

ELIZA - Program komputerowy do badania komunikacji w języku naturalnym między człowiekiem a maszyną

Joseph Weizenbaum

[Massachusetts Institute of Technology](#)

[Department of Electrical Engineering](#)

[Cambridge, Mass...](#)

[Communications of the ACM](#) Volume 9, Number 1 (styczeń 1966): 36-35.

Prace opisane w niniejszym dokumencie były wspierane (częściowo) przez [Project MAC](#), program badawczy [MIT](#) sponsorowany przez [Agencję Zaawansowanych Projektów Badawczych](#), [Departament Obrony](#), w ramach kontraktu [Biura Badań Marynarki Wojennej o](#) numerze Nonr-4102(01).

Streszczenie

ELIZA to program działający w ramach systemu podziału czasu [MAC](#) na MIT, który umożliwia pewne rodzaje konwersacji w języku naturalnym między człowiekiem a komputerem. Zdania wejściowe są analizowane na podstawie reguł dekompozycji, które są uruchamiane przez słowa kluczowe pojawiające się w tekście wejściowym. Odpowiedzi są generowane przez reguły ponownego montażu powiązane z wybranymi regułami dekompozycji. Podstawowe problemy techniczne, którymi zajmuje się ELIZA to:

1. identyfikacja słów kluczowych,
2. odkrycie minimalnego kontekstu,
3. wybór odpowiednich przekształceń,
4. generowanie odpowiedzi w przypadku braku słów kluczowych, oraz
5. zapewnienie końcowej pojemności dla "skryptów" ELIZA.

Artykuł kończy dyskusja na temat niektórych kwestii psychologicznych istotnych dla podejścia ELIZA, a także przyszłych zmian.

Wprowadzenie

Mówi się, że wyjaśniać znaczy wyjaśniać. Ta maksyma nigdzie nie sprawdza się tak dobrze, jak w dziedzinie programowania komputerowego, zwłaszcza w tak zwanym programowaniu heurystycznym i sztucznej inteligencji. W tych dziedzinach maszyny zachowują się w cudowny sposób, często wystarczający, by oszołomić nawet najbardziej doświadczonego obserwatora. Ale gdy dany program zostanie zdemaskowany, gdy jego wewnętrzne działanie zostanie wyjaśnione w języku wystarczająco prostym, aby wzbudzić zrozumienie, jego magia rozpada się; ujawnia się jako zwykły zbiór procedur, z których każda jest całkiem zrozumiała. Obserwator mówi do siebie: "Mógłbym to

Celem tego artykułu jest spowodowanie takiej właśnie ponownej oceny programu, który ma zostać "wyjaśniony". Niewiele programów kiedykolwiek potrzebowało tego bardziej.

Program ELIZA

ELIZA to program umożliwiający konwersację z komputerem w języku naturalnym. Jego obecna implementacja znajduje się w systemie podziału czasu MAC na MIT. Został napisany w języku MAD-SLIP [4] dla IBM 7094. Jego nazwa została wybrana, aby podkreślić, że może być stopniowo ulepszany przez użytkowników, ponieważ jego umiejętności językowe mogą być stale ulepszone przez "nauczyciela". Podobnie jak Eliza ze słynnego [Pigmaliона](#), można sprawić, by wyglądał na jeszcze bardziej cywilizowanego, jednak stosunek wyglądu do rzeczywistości pozostaje domeną dramaturga.

Dla obecnego celu wystarczy scharakteryzować system MAC jako taki, który pozwala jednostce obsługiwać pełnowymiarowy komputer ze zdalnie zlokalizowanej maszyny do pisania. Indywidualny operator ma złudzenie, że jest jedynym użytkownikiem kompleksu komputerowego, podczas gdy w rzeczywistości inni mogą "dzielić czas" z nim. Ważne jest tutaj to, że komputer może odczytywać wiadomości wpisywane na maszynie do pisania i odpowiadać poprzez pisanie na tym samym instrumencie. Czas pomiędzy otrzymaniem wiadomości przez komputer a pojawieniem się odpowiedzi jest funkcją programu kontrolującego okno dialogowe oraz takich parametrów systemu MAC jak liczba użytkowników aktualnie korespondujących z systemem. Te ostatnie parametry generalnie przyczyniają się w tak niewielkim stopniu do ogólnego czasu odpowiedzi, że interakcja konwersacyjna z komputerem nigdy nie musi wiązać się z naprawdę nieznośnymi opóźnieniami.

Podczas rozmowy z ELIZA użytkownik wpisuje jakieś stwierdzenie lub zestaw stwierdzeń w języku naturalnym, używając normalnej interpunkcji i struktury zdań. Jedynie znak zapytania nie może być używany, ponieważ jest on interpretowany przez system MAC jako znak kasowania linii. Wypowiedź użytkownika jest zakończona podwójnym powrotem karetki, który służy do przekazania kontroli do ELIZA. Następnie ELIZA analizuje instrukcję użytkownika i generuje odpowiedź, którą wpisuje. Następnie kontrola ponownie należy do użytkownika.

Typowa rozmowa wygląda następująco:

Wszyscy mężczyźni są tacy sami.

W JAKI SPOSÓB?

Zawsze nas o coś zaczepiają. MOŻESZ PODAĆ JAKIŚ

KONKRETNY PRZYKŁAD?

Cóż, mój chłopak zmusił mnie do przyjscia tutaj.

TWÓJ CHŁOPAK SPRAWIŁ, ŻE TU PRZYSZŁAŚ

Mówi, że przez większość czasu jestem w depresji.

PRZYKRO MI SŁYSZEĆ, ŻE JESTEŚ PRZYGNĘBIONY

To prawda. Jestem nieszczęśliwy

CZY MYŚLISZ, ŻE PRZYJAZD TUTAJ POMOŻE CI NIE BYĆ NIESZCZĘŚLIWYM?

Potrzebuję pomocy, to wydaje się pewne.

CO BY TO DLA CIEBIE ZNACZYŁO, GDYBYŚ OTRZYMAŁ POMOC?

Może nauczyłbym się dogadywać z matką.

OPOWIEDZ MI WIĘCEJ O SWOJEJ RODZINIE

Moja matka się mną opiekuje.

KTO JESZCZE W TWOJEJ RODZINIE SIĘ TOBĄ OPIEKUJE?

Mój ojciec.

TWÓJ OJCIEC

Pod pewnymi względami jesteś podobny

do mojego ojca. JAKIE WIDZISZ

PODOBIENSTWO?

Nie jesteś zbyt agresywny, ale myślę, że nie chcesz, żebym to zauważył.

DLACZEGO UWAŻASZ, ŻE NIE JESTEM ZBYT AGRESYWNY?

Nie kłóć się ze mną.

DLACZEGO UWAŻASZ, ŻE SIĘ Z TOBĄ NIE KŁÓCĘ?

Boisz się mnie.

CZY PODOBA CI SIĘ, ŻE SIĘ CIEBIE BOJĘ?

Mój ojciec boi się wszystkich.

CO JESZCZE PRZYCHODZI CI DO GŁOWY, GDY MYŚLISZ O SWOIM OJCU?

Prześladowcy.

CZY TO MA COŚ WSPÓLNEGO Z FAKTEM, ŻE TWÓJ CHŁOPAK ZMUSIŁ CIĘ DO
PRZYJŚCIA TUTAJ?

Linie pisane wielkimi literami to odpowiedzi maszyny.

Procedura brutto programu jest dość prosta; dane wejściowe są odczytywane i sprawdzane pod kątem
obecności *słowa kluczowego*. Gdy takie słowo zostanie znalezione, zdanie jest przekształcane zgodnie

z regułą powiązaną z tym słowem. 12/13/10 1:58 PM

ze słowem kluczowym, jeśli nie jest to uwaga bez treści lub, pod pewnymi warunkami, pobierana jest wcześniejsza transformacja. Tak obliczony lub pobrany tekst jest następnie drukowany.

W szczegółach, oczywiście, procedura naszkicowana powyżej jest znacznie bardziej złożona. Słowa kluczowe, na przykład, mogą mieć numer RANK lub numer pierwszeństwa. Procedura jest wrażliwa na takie numery, ponieważ porzuci słowo kluczowe już znalezione podczas skanowania tekstu od lewej do prawej na rzecz słowa o wyższej randze. Procedura rozpoznaje również przecinek lub kropkę jako separator. Za każdym razem, gdy jedno z nich zostanie napotkane, a słowo kluczowe zostało już znalezione, cały kolejny tekst jest usuwany z wiadomości wejściowej. Jeśli żaden klucz nie został jeszcze znaleziony, fraza lub zdanie po lewej stronie ogranicznika (jak również sam ogranicznik) są usuwane. W rezultacie przekształcane są tylko pojedyncze frazy lub zdania.

Słowa kluczowe i powiązane z nimi reguły transformacji stanowią SCRIPT dla określonej klasy konwersacji. [Słowo "transformacja" jest używane w jego ogólnym znaczeniu, a nie w znaczeniu nadanym mu przez [Harrisa](#) i [Chomsky'](#)ego w kontekście językowym]. Ważną właściwością ELIZA jest to, że skrypt jest danymi; tj. nie jest częścią samego programu. W związku z tym ELIZA nie ogranicza się do określonego zestawu wzorców rozpoznawania lub odpowiedzi, a nawet do żadnego konkretnego języka. Skrypty ELIZA istnieją (w chwili pisania tego tekstu) w języku walijskim i niemieckim, a także w języku angielskim.

Podstawowe problemy techniczne, którymi musi zająć się ELIZA, są następujące:

1. Identyfikacja "najważniejszego" słowa kluczowego występującego w wiadomości wejściowej.
2. Identyfikacja pewnego minimalnego kontekstu, w którym pojawia się wybrane słowo kluczowe; np. jeśli słowem kluczowym jest "ty", czy następuje po nim słowo "są" (w takim przypadku prawdopodobnie formułowane jest twierdzenie).
3. Wybór odpowiedniej reguły transformacji i oczywiście samo dokonanie transformacji.
4. Zapewnienie mechanizmu, który pozwoli ELIZA reagować "inteligentnie", gdy tekst wejściowy nie zawiera słów kluczowych.
5. Zapewnienie maszyn, które ułatwiają edycję, w szczególności rozszerzenie, scenariusza na poziomie pisania scenariusza.

Istnieją oczywiście zwykłe ograniczenia podyktowane potrzebą ekonomicznego wykorzystania czasu komputera i przestrzeni dyskowej.

Centralną kwestią jest oczywiście manipulacja tekstem, a jej sercem jest pojęcie *reguły transformacji*, o której mówi się, że jest związana z pewnymi słowami kluczowymi. Mechanizmy kryjące się pod hasłem "reguła transformacji" to szereg funkcji Slip, które służą do (1) dekomponowania ciągu danych zgodnie z pewnymi kryteriami, a zatem do testowania ciągu pod kątem tego, czy spełnia te kryteria, czy nie, oraz (2) do ponownego składania zdekomponowanego ciągu zgodnie z pewnymi specyfikacjami montażu.

Chociaż nie jest to miejsce na omawianie tych funkcji we wszystkich szczegółach (lub nawet na ujawnienie ich pełnej mocy i ogólności), ważne jest, aby zrozumieć działanie ELIZA, aby opisać je szczegółowo.

Rozważmy zdanie "Jestem ostatnio bardzo nieszczęśliwy". Załóżmy, że obcokrajowiec z ograniczoną znajomością języka angielskiego, ale z bardzo dobrym słuchem, usłyszał to zdanie wypowiedziane, ale

pierwsze dwa słowa "jestem". Chcąc wyglądać na zainteresowanego, a może nawet współczującego, może odpowiedzieć: "Jak długo jesteś ostatnio bardzo nieszczęśliwy?". To, co musiał zrobić, to zastosować rodzaj szablonu do oryginalnego zdania, którego jedna część pasowała do dwóch słów "jestem", a pozostała część izolowała słowa "bardzo nieszczęśliwy ostatnio". Musi on również posiadać zestaw do ponownego montażu, który jest specjalnie powiązany z tym szablonem, który określa, że każde zdanie w formie "Jestem BLAH" może zostać przekształcone w "Jak długo jesteś BLAH", niezależnie od znaczenia BLAH. Nieco bardziej skomplikowanym przykładem jest zdanie "Wygląda na to, że mnie nienawidzisz". Tutaj obcokrajowiec rozumie tylko słowa "ty" i "ja"; tj. stosuje szablon, który rozkłada zdanie na cztery części

(1)	(2)	(3)	(4)
Wygląda na to, że	ty	nien awis ć	ja

którego tylko druga i czwarta część są zrozumiałe. Reguła ponownego złożenia może wtedy brzmieć "Co sprawia, że myślisz, że cię nienawidzę"; tj. może odrzucić pierwszy komponent, przetłumaczyć dwa znane słowa ("ty" na "ja" i "ja" na "ty") i dołączyć frazę standardową (Co sprawia, że myślisz) na początku rekonstrukcji. Formalny zapis, w którym można przedstawić szablon dekompozycji, to

(0 YOU 0 ME)

i reguła ponownego montażu

(CO SPRAWIA, ŻE MYŚLISZ, ŻE MAM 3 LATA) .

"0" w regule dekompozycji oznacza "i nieokreśloną liczbę słów" (analogicznie do nieokreślonego znaku dolara w COMMIT) [6], podczas gdy "3" w regule ponownego montażu wskazuje, że trzeci składnik dekompozycji tematu ma zostać wstawiony na jego miejsce. Reguła dekompozycji

(0 YOU 1 ME)

działałaby równie dobrze w tym konkretnym przykładzie. Niezerowa liczba całkowita "*n*" pojawiająca się w regule dekompozycji wskazuje, że dany element powinien składać się z dokładnie "*n*" słów. Jednak z dwóch pokazanych reguł, tylko pierwsza *pasowałaby* do zdania. "Wygląda na to, że mnie kochasz i nienawidzisz", druga zawodzi, ponieważ między "ty" i "ja" znajduje się więcej niż jedno słowo.

W ELIZA kwestia tego, które reguły dekompozycji zastosować do tekstu wejściowego, jest oczywiście kluczowa. Zdanie wejściowe mogło brzmieć na przykład "Wygląda na to, że nienawidzisz", w którym to przypadku reguła dekompozycji (0 YOU 0 ME) zawiodłaby, ponieważ słowo "ME" nie zostałoby w ogóle znalezione, nie mówiąc już o przypisanym mu miejscu. W takim przypadku musiałaby zostać wypróbowana inna reguła dekompozycji, a w przypadku jej braku jeszcze inna, aż do uzyskania dopasowania lub całkowitego niepowodzenia. ELIZA musi zatem dysponować mechanizmem pozwalającym znacznie ograniczyć zestaw reguł dekompozycji, które mogą mieć zastosowanie do aktualnie aktywnego zdania wejściowego. Jest to mechanizm słów kluczowych.

Zdanie wejściowe jest skanowane od lewej do prawej. Każde słowo jest wyszukiwane w słowniku słów kluczowych. Jeśli słowo zostanie zidentyfikowane jako słowo kluczowe, wówczas (pomijając kwestię pierwszeństwa słów kluczowych) należy wypróbować tylko reguły dekompozycji

zawierające to słowo kluczowe. Sekwencja prób może być nawet częściowo uporządkowana. Na przykład reguła dekompozycji (0 YOU 0) powiązana ze słowem kluczowym "YOU" (i rozkładająca zdanie na

1. wszystkie słowa przed "TY",

2. słowo "TY", oraz
3. wszystkie słowa następujące po "YOU")

powinna być ostatnią próbą, ponieważ z pewnością się powiedzie.

Pojawiają się dwa problemy. Jeden wynika z faktu, że prawie żadne ze słów w danym zdaniu nie jest reprezentowane w słowniku słów kluczowych. Drugi dotyczy "kojarzenia" zarówno reguł dekompozycji, jak i ponownego składania ze słowami kluczowymi. Pierwszy z nich jest o tyle poważny, że ustalenie, czy danego słowa nie ma w słowniku, może wymagać więcej obliczeń (tj. czasu) niż zlokalizowanie słowa, które jest w nim reprezentowane. Atak na oba problemy rozpoczyna się od umieszczenia zarówno słowa kluczowego, jak i powiązanych z nim reguł na *liście*. Podstawowy format typowej listy kluczy jest następujący:

$$\begin{array}{ccccccc} (K & ((D_1) & (R_{1,1}) & (R_{1,2}) & \dots & (R_{1,m_1})) \\ & ((D_2) & (R_{2,1}) & (R_{2,2}) & \dots & (R_{2,m_2})) \\ & \cdot & & & & \cdot \\ & \cdot & & & & \cdot \\ & \cdot & & & & \cdot \\ & ((D_n) & (R_{n,1}) & (R_{n,2}) & \dots & (R_{n,m_n})) \end{array}$$

gdzie K jest słowem kluczowym, D_i i -tą regułą dekompozycji powiązaną z K , a R_{ij} j -tą regułą ponownego składania powiązaną z i -tą regułą dekompozycji.

Typowym obrazowym przedstawieniem takiej struktury jest diagram drzewa pokazany na rysunku 1. Najwyższy poziom tej struktury zawiera słowo kluczowe, po którym następują nazwy list; każda z nich jest ponownie strukturą listy rozpoczynającą się od reguły dekompozycji, po której następują reguły ponownego montażu. Ponieważ struktury list tego typu nie mają z góry określonych ograniczeń wymiarowych, dowolna liczba reguł dekompozycji może być powiązana z danym słowem kluczowym i dowolną liczbą reguł ponownego montażu z dowolną określoną regułą dekompozycji. SLIP jest bogaty w funkcje, które efektywnie sekwencjonują struktury tego typu. W związku z tym problemy programistyczne są zminimalizowane.

Skrypt ELIZA składa się głównie z zestawu struktur list pokazanego typu. Rzeczywisty katalog słów kluczowych jest konstruowany, gdy taki skrypt jest wczytywany do dotychczas pustego programu. Podstawowym składnikiem strukturalnym katalogu słów kluczowych jest wektor KEY (obecnie) 128 ciągłych słów komputerowych. Gdy odczytywana jest konkretna struktura listy kluczy, słowo kluczowe K na jej szczycie jest randomizowane (hashowane) przez procedurę, która wytwarza (obecnie) 7-bitową liczbę całkowitą " i ". Na przykład słowo "zawsze" daje liczbę całkowitą 14. $KEY(i)$, czyli i -te słowo wektora KEY, jest następnie sprawdzane w celu określenia, czy zawiera nazwę listy. Jeśli nie, tworzona jest pusta lista, jej nazwa umieszczana jest w $KEY(i)$, a dana struktura listy kluczy jest umieszczana na *tej* liście.

Dyskusja

W chwili pisania tego tekstu jedynymi poważnymi skryptami ELIZA, które istnieją, są te, które powodują, że ELIZA reaguje mniej więcej tak, jak niektórzy psychoterapeuci (Rogerianie). ELIZA działa najlepiej, gdy jej ludzki korespondent jest początkowo poinstruowany, aby "rozmawiać" z nią, oczywiście za pośrednictwem maszyny do pisania, tak jak z psychiatrą. Ten tryb rozmowy został

wybrany, ponieważ wywiad psychiatryczny jest jednym z niewielu przykładów skategoryzowanej diadycznej komunikacji w języku naturalnym, w której jeden z uczestników

Para może swobodnie przyjąć pozę niewiedzy prawie nic o prawdziwym świecie. Jeśli, na przykład, ktoś powiedziałby psychiatrze "Poszedłem na długą przejażdżkę łodzią", a on odpowiedziałby "Opowiedz mi o łodziach", nie można by założyć, że nie wiedział nic o łodziach, ale że miał jakiś cel w kierowaniu dalszą rozmową. Ważne jest, aby zauważyć, że to założenie jest dokonywane przez mówiącego.

To, czy jest to realistyczne, czy nie, jest zupełnie odrębną kwestią. W każdym razie ma to kluczowe znaczenie psychologiczne, ponieważ służy mówcy do utrzymania poczucia bycia wysłuchanym i zrozumianym. Mówca dalej broni swojego wrażenia (które nawet w prawdziwym życiu może być iluzoryczne), przypisując swojemu partnerowi konwersacyjnemu wszelkiego rodzaju wiedzę, spostrzeżenia i zdolność rozumowania. Są one jednak wkładem *mówiącego* w rozmowę.

Referencje

- 1.
- 2.
3. Weizenbaum, J. Symmetric list processor. [Comm. ACM](#) 6, (wrzesień 1963), 524-544.
- 4.
5. Yngve, J. [COMIT Programming Manual](#). MIT Press, Cambridge, Mass., 1961. 6.

DODATEK. Skrypt Elizy