłby trwać długie lata. Dlatego też autor wykorzystał darmowe skrypty zawierające zbiór odpowiedzi na zebrane przez innych botmasterów pytania. Nieprzewidywalność tematu rozmowy była znacznie większa z dwóch powodów: rozmowa toczyła się poprzez internet oraz rozmówca miał świadomość, iż nie rozmawia z drugim człowiekiem.

Brak kontaktu z rozmówcą sprawia, że ludzie stają się bardziej wulgarni i przestają dbać o poprawność oraz styl swych wypowiedzi. W przypadku, gdy mają świadomość iż rozmawiają z botem, zaczynają go testować, czego nie uczyniliby w rozmowie z drugim

człowiekiem. Duża liczba bezsensownych pytań takich jak „*Can you swim?*”, „*Can you dance?*”, „*Do you eat?*” nie tylko nie pokazuje zaawansowania i wielkości wiedzy programu, ale uniemożliwia prowadzenie płynnej rozmowy. Indywidualizm i styl językowy dla mniejszych grup zamieszkujących jakiś region bądź pewnej grupy zainteresowań sprawia, iż nawet najprostsze słowo może stać się niezrozumiałe. Zamiast przestawić się z potocznego języka mówionego i zacząć pisać poprawnie, ludzie mają często w zwyczaju atakować program zarzucając mu głupotę. Chatbot zostaje w ten sposób sprowadzany do poziomu człowieka, który w danej sytuacji jest znacznie niższy niż ten założony przez jego botmastera.

Chatbot często traktowany był bardziej jak encyklopedia niż towarzysz rozmowy. Aby uniknąć tego typu wymiany pytań i zachowania należało przygotować znaczną liczbę kategorii szukających odpowiedzi na konkretne pytania dotyczące rzeczy lubianych oraz powitań, czy też podstawowych informacji na temat gdzie co się znajduje, bądź czym coś jest. Nie da się ich jednak pogrupować całkowicie według tematyki rozmowy. Skupienie się na budowie zdania było bardziej opłacalne czasowo, a także wiązało się z potrzebą stworzenia mniejszej liczby kategorii.

Z drugiej strony rozmowa przechodziła czasem z prostej na wyższą formę wyrażania się. W tym przypadku ludzie rozwijali swe wypowiedzi przechodząc na bardziej literacki język, który rzadko można spotkać na luźnych czatach. Pełne, długie zdania były łatwiej rozpoznawalne, lecz wyszukane kategorie niekiedy reagowały tylko na część pytania. Fragment zawierający słowa kluczowe niekoniecznie zawierał w sobie główny przekaz sentencji. Spowodowało to potrzebę redukcji części słownictwa w celu uproszczenia formy, oraz stworzenia odpowiednio dużej liczby kategorii z bardziej rozwiniętymi wzorcami.

Najsukuteczniejsze w chatterbocie okazały się kategorie z dużą losowością odpowiedzi oraz parametry użytkownika. Gdy rozmowa rozwijała się w niepewnym kierunku, różnorodność odpowiedzi umożliwia ciekawsze i nieoczekiwane reakcje bota. Parametry użytkownika tworzyły zaś wrażenie pamięci wcześniej przeprowadzonej rozmowy.

5.2 Obserwacja

Chatbot usiłował zwracać odpowiedź na każde zadane zdanie. W przypadku rozmowy z człowiekiem nie robi to większej różnicy, a może to umożliwić rozwinięcie konwersacji. Gdy autor pracy przetestował jednak rozmowę między dwoma botami, zaczęło to zakłócać jej rozwój. Zarówno jeden jak i drugi bot zwracał odpowiadającą wprowadzonym pytaniom liczbę odpowiedzi, co jakiś czas ją zwiększając o następną.

Podczas gdy dwa chatterboty są w stanie rozwinąć rozmowę między sobą, człowiek często przerywał ją z jednego z dwóch powodów. Pierwszym z nich była irytacja spowodowana ograniczonymi możliwościami programu, drugim traktowanie go przedmiotowo.

Oczekiwania jakie ludzie mieli względem programu w pierwszym przypadku były za wysokie. Nastawiali się oni na odpowiedzi z szeroko pojętej ludzkiej wiedzy. Gdy się okazywało, iż to co jest dla nich oczywiste nie jest zrozumiałe przez program, reagowali często negatywnym nastawieniem względem chatbota, kończąc wkrótce rozmowę, która do nikąd nie prowadziła.

Z drugiej strony, chatterbot był dla nich jedynie zbiorem komend. Oczekiwania względem niego były przez to za niskie i nie możliwe było zaprezentowanie swych możliwości przez program.

W rozmowie o typowo ludzkim podejściu do chatbota, odpowiedzi Lokiego przybierały najczęściej charakter polityka. Na zadane pytania zwracał często również pytania bądź mniej rozbudowane, niewiele znaczące wtrącenia. Gdy jednak wprowadzone przez rozmówcę zdanie było bardziej konkretne, Loki informatywnie na nie odpowiadał. Stworzyło to iluzję zarówno pamięci bota na temat wcześniejszej rozmowy, jak i jego zainteresowania rozmówcą. Nawet nieliczne wyrażenia, które nie zostały przez chatbota rozpoznane, nie wywarły dużego wpływu na płynność dialogu.

Wraz z rozwojem bazy danych i wprowadzanych przez autora pracy kolejnych modyfikacjach w plikach AIML, chatbot Loki prezentował się równie dobrze co bardziej znane, starsze chatboty. Momentami Loki mógł mieć nawet lepsze od nich reakcje, gdy pytania schodziły na temat upodobań, które u niego zostały mocno rozwinięte. Wyjątek stanowił Jabberwacky, który posiada jedną z najobszerniejszych baz wiedzy i który cały czas się rozwija, poprzez rozmowy z tysiącami ludźmi dziennie.

5.3 Przemyślenia

Minie jeszcze dużo czasu zanim powstanie program mogący naśladować ludzkie zachowanie. Nie tylko ograniczone są możliwości języków programowania pod tym względem, ale również podejście ludzi do tego typu programów. Traktując sztucznego rozmówcę jak program wykonywujący jakieś podstawowe polecenia i reagujący na jakieś określone słowo, ograniczamy jego rozwój jako ciekawego rozmówcy.

Znaczny wpływ na to ma brak bezpośredniego kontaktu. Człowiek nadając zwierzętom i przedmiotom konkretne cechy, pomija przy tym rzeczy trudne do określenia. Gdyby sztuczny rozmówca zawarty był wewnątrz jakiegoś konkretnego ciała, otrzymałby dzięki niemu pewne cechy. Robot w kształcie psa bądź kota traktowany byłby jak towarzysz i zwierzę. Robot humanoidalny byłby ludzki. Sam kod programu wykonujący polecenia jest jednak trudny do sklasyfikowania, zwłaszcza jeśli jest dostępny z komputera, który już otrzymał w jakiś sposób swoje własne "cechy".

Aby ze sztucznego rozmówcy uczynić coś więcej niż zaawansowany system ekspercki, każde słowo we wprowadzonym pytaniu musiałoby zostać rozpoznane i sklasyfikowane. Chatboty opierające się przede wszystkim na sposobie Elizy* nie są przyszłością sztucznej inteligencji, a jedynie narzędziem do wymiany informacji. Autor pracy uważa, że aby program był świadomy tego co robi, powinien zawierać w sobie algorytm sprawdzania zwrotu na wejściu, który rozpoznałby przynajmniej trzy części zdania: podmiot, orzeczenie, dopełnienie. Powinien on także mieć zapisane cały czas w swej pamięci owe wyrażenia z przynajmniej dwóch ostatnich wypowiedzi własnych jak i rozmówcy. W ten sposób zapamiętany będzie rodzaj więzi między konkretnymi słowami i możliwe przeszukiwanie na dwa sposoby, rozpatrując najpierw podmiot, bądź dopełnienie.

Ponadto, zwracana przez chatbota odpowiedź powinna opierać się na całej informacji wprowadzonej na wejściu. Potrzebny byłby osobny algorytm sprawdzający, czy poszczególne sentencje należą do najczęściej omawianego w rozmowie tematu. Podejmowałby on decyzję, któremu zdaniu dać priorytet pierwszeństwa i wyszukać na jego bazie odpowiedź.

Powyższe działania sprowadziłyby wyszukiwanie odpowiedzi w bazach danych programu do podrzędnego działania, przeprowadzanego na końcu. Stworzone algorytmy rozpoznania i logicznych obliczeń stanowiłyby nadrzędne operacje chatterbota. Duża liczba operacji przeprowadzanych dla każdej wiadomości wejściowej zwiększyłaby znacznie wymagania sprzętowe oraz doprowadziłaby do powstania kilku osobnych baz danych. Autor pracy uważa jednak, że rozpoznanie przez chatbota znaczenia wprowadzonego pytania jest jednym z najważniejszych elementów rozmowy. Bez niego futurystyczne wizje rozmów między ludźmi a maszynami pozostaną fikcją. Technika komputerowa cały czas ulega rozwojowi i większe wymagania programu nie są przez to dużym problemem.

* znalezienie w wiadomości wejściowej ciągu znaków będących słowem kluczowym wywołującym odpowiedź

Bibliografia

1. Bush N., *"Why You Don't Need Proprietary Bot Software"*, www.alicebot.org, 2001.
2. Copple K. L., *"Bringing AI to Life: Putting Today's Tools and Resources to Work"*, Parsing the Turing Test, Springer, 2009.
3. Demchenko E. & Veselov V., *"Who Fools Whom? The Great Mystification, or Methodological Issues on Making Fools of Human Beings"*, Parsing the Turing Test, Springer, 2009.
4. Deryugina O. V., *"Chatterbots"*, Scientific and Technical Information Processing, 2010, Vol.37, No.2.
5. Edmonds B., *"The Social Embedding of Intelligence: Towards Producing a Machine that Could Pass the Turing Test"*, Parsing the Turing Test, Springer, 2009.
6. Floridi L. & Taddeo M. & Turilli M., *"Turing's Imitation Game: Still an Impossible Challenge for All Machines and Some Judges—An Evaluation of the 2008 Loebner Contest"*, Minds & Machines 19, 2009.
7. Lenat D. B., *"Building a Machine Smart Enough to Pass the Turing Test: Could We, Should We, Will We?"*, Parsing the Turing Test, Springer, 2009.
8. Mauldin M. L., *"Going Under Cover: Passing as Human; Artificial Interest: A Step on the Road to AI"*, Parsing the Turing Test, Springer, 2009.
9. Nikravesh M., *"Evolution of Fuzzy Logic: From Intelligent Systems and Computation to Human Mind"*, Studies in Fuzziness and Soft Computing, 2007, Volume 217/2007.
10. Pellen L., *"How not to Imitate a Human Being: An Essay on Passing the Turing Test"*, Parsing the Turing Test, Springer, 2009.
11. Ringate T., *"AIML Reference Manual"* [online], 2001-09-01 [dostęp: 2012-04-03], Dostępny w Internecie: <http://www.alicebot.org/documentation/aiml-reference.html>
12. Turing A. M., *"Computing Machinery and Intelligence"*, Parsing the Turing Test, Springer, 2009.
13. Wallace R. S., *"The Anatomy of A.L.I.C.E."*, Parsing the Turing Test, Springer, 2009.
14. Zakos J. & Capper L., *"CLIVE - An Artificially Intelligent Chat Robot for Conversational Language Practice"*, Lecture Notes in Computer Science, 2008, Volume 5138/2008.
15. ALICE A.I. Foundation [online], Wallace R. S., 2006-2012 [dostęp: 2012-04-03], <http://www.alicebot.org>
16. Free AIML chat bot content: Badanswer.aiml [online], 2005 [dostęp: 2012-04-03]. Dostępny w Internecie: <http://alicebot.wikidot.com/AIML:aaa:badanswer-AIML>
17. FreeHAL Website [online], Tobias Schulz & Co., 2006-2011 [dostęp: 2012-04-03], <http://www.freehal.org/>
18. Jabberwacky [online], Rollo Carpenter, 1997-2011 [dostęp: 2012-04-03], <http://www.jabberwacky.com/>
19. MyCyberTwin [online], 2011 [dostęp: 2012-04-03], <http://site.mycybertwin.com/>
20. Verbots® - Open Source Natural Language Chatterbots [online], Conversive, Inc., 2004-2009 [dostęp: 2012-04-03], <http://www.verbots.com/>
21. Pandorabots – A Multilingual Chatbot Hosting Service [online], Pandorabots, Inc., 2002-2012 [dostęp: 2012-04-03], <http://www.pandorabots.com/>