

Sztuczna Inteligencja

Reprezentacja wiedzy IV

Systemy produkcyjne, ramy, skrypty, mapy argumentów

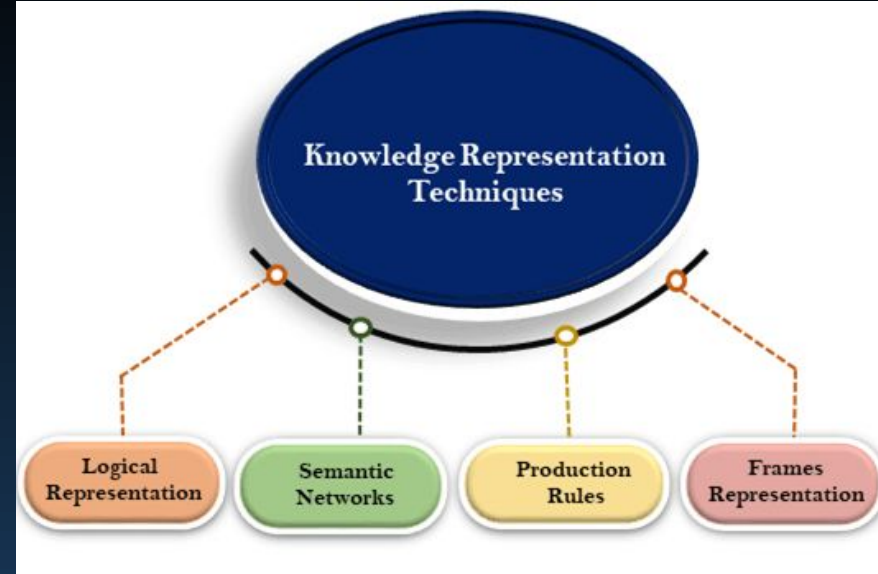
Włodzisław Duch

Katedra Informatyki Stosowanej UMK

Google: Włodzisław Duch

Co będzie

- Systemy produkcyjne
- Sieci semantyczne
- Reprezentacje bezpośrednie
- Ramy
- Skrypty
- Mapy argumentów



Zrobimy przegląd klasycznych metod reprezentacji wiedzy i systemów ekspertowych, które są na nich oparte, w tym dużych systemów, takich jak CyC, Watson, SOAR.

Potem zajmiemy się językiem i dużymi modelami językowymi, na których oparta jest generatywna AI.

Systemy produkcyjne

Newell i Simon (1972) – celem AI jest modelowanie sposobu działania percepcji i aparatu poznawczego człowieka. Do czasu powstania AI opartej na wielkich modelach językowych (LLM, Large Language Models) to była najlepsza strategia.

„System produkcyjny”: system oparty na „regułach produkcji”, czyli parach (warunek - działanie).

Reguły produkcji mają postać:

IF warunek1 AND warunek2 ... THEN akcja

Lewa część reguły produkcji określa warunki jej stosowalności.

Prawa część określa jej działanie.

Kilka reguł produkcji => łatwo przewidzieć skutki ich wykonywania.

Wiele reguł => system jest sterowany przez dane, a nie instrukcje!

Własności

Warunki wykonania akcji są jawne

Reguły nie odwołują się do siebie (tak jak procedury), minimalizując oddziaływanie pomiędzy sobą.

Da się zastosować w stosunkowo dużych systemach.

Działa dobrze do czasu aż liczba reguł nie robi się zbyt duża.

Można je też traktować jako mechanizm kontrolujący sprzężenie pomiędzy wiedzą deklaratywną i proceduralną.

Systemy produkcyjne składają się z 3 części:

- a) bazy reguł produkcji
- b) specjalnej struktury danych, nazywanej kontekstem
- c) interpretera, kontrolującego aktywność systemu

Własności cd

Setki reguł => trudno jest przewidzieć działanie systemu.

Kontekst, czyli dane na których działa system, zmienia się na skutek stosowania reguł produkcji.

Kontekst zawiera również dane, do których stosuje się lewa część reguł produkcji, określająca warunki ich stosowalności.

Kontekst może być:

- prostą listą faktów, określająca stan systemu,
- złożoną strukturą danych.

Interpreter określa strategię działania systemu - którą regułę należy następnie zastosować.

Przykład

System produkcyjny identyfikujący produkty spożywcze.

Mamy taśmę, a na niej różne produkty.

Kontekst = listy, na której są wprowadzone automatycznie do programu własności produktów.

F. logiczna JEST \Leftrightarrow jest na liście danych kontekstowych.

F. logiczna DOPISZ = operacja dopisywania do listy.

P1. IF JEST zielone THEN DOPISZ produkt rolny

P2. IF JEST mały pojemnik THEN DOPISZ delikatesowe

P3. IF JEST mrożonka OR JEST produkt rolny THEN DOPISZ nietrwałe

P4. IF JEST ciężkie AND JEST tanie AND NOT nietrwałe
THEN DOPISZ arbuz

P5. IF JEST nietrwałe AND JEST ciężkie THEN DOPISZ indyk

P6. IF JEST ciężkie AND JEST produkt rolny THEN DOPISZ arbuz



Interpreter

INTERPRETER:

I1. Znajdź wszystkie reguły których lewa strona jest spełniona.

I2. Jeśli daje się zastosować więcej niż jedna reguła usuń te, które prowadzą do powtarzających się symboli.

I3. Wykonaj po kolei dające się zastosować reguły. Jeśli nie można zastosować żadnej reguły przerwij działanie.

I4. Usuń listę dających się zastosować reguł i przejdź do I1.

Reguły produkcji wywoływane są cyklicznie.

W każdym cyklu ustala się możliwe do zastosowania reguły (faza dopasowania reguł do warunków, matching phase).

Jeśli do tych samych założeń stosuje się więcej reguł wybierana jest jedna z nich (faza rozstrzygnięcia sprzeczności, conflict resolution).

Ostatni etap to wykonanie działania (faza działania, action phase).

Działanie

Zaczynając np. od listy L

$L = (\text{zielone, ciężkie})$

możemy zastosować tylko regułę P1 otrzymując

$L = (\text{produkt rolny, zielone, ciężkie})$

W następnym kroku możemy zastosować P1 i P6;

P1 daje powtarzające się symbole,

więc stosujemy P6, otrzymując

$L = (\text{arbuz, produkt rolny, zielone, ciężkie})$

System zatrzymuje się, gdyż wszystkie stosowalne w kolejnym cyklu reguły prowadzą do takich samych symboli, które już są na liście.

Zalety i wady

The logo features the text "AI IN FUTURE PRODUCTION SYSTEMS" in a bold, white, sans-serif font, stacked in three lines. The background is a dark blue with a subtle pattern of light blue particles and a faint grid, suggesting a digital or futuristic theme.

Zalety systemów produkcyjnych:

- Modularność wiedzy: niezależne reguły.
- Jednolitość sposobu reprezentacji.
- Naturalność zapisu w wielu zastosowaniach.

Wady: trudno zapisać sekwencyjny algorytm postępowania, gdyż reguły nie odwołują się bezpośrednio do siebie.

Przykłady zastosowań: systemy eksperckie

Reprezentacje bezpośrednie.
Ramy, skrypty, mapy argumentów.

Reprezentacje bezpośrednie

Reprezentacje bezpośrednie to takie w których relacje i własności części odpowiadają relacjom i własnościom części rzeczywistego (modelowanego) systemu.

Reprezentacje mogą być bezpośrednie pod pewnym względem, a symboliczne pod innym, np. mapa.

Reprezentacje geometryczne.

Reprezentacja symboliczna dopuszcza wiele modeli rzeczywistych; reprezentacje bezpośrednie znacznie mniej.

Przestrzeń koncepcji – abstrakcyjna p. geometryczna, w której możemy próbować utworzyć reprezentacje bezpośrednie podobieństwa obiektów (między sobą lub do prawdziwych).

Word2Vec i inne algorytmy pozwalają na tworzenie abstrakcyjnych reprezentacji, będących aproksymacją reprezentacji bezpośrednich.

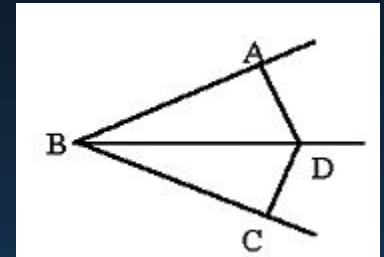
Przykład: Geometra

Przykład: **GEOMETRA** (Gelernter 1963)

Dane: Kąt $ABD = DBC$

AD prostopadłe do AB

DC prostopadłe do BC



Dowiedź: Odcinek $AD = CD$

Transformacje, które dają się zastosować nie zepsują diagramów.

General Space Planner (Eastman 1973) - planowanie przestrzenne

WHISPER (Funt 1977) - zdarzenia w świecie klocków

Ramy

Wprowadzone przez Marviną Minsky'ego w 1975 r.

Operatory/transformacje indukują zmiany, ale większość obiektów się nie zmienia i wiedza na ich temat może być ujęta w „ramy”.

Ramy to złożone struktury powstałe w wyniku nagromadzenia się wcześniejszych doświadczeń; ich zadaniem jest opisać obiekt lub koncepcje i możliwości jej użycia.

Ramy mają „gniazda” lub „haczyki” (slots, hooks) na fakty lub procedury.

Rozumowanie polega na zapełnianie tych gniazd, doczepianiu faktów.

Wiedza deklaratywna i proceduralna może zostać umieszczona w dobrze zdefiniowanych ramach. Łatwo dodać nowe gniazda, więc możliwości ekspresji wiedzy za pomocą ram są duże.

Ramy mogą być hierarchicznie zagnieżdżone, tworząc sieć semantyczną.

Wady: budowa ram dla wielu pojęć jest trudna, bazy danych oparte o ramy są znacznie mniejsze niż bazy leksykalne Wordnetu.

Ramy – prosty przykład

Ogólna ramka PIES

Co to: ZWIERZĘ; ZWIERZĄTKO DOMOWE

Rasa: ?

Właściciel: OSOBA

(jeśli-potrzebna: znajdź OSOBA z ZWIERZĄTKO DOMOWE =ja)

Imię: NAZWA WŁASNA (DEFAULT = Reks)

....

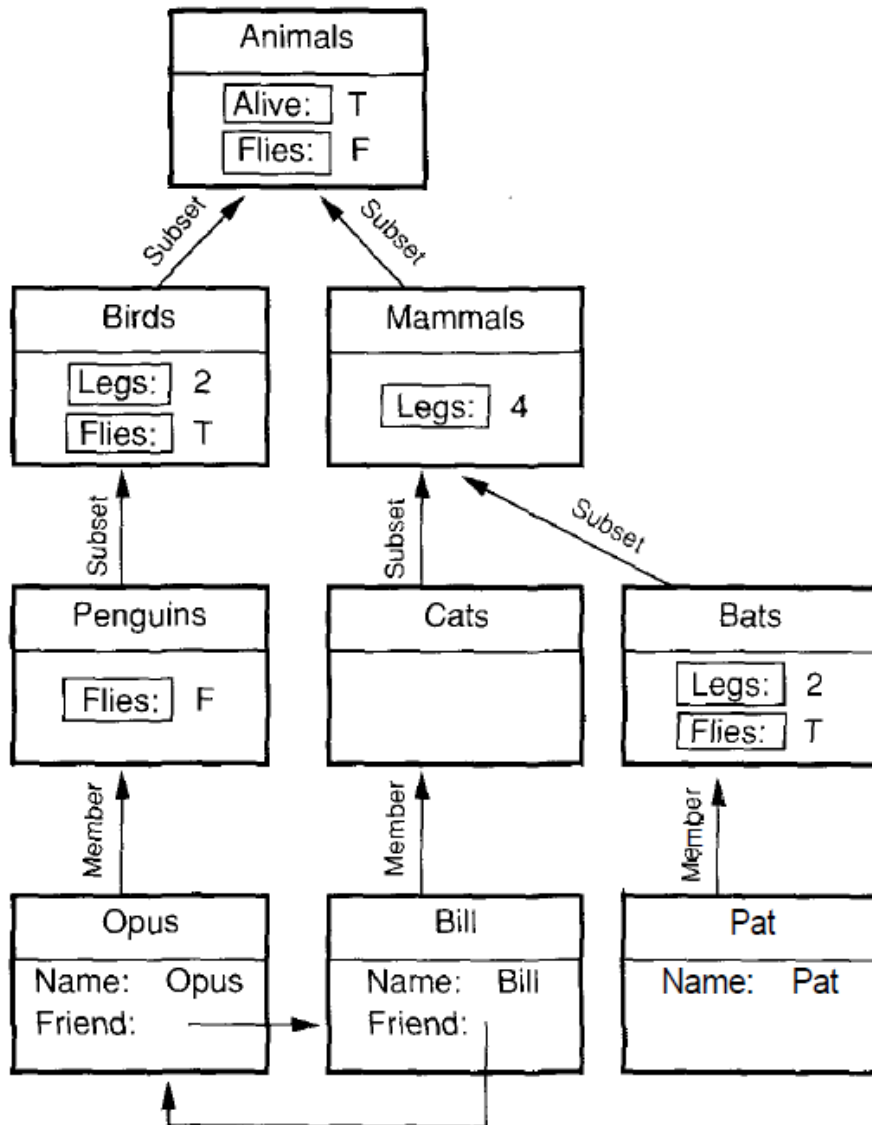
Ramka PIES-SAŚIADA

Co to: PIES

Rasa: kundel

Właściciel: Józek

Imię: Flejtuch



Rel(Alive,Animals,T)
 Rel(Flies,Animals,F)

Birds C Animals
 Mammals C Animals

Rel(Flies,Birds,T)
 Rel(Legs,Birds,2)
 Rel(Legs,Mammals,4)

Penguins C Birds
 Cats C Mammals
 Bats C Mammals
 Rel(Flies,Penguins,F)
 Rel(Legs,Bats,2)
 Rel(Flies,Bats,T)

Opus G Penguins
 Bill ∈ Cats
 Pat ∈ Bats
 Name(Opus,"Opus")
 Name(Bill,"Bill")
 Friend(Opus,Bill)
 Friend(Bill,Opus)
 Name(Pat,"Pat")

(a) A frame-based knowledge base

(b) Translation into first-order logic

Hierarchia ram (Russel & Norvig, AI. A modern approach.)

Ramy - przykład

Ramka Pokój hotelowy

Kategoria ogólna: Pokój

Miejsce: Hotel

Zawiera: krzesło hotelowe,
telefon hotelowy
łóżko hotelowe
.....

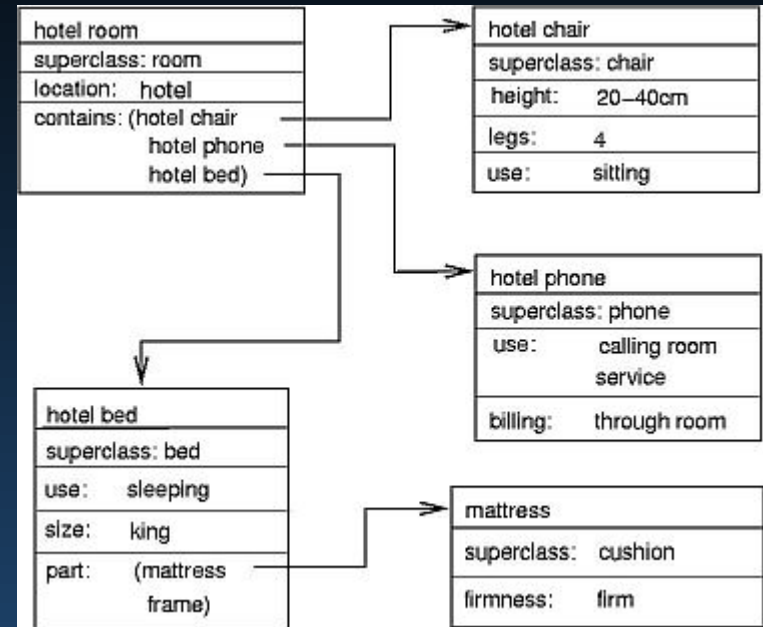
Ramka łóżko hotelowe

Kategoria ogólna: łóżko

Funkcja: Spanie

Rozmiar: king size, queen size, standard

Części: rama, materac ...



Własności

Którą ramę zastosować do danej sytuacji?

Problem „kruchości” reprezentacji za pomocą ram.

Ramy w sieciach semantycznych mogą tego uniknąć.

Dobór strategii rozumowania przez dobór ram.

Przykłady:

Bobrow i Winograd (1977) wprowadzili oparty na ramach język KRL (Knowledge Representation Language).

Program **GUS** (Bobrow 1977) do analizy języka naturalnego używał ram.

FrameNet to projekt International Computer Science Institute w Berkeley.

SEMAFOR używa FrameNet do analizy semantycznej zdań (CMU).

Anotacja tekstu za pomocą FrameNet – **Github tools** w Pythonie.

Sekwencja wydarzeń rozwijających się w czasie – do zrozumienia konieczny jest skrypt (Schank i Abelson 1977).

Skrypty

JEDZENIE-W-RESTAURACJI (skrypt)

Obiekty: (restauracja, pieniądze, jedzenie, menu, stoliki, krzesła)

Role: (klienci, kelnerzy, kucharze)

Punkt-widzenia: klient

Czas-zdarzenia (godziny otwarcia restauracji)

Miejsce-zdarzenia (położenie restauracji)

Sekwencje zdarzeń:

najpierw: Wchodzimy do restauracji (skrypt)

potem if (znak rezerwacja lub prosimy-czekać-na-miejsce)

then (zwrócić-uwagę-kelnera skrypt)

potem Prosimy-usiąść skrypt

potem Zamawiamy-jedzenie skrypt

Skrypty cd

potem Spożywamy-jedzenie skrypt unless (długie-czekanie) when
Wychodzimy-zdenerwowani-z-restauracji skrypt

potem if (jedzenie-bardzo-dobre)
then Gratulacje-dla-kucharza skrypt

potem Płacimy-za-jedzenie skrypt

koniec:

Opuszczamy-restaurację skrypt

Skrypty umożliwiają analizę typowych historyjek, np. wycinków z gazet opisujących przejmowanie banków, fuzje firm itp.

Dzięki temu można łatwo dopasować formę pytań i odpowiedzi do tekstu

Mapy argumentów

Stara technika, opisana w

Richard Whately's *Elements of Logic*. London 1867.

Po ponad 100 latach Robert Horn (1998) zaczął ją używać na większą skalę do wspomaganie rozumowania.

Podjęto próby standaryzacji: Argument Interchange Format, AIF oraz stworzenia ontologii AIF-RDF. Postał też standard dla informacji prawniczej, Legal Knowledge Interchange Format.

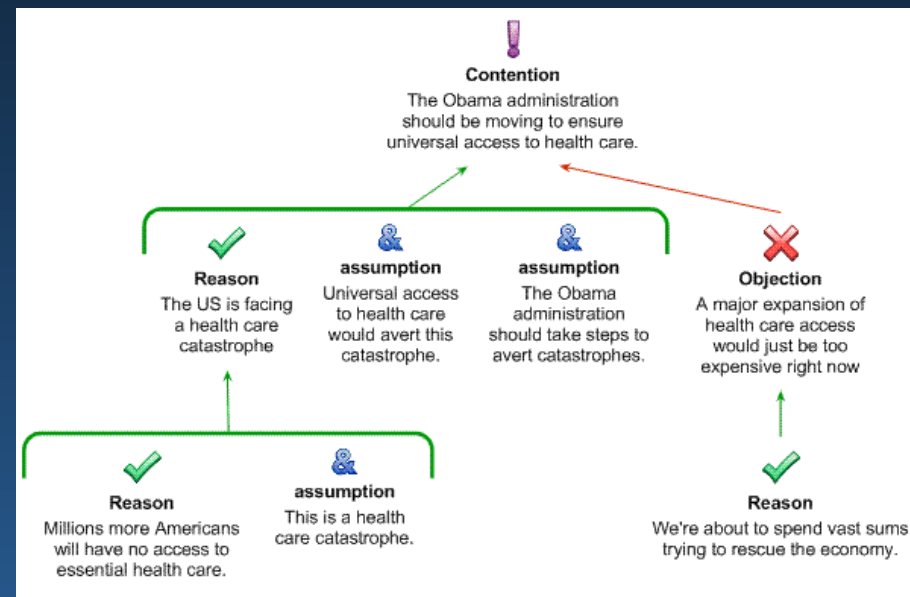
Dokładniejsze informacje:

[Argument map Wiki](#)

Jest sporo programów wspomagających rozumowanie przez tworzenie graficznych map, np.

[Rationale in ReasoningLab](#) software.

[Argunet](#)



Reprezentacja wektorowa

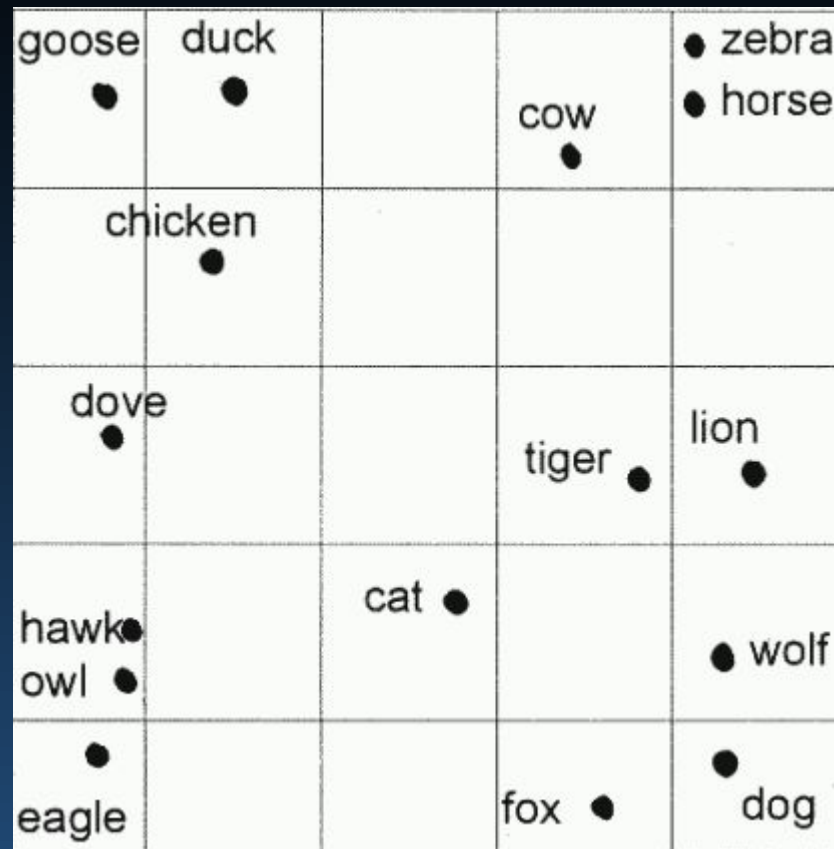
| | | d | o | g | o | h | e | w | t | h | z | | | | | | |
|----------|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| animal | | o | h | u | o | a | g | f | d | o | c | g | i | r | b | e | |
| | | v | e | c | s | w | w | l | o | o | l | a | e | o | s | r | o |
| | | e | n | k | e | l | k | e | x | g | f | t | r | n | e | a | w |
| is | small | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | medium | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | big | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| has | 2 legs | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 4 legs | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | hair | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | hooves | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | mane | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| | feathers | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| likes to | hunt | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| | run | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| | fly | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | swim | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Podobieństwo wektorów można ocenić za pomocą miary cosinusowej,
 $\cos(V_1, V_2) = \frac{V_1 \cdot V_2}{|V_1| |V_2|} = \text{il. składowy } V_1 \cdot V_2 \text{ dzielony przez il. norm } V.$

Odległość Hamminga = liczba różnych bitów pary wektorów, np.

$H(\text{dove}, \text{hen}) = |V(\text{dove}) - V(\text{hen})| = 1$, lub $\cos(V_1, V_2) = \frac{3}{\sqrt{4 * 3}} = \frac{\sqrt{3}}{2}$

Mapy semantyczne: MDS



Jeśli przedstawić wektory jako punkty X_i na płaszczyźnie tak by ich odległości $\|X_1 - X_2\|$ były maksymalnie podobne do $\|W(X_1) - W(X_2)\|$ (robi to algorytm dostaniemy taką mapę.

Widać naturalne grupowanie różnych gatunków.

Podobne własności \Leftrightarrow bliskie wektory, łatwe skojarzenia.

Reprezentacja wektorowa

Postało szereg technik przypisywania wektorów numerycznych słowom.

Word2Vec (Google, 2013) korzysta z dwóch technik:

- Continuous Bag of Words (CBOW) przewiduje brakujące słowa na podstawie otoczenia.
- Skip-gram przewiduje kontekst dla danego słowa.

Word2Vec i inne algorytmy tworzą reprezentacje o wymiarach 100-300, trenując płytkie sieci neuronowe na dużych korpusach tekstowych.

GloVe (Global Vectors for Word Representation, 2014), to konkurencyjny algorytm - [strona projektu \(Stanford\)](#)..

Macierze ko-okurencji obliczone na dużym korpusie pozwalają określić iloraz współwystępowania wyrazów, oraz wektory, których iloczyn skalarny jest równy logarytmowi tego ilorazu. Dzięki temu odległość wektorów staje się miarą ilorazu prawdopodobieństw, odzwierciedlającego semantykę słów. Dostępne są wektory wyliczone na Wikipedii o wymiarach 25-300.

FastText (Facebook, 2016) rozszerza Word2Vec dla słów traktowanych jako ciąg n-gramów, dodając informację o morfologii wyrazów.

To ułatwia analizę neologizmów (out-of-vocabulary, OOV).

Towards Data Science przedstawia kilka przykładów jak się to robi.

Intuitive Guide to Understanding Word2vec.

The Illustrated Word2vec