

Archipeląg sztucznej inteligencji. Część IV

Ryszard Tadeusiewicz

Wprowadzenie

Artykuł ten jest (jak wynika z tytułu) czwartym z serii artykułów poświęconych przeglądowej prezentacji poszczególnych metod sztucznej inteligencji (AI) prezentowanych jako wyspy archipelagu. Wyjaśnienie, dlaczego przyjęto taką właśnie metaforę, znaleźć można w pierwszym artykule tego cyklu, opublikowanym w numerze 12/2020 miesięcznika „Napędy i Sterowanie”. W tym samym artykule, zapoczątkującym cały cykl, zaproponowałem zasadę, że chociaż mamy tu do czynienia z metaforami (gdy mowa o wyspach i o archipelagu), to jednak nazw tych nie będę ujmował w cudzysłów, pozostawiając właściwą interpretację domyślności Czytelnika.

Do tej pory ukazały się trzy artykuły z prezentowanego cyklu. W pierwszy z nich (NiS nr 12/2020) zaprezentowano metody AI oparte na stosowaniu rachunku symbolicznego, sieci neuronowe oraz systemy ekspertowe. Drugi artykuł (NiS nr 1/2021) zawierał opisy zbiorów rozmytych i logiki rozmytej, zbiorów przybliżonych i rozpoznawania obrazów. Trzeci artykuł (NiS nr 2/2021) omawiał analizę skupień, algorytmy genetyczne i algorytmy mrówkowe. Zawartość czwartego odcinka cyklu znajduje się poniżej.

Programy grające z ludźmi w gry strategiczne

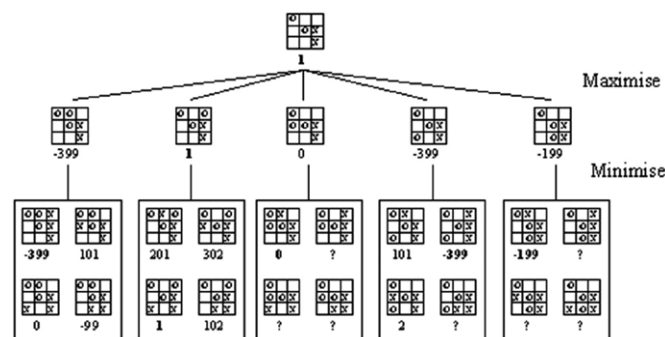
Od początku istnienia sztucznej inteligencji twórców jej metod fascynowało tworzenie programów wykazujących inteligentne działanie podczas rywalizacji z człowiekiem w grach strategicznych [1]. Wyspa omawianego w tym cyklu archipelagu, reprezentująca metody sztucznej inteligencji wykorzystywane w tych programach rozgrywających, może być przedstawiona tak, jak na rysunku 1.

Historia zaczęła się od programów komputerowych grających z ludźmi w warcaby – grę prostszą i łatwiejszą od szachów. Pierwsze programy grające w tę grę były bardzo mało



Rys. 1. Wyspa będąca metaforą programów grających w gry strategiczne

inteligentne i łatwo pozwalały się pokonać. Ale już w 1954 roku, czyli u zarania techniki komputerowej, profesor A. Samuel pracujący dla IBM stworzył program, który pokonał nawet czwartego gracza w rankingu Stanów Zjednoczonych. Sprawą zainteresowało się wojsko i zaczęło finansować badania zmierzające do stworzenia programów grających w gry strategiczne, zakładając, że takie programy odpowiednio udoskonalone mogą pełnić rolę doradców dowódców na prawdziwym placu boju. Rozwój nie następował jednak tak szybko, jak by chciała armia, i ludzie ciągle jednak wygrywali z komputerami mimo dużego zaangażowania twórców. Dopiero w 1992 roku program Chinook stworzony przez J. Schaerffera „pokazał lwi pazur”. Program ten wygrał rywalizację US Open, ale w mistrzostwach zorganizowanych w Londynie przegrał z dr M. Tinsleyem wynikiem 4:2 po 33 remisach. Jednak w 1995 roku udoskonalony Chinook potrafił przewidywać dalekosiężne skutki każdego ruchu (rys. 2) i wygrywał wszystkie mecze [2].



Rys. 2. Sposób działania komputera przy układaniu strategii gry

Trzeba było przyjąć do wiadomości, że w warcabach żaden człowiek nie dorówna maszynie, więc można w nie grać dla przyjemności tylko z innym człowiekiem albo z komputerem mającym celowo wprowadzone ograniczenia, natomiast gdy odpowiedni program osiągnie pełną moc przetwarzania informacji – to do gry o przesądzonej przegranej nie ma co siadać.

Grą, która prawdziwie budziła emocję naukowców i kibiców, były szachy. Pierwsze teoretyczne prace na temat możliwości stworzenia programu do gry w szachy pisali „ojcowie informatyki” Norbert Wiener, Claude Shannon i Alan Turing w latach 1948–1951. Pierwsze działające programy powstały w 1957 roku w USA i ZSRR. W latach 1966–1967 rozgrywany był pierwszy mecz szachowy między programem radzieckim ITEP (Instytut Fizyki Teoretycznej i Eksperymentalnej w Moskwie) a amerykańskim programem Kotok-McCarthy (Uniwersytet Stanforda). Mecz trwał 9 miesięcy i zakończył się wygraną rosyjskiego

reklama

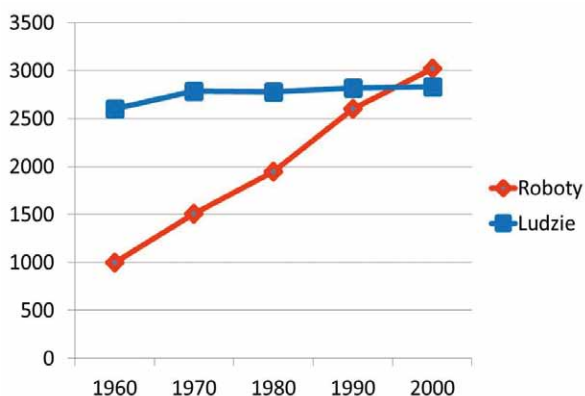


Rys. 3. David Levy – twórca słynnego zakładu dotyczącego szachów komputerowych

programu. Niestety twórca tego sukcesu, dyrektor ITEP Aleksander Kronrod, został wyrzucony ze stanowiska i pozbawiony tytułu profesorskiego za to, że używał komputera mającego projektować bomby atomowe do jakichś szachowych turniejów. Od tej pory programy do komputerowej gry w szachy powstawały głównie w USA. Programy te początkowo grały słabo, więc w roku 1968 szkocki mistrz szachowy David Levy (rys. 3) ufundował nagrodę dla programu, który ogra go chociaż raz na cztery partie.

Ale programy szachowe stale się doskonaliły i w 1989 roku program Deep Thought pokonał Davida Levyego, wygrywając wszystkie 4 partie.

Nie jest to właściwe miejsce, żeby opisywać owe kolejne, coraz sprawniejsze programy szachowe, ich sukcesy i porażki – ale warto spojrzeć na rysunek 4. Podano na nim punktację Elo przyznaną szachistom w poszczególnych latach. Niebieska linia pokazuje poziom gry kolejnych arcymistrzów, a czerwona (opisana „Roboty”) – poziom gry stale doskonalonych programów szachowych.



Rys. 4. Ranking kolejnych programów szachowych i kolejnych arcymistrzów wg punktacji Elo

Jak widać, programy szachowe stale zbliżały się do poziomu gry najlepszych szachistów (ludzi) i nieuchronnie musiał nadejść moment, kiedy program szachowy wygra z arcymistrzem. Chwila ta nadeszła w maju 1997 roku. Ówczesny



Rys. 5. Kasparow przegrywa z komputerem Deep Blue

niekwestionowany mistrz świata, Garri Kasparow, przegrał mecz szachowy z programem komputerowym Deep Blue (rys. 5).

To zwycięstwo komputera nad arcymistrzem miało doniosłe skutki. Po pierwsze, obaliło dość rozpowszechnione przekonanie, że sztucznej inteligencji jako takiej nie ma, ponieważ pozornie inteligentne zachowania komputerów są z góry zaprogramowane i tym, co działa, jest w istocie inteligencja programisty „zaszyta” w komputerowym programie. Tymczasem Kasparow był niekwestionowanym mistrzem świata (rys. 6). Wygrywał z każdym, kto się ośmielił z nim zmierzyć.

Można powiedzieć, że na świecie nie było człowieka, który by wygrał z Kasparowem. Tym bardziej nie było programisty, który by mógł umieścić w programie metodę, jak tego mistrza pokonać. A jednak Deep Blue wygrał. To znaczy, że **sam musiał znaleźć zwycięskie strategie** – a zatem działał inteligentnie!

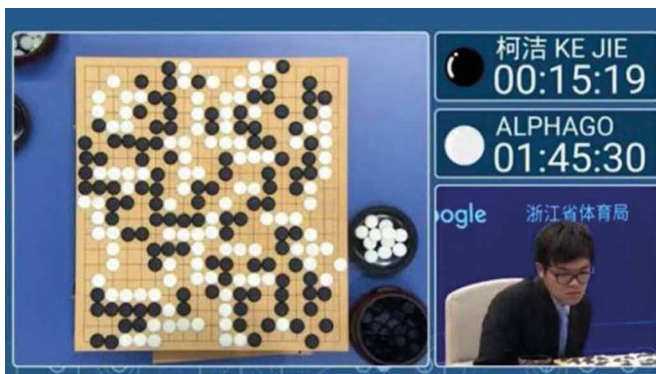
Od 1997 roku programy szachowe są nadal doskonalone, więc podobnie jak w warcabach nie ma już sensu prowadzenie rozgrywek w układzie człowiek przeciw maszynie, bo wynik jest przesądzony – ale szachy są na tyle ciekawą grą, że ludzie nadal rywalizują ze sobą, rozgrywają turnieje, wyłaniają mistrzów itp. Pojawił się jednak problem „dopingu”: przyłapano kilku szachistów na tym, że wychodząc w trakcie meczu do toalety, zabierali ze sobą smartfony i korzystali z podpowiedzi komputerowych programów szachowych, wykorzystując je potem podczas rozgrywki. Posypały się dyskwalifikacje... Czy u narodzin pierwszych programów szachowych ktoś mógł to przewidzieć?

Szachy są grą trudną i wymagającą, ale prawdziwym wyzwaniem była dalekowschodnia gra Go. Owa gra jest o wiele trudniejsza niż szachy, co widać, gdy się porówna kilka liczb. Liczba



Rys. 6. Kasparow po jednym z licznych tryumfów podczas szachowych mistrzostw świata

ruchów, jakie trzeba wykonać w trakcie partii Go, to średnio 260. Szachy to średnio tylko 35. Liczba możliwych do rozegrania partii w Go to 10^{260} . W szachach to 10^{123} . Liczba różnych pozycji na planszy w Go to 10^{172} . W szachach to 10^{46} . Opierając się na tych wyliczeniach, różni opiniodawcy sztucznej inteligencji twierdzili (zwłaszcza po sukcesie Deep Blue w szachach), że w Go żaden komputer z człowiekiem nie wygra. Nie trzeba było długo czekać, by się przekonać, że ta zapowiedź także okazała się nieprawdziwa. Program AlphaGo, stworzony przez firmę DeepMind, ale rozwijany potem przez Google, które 26.01.2014 roku przejęły DeepMind za 500 mln USD, pokonał najpierw (2015) Mistrza Europy w Go, Fana Huia, a w marcu 2016 roku po czterodniowym meczu pokonał Lee Sedola, jednego z arcymistrzów tej gry. Specjaliści spierali się, czy Lee Sedol był drugim najlepszym graczem w Go, czy może czwartym, ale ta dyskusja została przecięta, gdy AlphaGo w maju 2017 roku pokonał dwukrotnie niekwestionowanego mistrza świata w Go, Ke Jie (rys. 7). Ogólnie w 2017 roku AlphaGo rozegrał w internecie 60 rozgrywek Go i wszystkie wygrał.



Rys. 7. Rozgrywka w Go między programem AlphaGo a arcymistrzem Ke Jie

Przetwarzanie języka naturalnego

Przechodząc do kolejnej wyspy opisywanego tu archipelagu, napotykamy wizerunek przedstawiony na rysunku 8.

Wyspa jest ogólnie znana, a jej najważniejszym elementem jest księga z ważnym napisem. I właśnie o ten napis chodzi, bo cywilizacja ludzka od czasów starożytnych gromadzi wiedzę



Rys. 8. Wyspa symbolizująca metody przetwarzania języka naturalnego

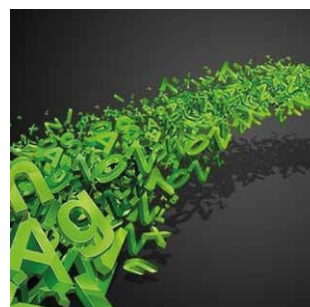
w postaci tekstów. Dawniej teksty te gromadzono w postaci glinianych tabliczek (Babilon), papirusów, zwojów pergaminów, ręcznie przepisywanych, a później drukowanych ksiąg, czasopiśm i wielu innych form zapisów, natomiast obecnie wszystkie nowe teksty mają formę elektroniczną, która podlega teleinformatycznej dystrybucji do czytelników, a w jakiejś części (coraz mniejszej procentowo) jest także drukowana na papierze.

Formę cyfrowych tekstów mają rejestrowane w mediach społecznościowych (głównie Facebook i Twitter, a także liczne blogi) wypowiedzi różnych ludzi, a także przechwytywane (często nielegalnie) treści sms i e-mail. Jest tego mnóstwo!

Również dawne dokumenty pisane i drukowane są w coraz większym zakresie przenoszone do formy cyfrowej za pomocą ich skanowania, przy czym użycie technik OCR (*Optical Character Recognition*), należącej do omawianej w drugiej części cyklu techniki rozpoznawania obrazów (NiS nr 1/2021), pozwala na zamianę obrazu skanowanych stron do formy cyfrowo kodowanych łańcuchów znaków [3]. Ta konwersja czasem wprowadza błędy, które wymagają „ręcznej” korekty, ale korzyści z takiej transformacji są ewidentne.

Również techniki zamiany mowy na tekst pisany (także należące do dziedziny rozpoznawania obrazów) pozwalają uzyskać nagrania wypowiedzi słownych w postaci łańcuchów znaków [4]. Do konwersji tekstu mówionego na tekst pisany służyć może cenione na świecie narzędzie, jakim jest program **IBM Watson Speech**. Niestety chwilowo nie radzi on sobie z zapisywaniem tekstów mówionych w języku polskim. Natomiast do pakietu Microsoft Office można już pobrać (za darmo!) program **Dictate**, który pozwala na głosowe sterowanie pracą komputera, ale także na rejestrowanie w formie tekstu całych wypowiedzi. Co więcej, Dictate może także tłumaczyć z jednego języka na drugi w ramach 20 języków, wśród których jest także język polski. Inna duża firma, Google, wypuściła program **VoiceNote II**, który współpracując z przeglądarką Google Chrome, dokonuje także konwersji mowy na tekst. Przez niektórych użytkowników polecany jest **SkryBot domowy**, szybki jest **Dictation Pro**, wygodny (ale drogi!) **MagicScribe**. Dyktowanie tekstu, który potem dostępny jest w formie pisanej, jest także jedną z usług, jakie może świadczyć system Google, a w nim narzędzie **Google Docs**. Ja sam budowałem kiedyś system, który na podstawie nagrań z przesłuchań policyjnych automatycznie sporządzał pisane protokoły, łatwiejsze do gromadzenia i przeszukiwania [5].

Jak widać, źródeł cyfrowych tekstów jest obecnie bardzo dużo. Efekt jest taki, że ogromna liczba ważnych informacji dostępna jest obecnie właśnie w takiej formie, która dla komputerów jest po prostu łańcuchem znaków (rys. 9).

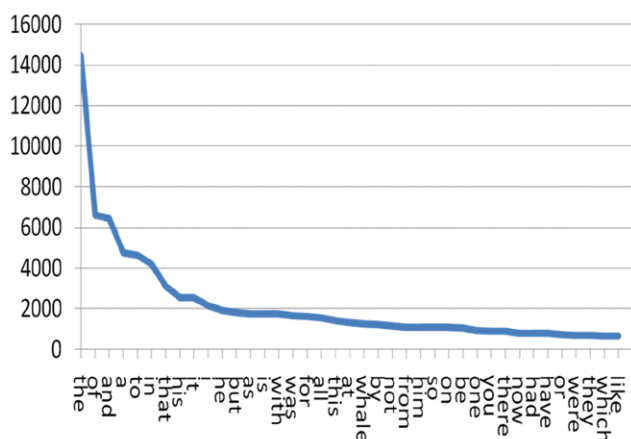


Rys. 9. Dla komputerów wszystkie teksty są wyłącznie łańcuchami znaków. Ich inteligentne wykorzystanie jest możliwe wyłącznie dzięki rozwojowi metod automatycznego przetwarzania języka naturalnego

Żeby mieć z tych cyfrowych tekstów konkretny pożytek, trzeba jednak było wypracować liczne narzędzia do przetwarzania języka naturalnego – i takie właśnie narzędzia zostały w łonie sztucznej inteligencji stworzone i udoskonalone.

Prowokacyjny dla badaczy sztucznej inteligencji był tak zwany test Turinga, opisany przez niego w wydanim w 1950 roku artykule „Computing Machinery and Intelligence”. Istotę tego testu opisałem w pierwszym odcinku tego cyklu (NiS nr 12/2020). Mnóstwo badaczy podejmowało więc próby budowy programów komputerowych, które mogłyby prowadzić z człowiekiem rozmowę na dowolny temat. Najstarsze publikacje innych autorów dotyczące tej tematyki opisałem w sposób popularny w artykułach [6, 7], a wyniki moich własnych prac badawczych związanych z budową pierwszych takich systemów dla języka polskiego, zostały opublikowane w artykułach [8] i [9], a potem retrospektywnie w opracowaniu [10].

W obszarze naukowym problematyka przetwarzania języka naturalnego przez długi czas opierała się na tworzeniu tak zwanych słowników częstościowych. Słownik taki w najprostszej postaci przypisuje do każdego wyrazu (czy nawet fleksyjnej formy wyrazu) częstość jego wystąpienia (rys. 10).



Rys. 10. Przykład słownika frekwencyjnego

Przykład pokazany na rysunku 10 dotyczy małej próbki tekstu w języku angielskim (stąd niewielkie liczby zliczeń częstości występowania wyrazów, podane na osi pionowej) i służy wyłącznie do ilustracji tego, czym jest słownik frekwencyjny, ale rzeczywiste słowniki buduje się w oparciu o teksty zawierające setki milionów słów. Bez pomocy komputerów takie badania były ogromnie uciążliwe. Ale w momencie wprowadzenia metod komputerowego przetwarzania języka naturalnego stało się możliwe tworzenie takich słowników zarówno dla całego języka (na przykład polskiego), jak i dla wybranych podzbiorów tego języka – na przykład artykułów prasowych, książek beletrystycznych, referatów konferencyjnych, przemówień polityków itp. Porównując takie „dziedzinyorientowane” słowniki frekwencyjne, językoznawcy mogą wyciągać daleko idące i empirycznie uzasadnione wnioski na temat właściwości poszczególnych podzbiorów języka. Co więcej, istnieje tzw. prawo Zipfa, które mówi, że kształt krzywej wynikającej ze

słownika frekwencyjnego jest charakterystyczny dla każdego autora, więc może służyć do ustalania autorstwa dzieł, których autentyczność budzi wątpliwości. Metodę tę zastosowano z dobrym skutkiem do analizy dzieł Szekspira.

Przechodząc do zastosowań praktycznych, metody przetwarzania języka naturalnego znajdują zastosowanie w przyjaznych dla użytkownika interfejsach, pozwalających na wygodne (z użyciem zdań języka naturalnego) stawianie zadań różnym systemom informatycznym oraz na uzyskiwanie od tych systemów odpowiedzi także formułowanych w zrozumiałej, prostej językowo formie. O takim interfejsie była mowa w pierwszym artykule tego cyklu (NiS nr 12/2020) przy okazji opisywania systemów ekspertowych.

Obecnie systemy do naturalnej konwersacji z ludźmi (nazywane niekiedy *chatterbot*) są też dość często budowane dla różnego rodzaju systemów doradczych albo *call centers* (rys. 11).

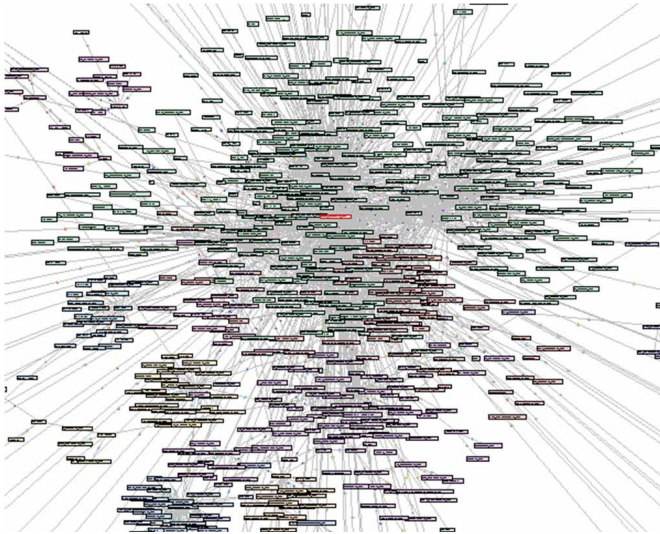


Rys. 11. Jeden z wielu systemów dialogowych typu *chatterbot*

Praktyczne korzyści z systemów przetwarzania języka naturalnego mamy też przy zastosowaniu ich do automatycznego tłumaczenia tekstów lub mówionych wypowiedzi z jednego języka na drugi. Systemy do automatycznej translacji tworzone i doskonalono przez blisko 70 lat (pierwsze były demonstrowane publicznie w 1954 roku) – i chociaż jakość komputerowego tłumaczenia była początkowo dość marna – obecnie systemy te mają już zdecydowanie zadowalającą sprawność. W powszechnym użyciu jest tu zwłaszcza łatwo dostępny i dobrze działający translator firmy Google.

Systemy przetwarzania języka naturalnego pełnią też ważną funkcję przy automatycznym porządkowaniu dużych zbiorów tekstów wraz z automatycznym ustalaniem związków pomiędzy nimi (rys. 12)

Cenione są także narzędzia wywodzące się z przetwarzania języka naturalnego, które pozwalają na automatyczne generowanie skrótów i streszczeń bardzo rozbudowanych zbiorów tekstów wyselekcjonowanych automatycznie metodami wybierania na podstawie słów kluczowych lub charakterystycznych fraz występujących w tekstach.



Rys. 12. Automatyczne ustalanie związków pomiędzy tekstami, zdaniami, a nawet pojedynczymi słowami za pomocą narzędzi przetwarzania języka naturalnego

Uczące się drzewa decyzyjne

Metaforą wyspy reprezentującej uczące się drzewa decyzyjne w archipelagu sztucznej inteligencji jest sztuczna wyspa Palma Dżamira zbudowana w Dubaju (rys. 13). Uczące się drzewa



Rys. 13. Wyspa w archipelagu sztucznej inteligencji symbolizująca uczące się drzewa decyzyjne

decyzyjne też są sztuczne i też są mocno rozgałęzione, więc analogii łatwo się tu dopatrzeć [11].

Generowanie reguł za pomocą drzew decyzyjnych jest jednym ze sposobów pozyskiwania wiedzy w sposób **indukcyjny**. Znaczenie indukcji w sztucznej inteligencji wynika z tego, że w wielu przypadkach nie dysponujemy wiedzą pozwalającą na stworzenie ogólnej metody rozwiązywania rozważanego zbioru problemów, więc niemożliwe jest zastosowanie metody dedukcji, natomiast mamy do dyspozycji przykłady zadań, które zostały prawidłowo rozwiązane. Metoda uczących się drzew decyzyjnych pozwala na zbudowanie efektywnej metody rozwiązywania różnych zadań, ale najczęściej jest wykorzystywana

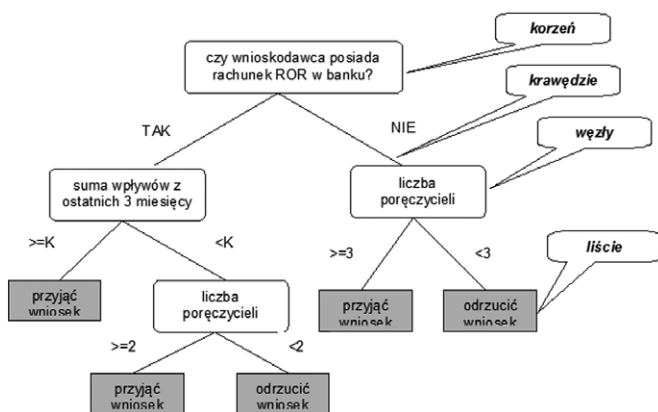
jako narzędzie wspomagające proces podejmowania decyzji [12].

Swoją popularność drzewa decyzyjne zawdzięczają dużej efektywności, możliwości prostej programowej implementacji oraz czytelnej dla człowieka reprezentacji.

Opisując istotę metody drzew decyzyjnych, najkrócej można stwierdzić, że jest to metoda pozyskiwania wiedzy opierająca się na analizie przykładów, z których każdy musi być opisany przez zestaw atrybutów (przesłanek będących podstawą podjęcia decyzji) oraz musi mieć podaną tę właściwą decyzję, którą ktoś gdzieś podjął – i która okazała się trafna. Metoda nadaje się do zastosowania w przypadku atrybutów przyjmujących wartości dyskretne. Jest to ważne, bo te dyskretne wartości będą wybierać kolejne gałęzie drzewa. W przypadku problemów decyzyjnych wymagających użycia atrybutów ciągłych dokonuje się zwykle ich dyskretyzacji, na przykład dzieląc zakres ich zmienności na przedziały i wybierając przedział zamiast konkretnej wartości.

Drzewa decyzyjne tworzone są w nieco przewrotny sposób, bo **korzeń** mają u góry, gałęzie rosną w dół, a zakończenia (czasem określane jako **liście**) znajdują się na samym dole. Korzeń związany jest z pierwszym (najważniejszym) atrybutem, a wychodzące z niego gałęzie (krawędzie grafu reprezentującego drzewo) dochodzą do kolejnych **węzłów**, w których umieszczone są następane atrybuty i z których wyrastają kolejne gałęzie. Rozbudowując systematycznie kolejne piętra drzewa, dochodzimy wreszcie do miejsca (**liścia**), w którym możliwe jest podanie wzorcowej decyzji. Obejrzyjmy takie przykładowe drzewo decyzyjne związane z podejmowaniem decyzji, czy klientowi banku zwracającemu się o przyznanie kredytu w wysokości **K** złotych – przyznać ten kredyt, czy nie?

Odpowiednie drzewo decyzyjne pokazane jest na rysunku 14.

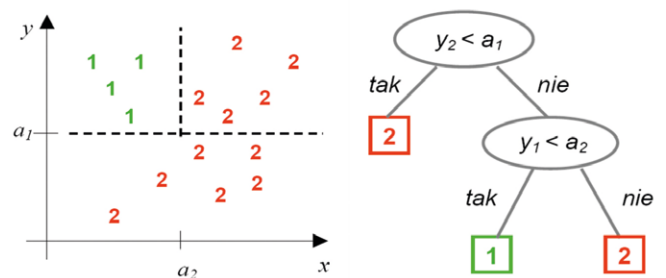


Rys. 14. Przykładowe drzewo decyzyjne

Jako korzeń w tym przypadku przyjęto najważniejszą okoliczność: czy wnioskodawca ma rachunek ROR (oszczędnościowo-rozliczeniowy) w tym banku. Jeśli tak, to posuwamy się do węzła określającego wpływy na ten rachunek w ciągu ostatnich trzech miesięcy i jeśli są one większe (lub równe) **K**, to można przyznać kredyt bez poręczycieli. Jeśli natomiast owe wpływy są małe, to wymaganych jest co najmniej dwóch poręczycieli. Jeśli natomiast wnioskodawca rachunku nie posiada – to warunkiem

przyznania kredytu jest przedstawienie co najmniej trzech poręczycieli. Odpowiednie decyzje końcowe pokazane są jako „liście” na końcu odpowiednich gałęzi.

W przypadku atrybutów przyjmujących wartości liczbowe drzewo decyzyjne może być utożsamiane z serią granic tnących przestrzeń tych atrybutów na obszary, w których należy podejmować takie, a nie inne decyzje. Ilustruje to rysunek 15, na którym po lewej stronie pokazana jest przestrzeń wyznaczona przez atrybuty x i y oraz przykłady punktów, w których powinna być podejmowana odpowiednia decyzja 1 lub 2, a po prawej stronie pokazane jest odpowiadające temu drzewo decyzyjne.



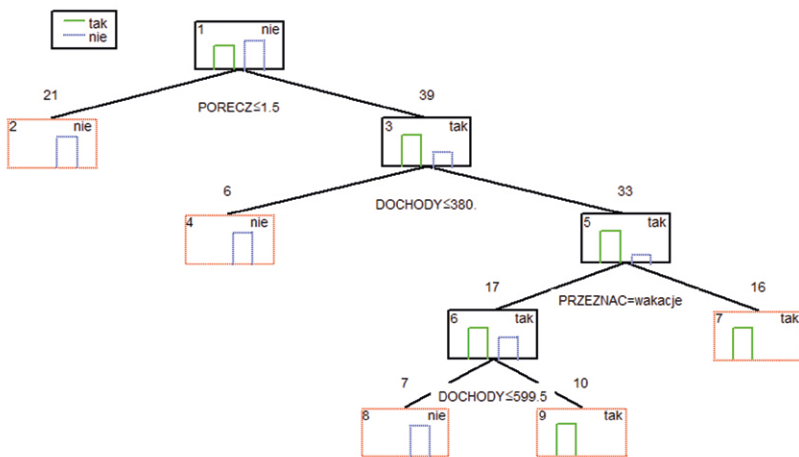
Rys. 15. Podział przestrzeni atrybutów na obszary różnych decyzji przy użyciu drzewa decyzyjnego

Jak wynika z tytułu tej części artykułu – drzewa decyzyjne mogą powstawać automatycznie w wyniku procesu uczenia. Proces ten przebiega w sposób następujący:

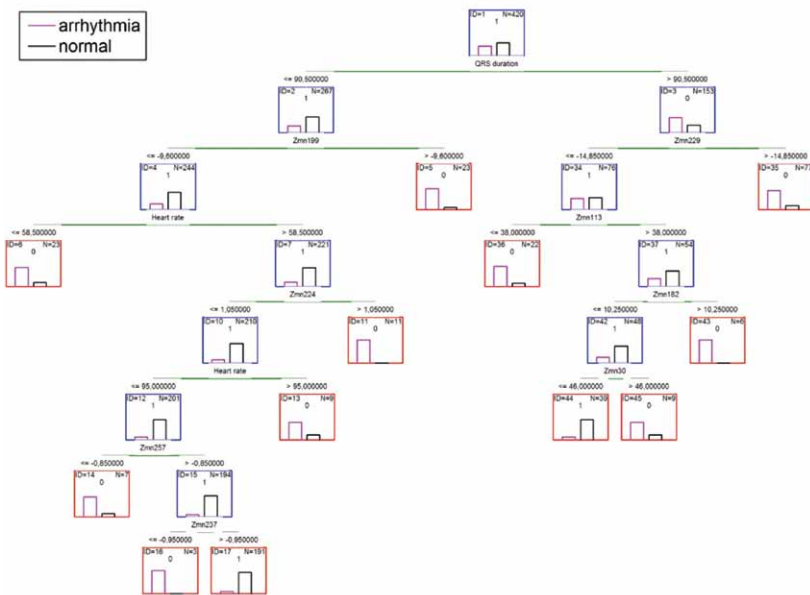
Najpierw gromadzi się dane na temat decyzji, jakie podejmowali ludzie. Przykład takiego zbioru danych pokazano na rysunku 16. Jest to oczywiście przykład wymyślony dla tego artykułu, ale tak mógłby wyglądać rejestr decyzji podjętych przez ekspertów banku, którzy pewnym ludziom przyznawali

lp.	DOCHODY	POREZC	PRZEZNAC	KREDYT	lp.	DOCHODY	POREZC	PRZEZNAC	KREDYT
1	140	1	sprzet	nie	31	500	1	sprzet	nie
2	140	1	wakacje	nie	32	500	1	wakacje	nie
3	180	2	sprzet	nie	33	500	2	sprzet	tak
4	220	2	wakacje	nie	34	500	2	sprzet	tak
5	220	3	sprzet	nie	35	500	2	sprzet	tak
6	280	2	wakacje	nie	36	500	2	wakacje	nie
7	350	1	sprzet	nie	37	520	2	sprzet	tak
8	350	1	sprzet	nie	38	520	2	sprzet	tak
9	360	1	sprzet	nie	39	530	2	sprzet	tak
10	360	1	wakacje	nie	40	550	2	wakacje	nie
11	360	2	sprzet	nie	41	590	2	wakacje	nie
12	360	2	wakacje	nie	42	599	2	wakacje	nie
13	400	2	sprzet	tak	43	600	2	sprzet	tak
14	400	2	wakacje	nie	44	600	2	wakacje	tak
15	410	2	sprzet	tak	45	600	2	wakacje	tak
16	410	2	sprzet	tak	46	600	2	wakacje	tak
17	410	2	sprzet	tak	47	610	1	wakacje	nie
18	410	2	wakacje	nie	48	610	2	wakacje	tak
19	420	0	sprzet	nie	49	620	2	wakacje	tak
20	420	1	sprzet	nie	50	630	0	sprzet	nie
21	420	1	wakacje	nie	51	630	0	wakacje	nie
22	420	2	sprzet	tak	52	650	1	sprzet	nie
23	440	2	sprzet	tak	53	650	2	wakacje	tak
24	440	2	wakacje	nie	54	680	2	wakacje	tak
25	450	1	sprzet	nie	55	700	1	wakacje	nie
26	450	1	wakacje	nie	56	700	2	sprzet	tak
27	450	2	sprzet	tak	57	700	2	wakacje	tak
28	480	2	sprzet	tak	58	700	2	wakacje	tak
29	500	0	wakacje	nie	59	720	0	wakacje	nie
30	500	1	sprzet	nie	60	850	2	wakacje	tak

Rys. 16. Przykład zbioru danych, na podstawie którego można nauczyć drzewo decyzyjne



Rys. 17. Sposób automatycznego uczenia drzewa decyzyjnego. Omówienie w tekście. Oglądając rysunek, należy pamiętać, że w każdym węźle odpowiedź „tak” kieruje w lewo, a odpowiedź „nie” – w prawo



Rys. 18. Uczące się drzewo decyzyjne do rozpoznawania arytmii (w medycynie)

kredyty – a innym nie. Atrybutami, które brali pod uwagę, były: **dochody** (kwoty w tabeli dotyczą tygodniowych dochodów w funtach brytyjskich), liczba **poręczycieli** zgłoszonych przez osobę ubiegającą się o kredyt oraz **przeznaczenie** pożyczanych pieniędzy (brane pod uwagę były tylko **wakacje** albo zakup **sprzętu**). Wiadomo było też, czy danej osobie kredyt przyznano, czy też nie.

Proces uczenia drzewa decyzyjnego pokazany jest na rysunku 17. Korzeniem

drzewa (okienko z numerem 1) jest atrybut mówiący o liczbie poręczycieli. Z analizy tabeli (rys. 16) wynikało, że jeśli liczba poręczycieli była mniejsza lub równa 1,5, wówczas proces podejmowania decyzji kończył się negatywnie, czyli na podstawie tego warunku można dojść do liścia o numerze 2, gdzie końcowa decyzja jest negatywna. Takich przypadków w tabeli było 21 – i taką liczbą opatrzona jest gałąź wiodąca od węzła 1 do węzła 2.

Negatywna odpowiedź na pytanie zadane w węźle 1 prowadzi do węzła nr 3. Takich przypadków w tabeli było 39. W węźle nr 3 badane jest, czy dochody kredytobiorcy są mniejsze lub równe 380. To odpowiada w tabeli granicy pomiędzy przypadkami od 1 do 12 (w których podjęto decyzję negatywną), a przypadkiem 13, który jako pierwszy otrzymał od ekspertów decyzję pozytywną. W związku z tym odpowiedź „tak” na pytanie w węźle 3 kieruje do liścia nr 4, który też oznacza decyzję negatywną. W ten sposób zostało „załatwionych odmownie” przez ekspertów następnych 6 amatorów – i drzewo decyzyjne skwapliwie się tego uczy.

33 klientów, których dochody przekroczyły próg 380 funtów tygodniowo, trafia do węzła nr 5, gdzie badane jest przeznaczenie kredytu. Jeśli klient deklaruje, że zamierza użyć kredytu do sfinansowania wakacji (a w przykładowej tabeli było takich marzycieli 17) – to jest kierowany do węzła nr 6, gdzie są ponownie sprawdzane jego dochody. Jeśli nie osiągają one pułapu 600 funtów tygodniowo – to klient trafia do węzła (liścia) nr 8, gdzie też czeka go końcowa decyzja odmowna. W tabeli opisującej decyzje ekspertów było takich pechowców 7. Jeśli natomiast odpowiedź na pytanie w węźle 6 była negatywna – to klienci trafiają o węzła nr 9, gdzie nareszcie są przydzielane decyzje pozytywne. Takich szczęściarzy jest 10. Liść z pozytywną decyzją znajduje się też w węźle nr 7, do którego trafiają ci średnio zamożni klienci, którzy jednak nie poślakomili się na wakacje (negatywna odpowiedź w węźle 5). Jest takich klientów 16.

Warto to wszystko prześledzić na rysunku 17.


Przedstawiony wyżej krok po kroku sposób budowy uczącego się drzewa decyzyjnego był możliwy do realizacji „ręcznie”, bo drzewo, które budowaliśmy, składało się z zaledwie 4 węzłów, w których były umieszczone określone warunki, oraz miało zaledwie 5 liści. W rzeczywistych zastosowaniach używane są drzewa, w których jest znacznie więcej węzłów – patrz na przykład rys. 18.

Budowa i uczenie takich drzew decyzyjnych oparte są (najczęściej) na programach ID3 i C4.5, dostępnych w Internecie. Ale sięgając po te programy, które przy dobrze zebranych zbiorze przykładów wykonają całą pracę automatycznie, warto jednak wiedzieć, co i jak one robią – i temu właśnie poświęcony był ten artykuł.

Literatura

- [1] TADEUSIEWICZ R.: *Uroki gier [Charms of computer games]* (felieton z serii „W laboratorium i w praktyce – ciekawostki naukowo-techniczne”) „Dziennik Polski” nr 161 (19475) 11.07.2008, PDF: https://www.academia.edu/34362713/Uroki_gier.
- [2] Tadeusiewicz R.: *Komputery grające w ludzkie gry [Computers playing human games]* (felieton z serii „W laboratorium i w praktyce”) „Dziennik Polski” nr 131 (19749) 05.06.2009, PDF: https://www.academia.edu/34921268/Komputery_graj%C4%85ce_w_ludzkie_gry; <http://www.dziennikpolski24.pl/artukul/2603946,komputery-grajace-w-ludzkie-gry,id,t.html>.
- [3] TADEUSIEWICZ R., MIKRUT Z.: *Neural networks for image recognition* [In Polish: *Sieci neuronowe rozpoznające obrazy*]. Referat plenarny – I Krajowa Konferencja Sieci Neuronowe i Ich Zastosowania, Kule, 12–15.04.1994, PDF: https://www.academia.edu/37437194/Sieci_neuronowe_rozpoznaj%C4%85ce_obrazy.

- [4] GAJECKI L., TADEUSIEWICZ R.: *Language Modeling and Large Vocabulary Continuous Speech Recognition*. „Journal of Applied Computer Science”, vol. 17, no. 2 (2009), PDF: https://www.academia.edu/37020601/Language_Modeling_and_Large_Vocabulary_Continuous_Speech_Recognition.
- [5] DEMENKO G., TADEUSIEWICZ R.: *Technologies of computer speech analysis and their application in police activity* [In Polish: *Technologie komputerowego przetwarzania mowy i ich możliwe zastosowania w pracy policji*]. Rozdział w książce: SZYMANIAK A., CIEPIELA W.: *Policja w Polsce – stan obecny i perspektywy. Tom I*. Wydawnictwo Naukowe Instytutu Nauk Politycznych i Dziennikarstwa UAM, Poznań 2007, PDF: https://www.academia.edu/37183918/Technologie_komputerowego_przetwarzania_mowy_i_ich_mo%C5%BCliwe_zastosowania_w_pracy_policji
- [6] TADEUSIEWICZ R.: *Maszyna cyfrowa – partner do pogawędki*. [Computer – Partner to a Chat]. „Młody Technik” 3/1977, PDF: https://www.academia.edu/34703955/Maszyna_cyfrowa_-_partner_do_pogaw%C4%99dki.
- [7] TADEUSIEWICZ R.: *Mowa i maszyny*. [Speech and Machines]. „Młody Technik” 1/1977, PDF: https://www.academia.edu/34703498/Mowa_i_maszyny.
- [8] MAZOŃ S., TADEUSIEWICZ R.: *Man-computer natural language conversation* [In Polish: *Konwersacja człowieka z maszyną cyfrową w języku naturalnym*]. „Informatyka” 9/1978, PDF: https://www.academia.edu/28247444/Konwersacja_cz%C5%82owieka_z_maszyn%C4%85_cyfrow%C4%85_w_j%C4%99zyku_naturalnym.
- [9] MAZOŃ S., TADEUSIEWICZ R.: *An attempt to description of some class of natural man – computer communication*. [In Polish: *Próba opisu pewnej klasy systemów naturalnej konwersacji człowieka z maszyną cyfrową*]. Archiwum Automatyki i Telemechaniki, Tom XXIV, nr 2, 1979, PDF: https://www.academia.edu/36357748/Pr%C3%B3ba_opisu_pewnej_klasy_system%C3%B3w_naturalnej_konwersacji_cz%C5%82owieka_z_maszyn%C4%85_cyfrow%C4%85_w_j%C4%99zyku_naturalnym; https://www.researchgate.net/publication/265546583_An_attempt_at_describing_a_class_of_man-machine_dialogue_systems_based_on_a_natural_language.
- [10] TADEUSIEWICZ R.: *Natural language and computers* [In Polish: *Język naturalny i komputery*]. Rozdział w pracy zbiorowej LUBASZEWSKI W. (RED.): *Słowniki komputerowe i automatyczna ekstrakcja informacji z tekstu*. UWND, Kraków 2009, PDF: https://www.academia.edu/39535742/J%C4%99zyk_naturalny_i_komputery
- [11] TADEUSIEWICZ R.: *Uczące się drzewa decyzyjne [Learning decision trees]* (felieton z serii „W laboratorium i w praktyce – ciekawostki naukowo-techniczne”) „Dziennik Polski”, nr 208 (19522) 05.09.2008, https://www.academia.edu/34367315/Ucz%C4%85ce_si%C4%99_drzewa_decyzyjne.
- [12] TADEUSIEWICZ R., DUDEK-DYDUCH E.: *Construction of Branch & Bound Decision Tree in Optimization of Discrete Manufacturing Processes*, in: *Computer Science in Management*, Tempus S-JEP-09139-95, 1998, https://www.academia.edu/36360110/Construction_of_Branch_and_Bound_Decision_Tree_in_Optimization_of_Discrete_Manufacturing_Processes.

 prof. dr hab. inż. Ryszard Tadeusiewicz
AGH – Akademia Górniczo-Hutnicza w Krakowie
e-mail: rtad@agh.edu.pl, www.Tadeusiewicz.pl