

Sztuczna Inteligencja

Język naturalny

Włodzisław Duch

Katedra Informatyki Stosowanej UMK

Google: Włodzisław Duch

Co było:



- Reprezentacja wiedzy
 - Systemy ekspertowe
 - Predykcyjne metody uczenia maszynowego
- Analiza języka w stylu GOFAI ma tylko historyczne znaczenie, jesteśmy w epoce dużych modeli językowych (LLM) i multimodalnych.
- Czym się zajmuje NLP
 - Języki formalne i ich gramatyki
 - Rozpoznawanie mowy
 - Przykłady starszych programów
 - Tłumaczenie maszynowe
 - Sieci neuronowe i algorytmy językowe.
 - Transformery
 - Inne techniki.

Natural Language Processing

- NLP to dział AI, w skład którego wchodzi:
- Teoria gramatyk i języków formalnych
- Szukanie dobrych reprezentacji wiedzy zawartej w tekstach
- Cele: wspomaganie prac z tekstami, tworzenie słowników, programów nauczających języki obce, dialog z komputerem w języku naturalnym (ludzkim), pisanym i mówionym;
- analiza sensu/gramatyki istniejących tekstów, streszczenia;
- komentowanie tekstu, wydobywanie informacji – szukarki!
- Tłumaczenie maszynowe, analiza i synteza mowy.
- Wymagania sprzętowe i programowe – wielkie, ale nie dla drugiej dekady XXI wieku!

Podstawowe pojęcia

Nauczyć się języka nie jest łatwo ludziom ani maszynom, ale postępy w ostatnich latach są ogromne.

Syntaktyka języka: zajmuje się formatem, regułami budowy i formalną składnią.

Gramatyka jest zbiorem reguł syntaktycznych.

Rozbiór gramatyczny (parsing) – uwidacznia nieliniowe powiązania części zdania, do pewnego stopnia widoczne w tradycyjnym rozborze (podmiot, orzeczenie).

Semantyka języka: znaczenie (sens) wyrażeń i konstrukcji w danym języku – zrozumiały tylko w kontekście, często szerokim (społecznym).
To najtrudniejsza część NLP.

Historia NLP

1950, Wczesne wysiłki: identyfikacji słów kluczowych, prymitywne programy do tłumaczenia maszynowego.

1957, N. Chomsky, Syntactic Structures, formalizacja języków.

1960, Usprawnienia: baza danych, indeksacja, dopasowanie do wzorców wypowiedzi, podstawowe analizy gramatyczne.

1970, Reprezentacja logiczna, analiza semantyczna, rola kontekstu, rozszerzone sieci semantyczne (ATM).

1975, Systemy wykorzystujące bazy wiedzy, wnioskujące, rozumujące.

1980, Programowanie logiczne, gramatyki ograniczeń, modele planowania do rozumienia i generacji tekstu.

1990, Formalizacja sieci semantycznych, modele statystyczne, metody koneksjonistyczne, integracja mowy i tekstu, obszerne językowe bazy danych, pragmatyka języka.

2000, Gramatyki probabilistyczne (data oriented parsing)

2010, Deep learning i sieci neuronowe z rekurencją.

- 2018, **Bidirectional Encoder Representations from Transformers (BERT)**
- 2022, Szeroki dostęp do Large Language Models: ChatGPT, Bing, Bard ...

Gramatyki i języki formalne

- Jak opisać język, proces analizy od dźwięków i grafemów do znaczenia słów i komunikacji?
- *Język formalny*: zbiór łańcuchów symboli utworzonych ze skończonego zbioru symboli końcowych
- Gramatyka:
 - 1. *Kategorie syntaktyczne*, takie jak <zdanie> i <faza>, czyli zmienne.
 - 2. *Symbole końcowe* lub terminalne języka, np. słowa języka naturalnego, łączone w zdania.
 - 3. Symbole nieterminalne, zmienne syntaktyczne, używane tymczasowo.
- Język $\Leftrightarrow$ zbiór wszystkich zdań akceptowalnych przez jakąś gramatykę. Zdania określone są przez:
 - 3. *Symbol początkowy* S , zdania generowane są z S (np. symbolu nieterminalnego).
 - 4. *Reguły przepisania lub produkcji*, relacje pomiędzy łańcuchami symboli; pozwala to tworzyć frazy.

Gramatyki - przykłady

Bardzo prosta gramatyka: $S \rightarrow NP VP$

$\langle ZDANIE \rangle \rightarrow \langle FRAZA Z RZECZOWNIKIEM \rangle NP$
 $\langle FRAZA Z CZASOWNIKIEM \rangle VP$

$\langle FRAZA Z RZECZOWNIKIEM \rangle \rightarrow \langle RZECZOWNIK \rangle$

$\langle RZECZOWNIK \rangle \rightarrow \text{pies}$

$\langle RZECZOWNIK \rangle \rightarrow \text{kot}$

$\langle FRAZA Z CZASOWNIKIEM \rangle \rightarrow \text{biegnie}$

Gramatyka wyrażeń logicznych - mamy symbole logiczne: $P, Q, \wedge$;
konstrukcja poprawna $P \wedge Q$,

konstrukcja niegramatyczna: $P Q \wedge$

Gramatyki formalne są przydatne do tworzenia języków programowania.
Trzeba zdefiniować reguły przypisania symboli.

Typy gramatyk (Chomsky 1957)

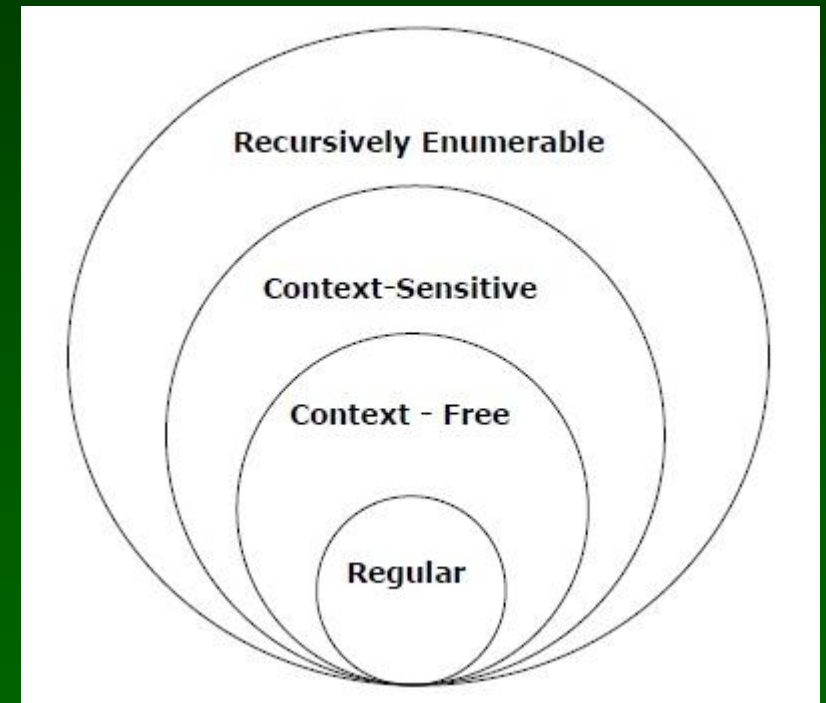
- 4 typy gramatyk formalnych, z różnymi ograniczeniami na reguły gramatyczne, od całkiem ogólnych do coraz bardziej ograniczonych, rozpoznawalne przez automaty skończone różnego typu.

Typ 0, Recursively Enumerable, to gramatyki rekurencyjnie przeliczalne, bez ograniczeń na postać reguł, generowane przez maszynę Turinga

Typ 1 Context-sensitive, gramatyki kontekstowo=wrażliwe.

Typ 2 Context-free grammar, gramatyki bezkontekstowe.

Typ 3 Regular grammar, gramatyki i języki regularne.



Typ 0

- **Typ 0**, recursively enumerable, czyli **rekurencyjnie przeliczalna**, to gramatyka najbardziej ogólna.
- Nie ma ograniczeń co do reguł przepisania, $X \rightarrow Y$ dla dowolnych ciągów X, Y , np. $S \rightarrow ACaB$.
- Generuje wszystkie możliwe języki formalne rozpoznawalne przez maszyny Turinga, obejmuje wszystkie typy gramatyk: regularne, kontekstowe i bezkontekstowe.
- Język rekurencyjnie przeliczalny jest rekurencyjnie przeliczalnym podzbiorem w zbiorze wszystkich możliwych słów w alfabecie języka.
- Język rekurencyjnie przeliczalny to język formalny, dla którego istnieje maszyna Turinga (lub inna obliczalna funkcja), która odróżnia wszystkie prawidłowe ciągi języka.
- Język rekurencyjnie przeliczalny to język formalny, dla którego istnieje maszyna Turinga (lub inna obliczalna funkcja), która zaakceptuje dowolny ciąg tego języka jako dane wejściowe.
- Maszyna Turinga może zatrzymać się, odrzucić lub zapętlić, jeśli przedstawić jej ciąg, który nie należy do języka rekurencyjnie przeliczalnego.

Typ 1

- **Typ 1, context-sensitive**, czyli **gramatyka kontekstowa**.
- Regułę przepisania $X \rightarrow Y$ można stosować tylko w danym kontekście.
- Reguły przepisania są typu $aXb \rightarrow aYb$, gdzie: X jest symbolem nieterminalnym, a, b, Y są dowolnymi ciągami symboli terminalnych lub nieterminalnych, Y jest ciągiem niepustym, a, b mogą być puste.
Np. $aXc \rightarrow aadc$, można zamienić $X \rightarrow ad$ tylko w kontekście $a .. c$.
- Gramatyka generuje język kontekstowy, rozpoznawalny przez automat liniowo ograniczony (jego pamięć jest funkcją liniową długości wejścia).
- Definiuje się też języki w oparciu o gramatyki lewo-kontekstowe i prawo-kontekstowe, ograniczając reguły do $aA \rightarrow ac$ albo $Ab \rightarrow cb$.
- Przykłady generacji języków kontekstowo-wrażliwych są tu.
- Słowo => wiele reprezentacji. Sieci neuronowe konstruuja reprezentacje słów w danym zdaniu w określonym kontekście. Model LSTM może utworzyć zanurzenie (embedding) kontekstowe.
- Embeddings from Language Model (ELMo): sekwencja słów => sekwencja wektorów. Z tego algorytmu rozwinęły się transformery.

Typ 2

Typ 2, gramatyka bezkontekstowa (context-free, CFG).

Skończony zbiór reguł przepisania $X \rightarrow Y$, gdzie X jest pojedynczym, niekończącym symbolem, a Y dowolnym zbiorem symboli terminalnych i nieterminalnych.

Języki bezkontekstowe odpowiadają niedeterministycznym automatom ze stosem, dają się przedstawić graficznie za pomocą struktur drzewiastych.

Np: gramatyka $(\{a,b\},\{S\},\{S \rightarrow aSb \mid \emptyset\})$ generuje język $a^n b^n$

Język palindromów nad alfabetem $\{a,b\}$ można wygenerować za pomocą gramatyki:

$(\{a,b\},\{S\},\{S \rightarrow aSa \mid bSb \mid a \mid b \mid \emptyset\},S)$

Np. $S \rightarrow aSa \rightarrow baSab \rightarrow abaSaba \rightarrow abaaaba \}$

Problem: jedno słowo może mieć tylko jedno znaczenie.

Typ 3, języki regularne

Typ 3, gramatyka regularna (liniowa) dopuszcza następujące reguły przypisania:

$X \rightarrow a$; lub $X \rightarrow a Y$ lub $X \rightarrow \emptyset$.

X, Y pojedyncze zmienne nieterminalne, a to pojedyncza zmienna końcowa.

Tworzy wyrażenia regularne, realizowane przez automaty skończone.

Wyrażenia regularne to zdania języków regularnych, pozwalają na identyfikację wzorców tekstu, są efektywne algorytmy by określić, czy szukany ciąg znaków pasuje do wzorca. Jest wiele narzędzi i bibliotek dla wszystkich używanych obecnie języków programowania. W praktyce wyrażenia regularne mogą nie być częścią języka regularnego ze względu na referencje wsteczne (np. w Perl).

Język typu p = język generowany przez gramatykę p .

Gramatyki typów wyższych to podzbiory typów niższych, języki też.

Typy 2 i 3 mają silne ograniczenia, ale 2 jest popularna w analizie języka naturalnego, daje spore możliwości.

Typ 3 nadaje się tylko do języków sztucznych, ale jest szybka i dlatego wykorzystywana w procesach szukania.

Analiza syntaktyczna

Zadanie: wydobyć powiązania elementów zdania, zrozumieć funkcję jaką w tym zdaniu pełnią. Używane gramatyki narzucają ograniczenia na rozbiór zdania.

Drzewo syntaktyczne tworzone jest przez parser korzystając z leksykonu. Dla języków naturalnych potrzebna jest przynajmniej gramatyka bezkontekstowa.

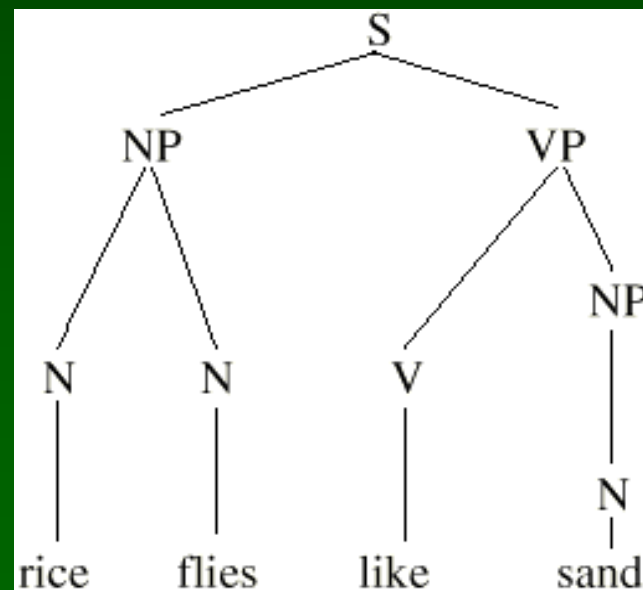
Leksykon zawiera informację o kategoriach syntaktycznych słów, gramatyka określa ich możliwe powiązania.

Drzewo dla zdania:

S = Rice flies like sand.

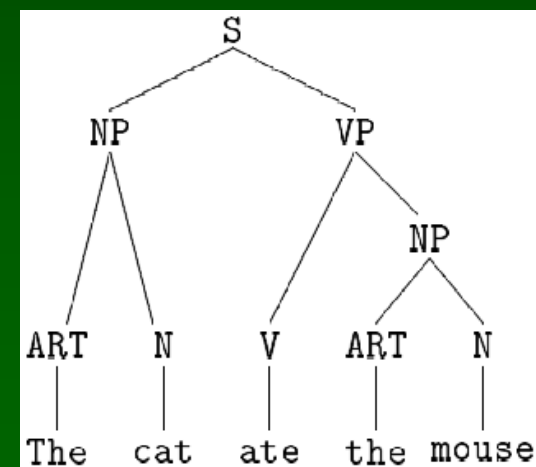
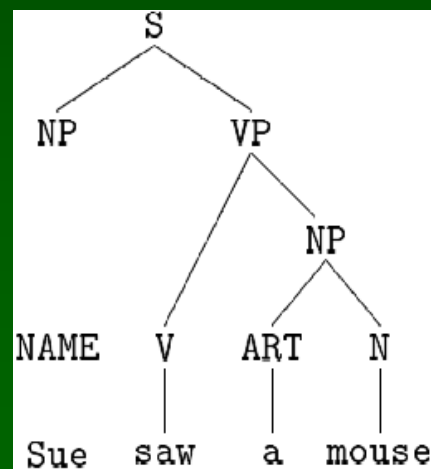
„Ryżowe muszki”, czy „ryż sypie się”?

Do tego potrzebna jest analiza kontekstu, nie wystarcza formalna analiza.



Przykład: gramatyka bezkontekstowa

Sentence	$S \rightarrow NP VP$
Verb Phrase	$VP \rightarrow V NP$
Noun Phrase	$NP \rightarrow NAME$ $NP \rightarrow ART N$
Article	$ART \rightarrow a \mid the$
Verb	$V \rightarrow ate \mid saw$
Noun	$N \rightarrow cat \mid mouse$
Name	$N \rightarrow Sue \mid Zak$



Inne rodzaje gramatyk

Jest wiele różnych rodzajów gramatyk, np:

Gramatyka generatywna – tworzy strukturalny opis zdania, pozwala rozpoznać i stworzyć nieskończoną liczbę zdań o prawidłowej strukturze.

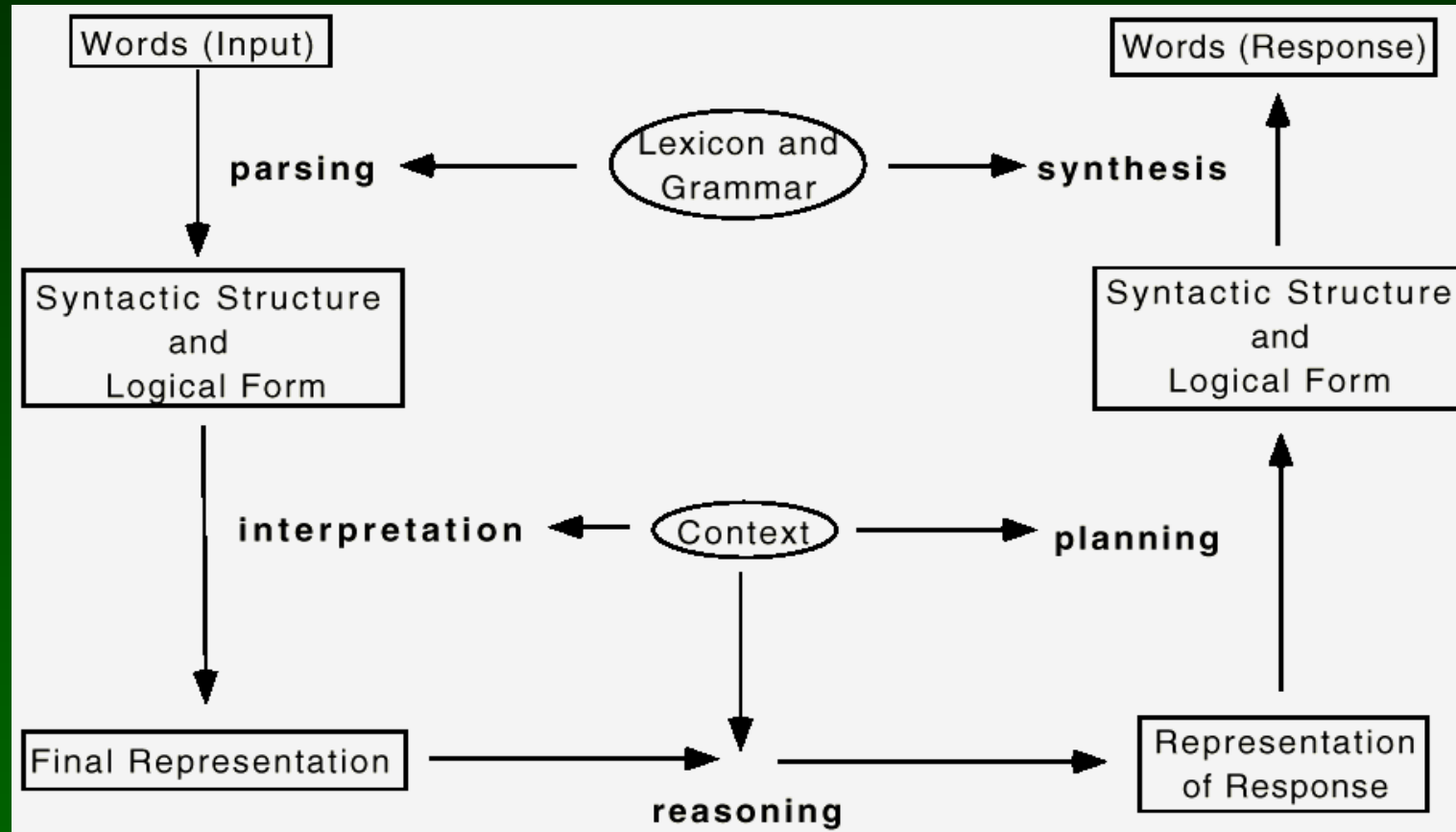
Gramatyka transformacyjna – głębokie znaczenie zdań jest niezmiennicze ze względu na transformacje „powierzchniowe” zdania.

Gramatyki systemiczne – część funkcjonalnej lingwistyki systemicznej, sieci systemowe, opisujące możliwości wyboru powiązanych znaczeń za pomocą „leksykogramów”, słów w powiązaniu z regułami.

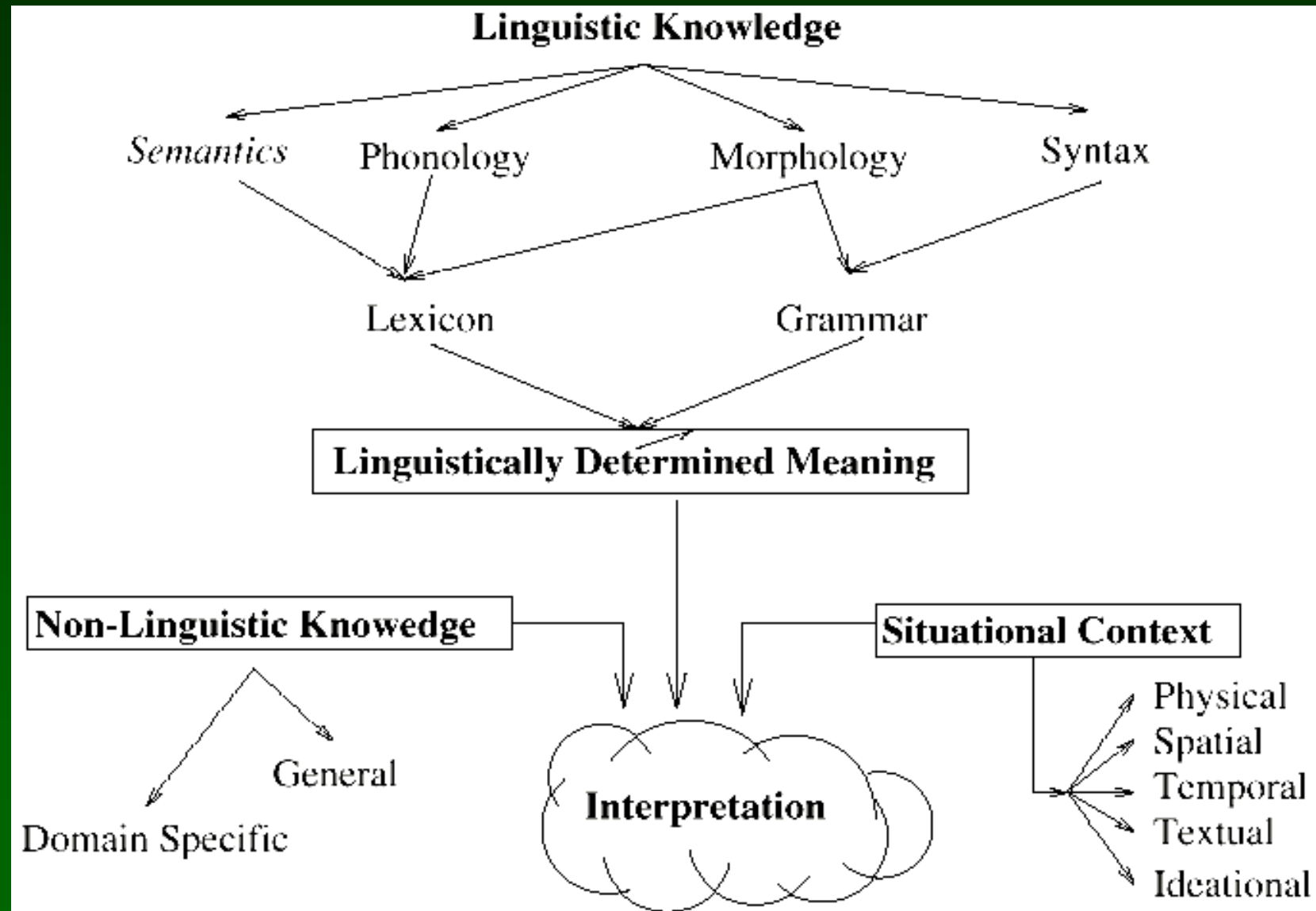
Gramatyki „przypadków” (case grammars), czyli „semantycznie istotnych powiązań syntaktycznych”; np. agent, obiekt, źródło, cel.

Parsery (analizatory składniowe), to programy do analiz struktur syntaktycznych. Różne strategie i rodzaje gramatyk w efekcie generują różne drzewa syntaktyczne.

Architektura systemów NLP



Język i interpretacja



5 kroków analizy

- **Fonologiczna** – fonetyka zajmuje się strukturą dźwięków; fonemy, formantami – istnieje międzynarodowy alfabet fonetyczny; fonologia zajmuje się organizacją fonemów w wypowiedziach.
- **Morfologiczna** – struktura słów, fleksja, rdzenie, leksykony wykorzystujące regularności języka na poziomie słów.
- **Syntaktyczna** – konstrukcja zdań, rozbiór gramatyczny.
- **Semantyczna** – sens, jednoznaczność, wybór słów, zależnie od kontekstu; markery semantyczne i organizacja leksykonu.
- **Pragmatyczna** – sposób użycia wyrazów, relacje zaimki-fraza; relacje między frazami, relacje przyczynowe, rozumienie całości, założenia.

Kroki analizy językowej.

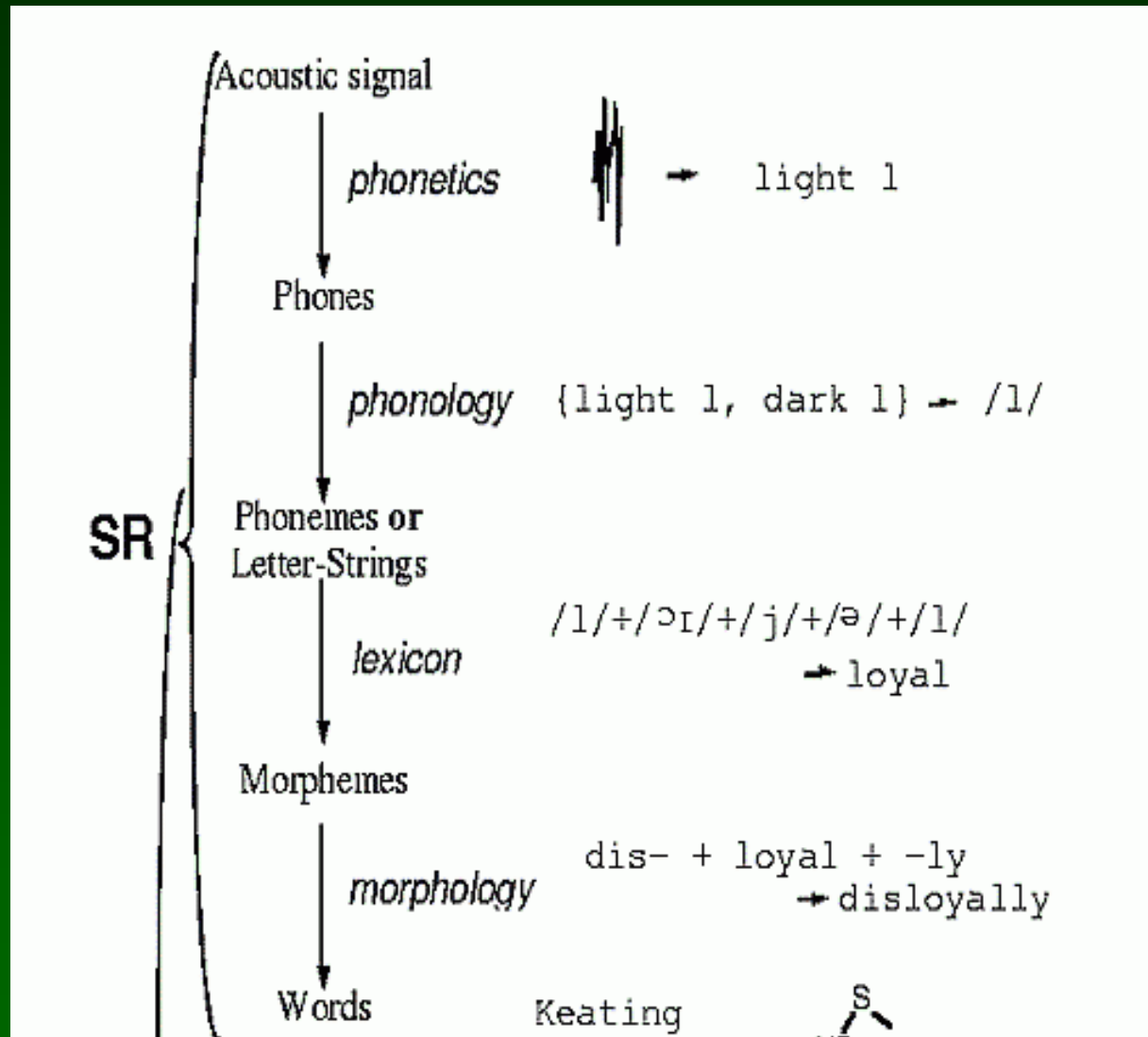
Speech recognition, czyli rozpoznawanie mowy, odbywa się tylko na poziomie fonetycznym + morfologicznym, ale korzysta ze słownika by wybrać bardziej prawdopodobną interpretację rozpoznawanych dźwięków.

Wymaga to dużych baz nagrań wypowiedzi różnych osób, i zaawansowanych algorytmów analizy sygnałów mowy.

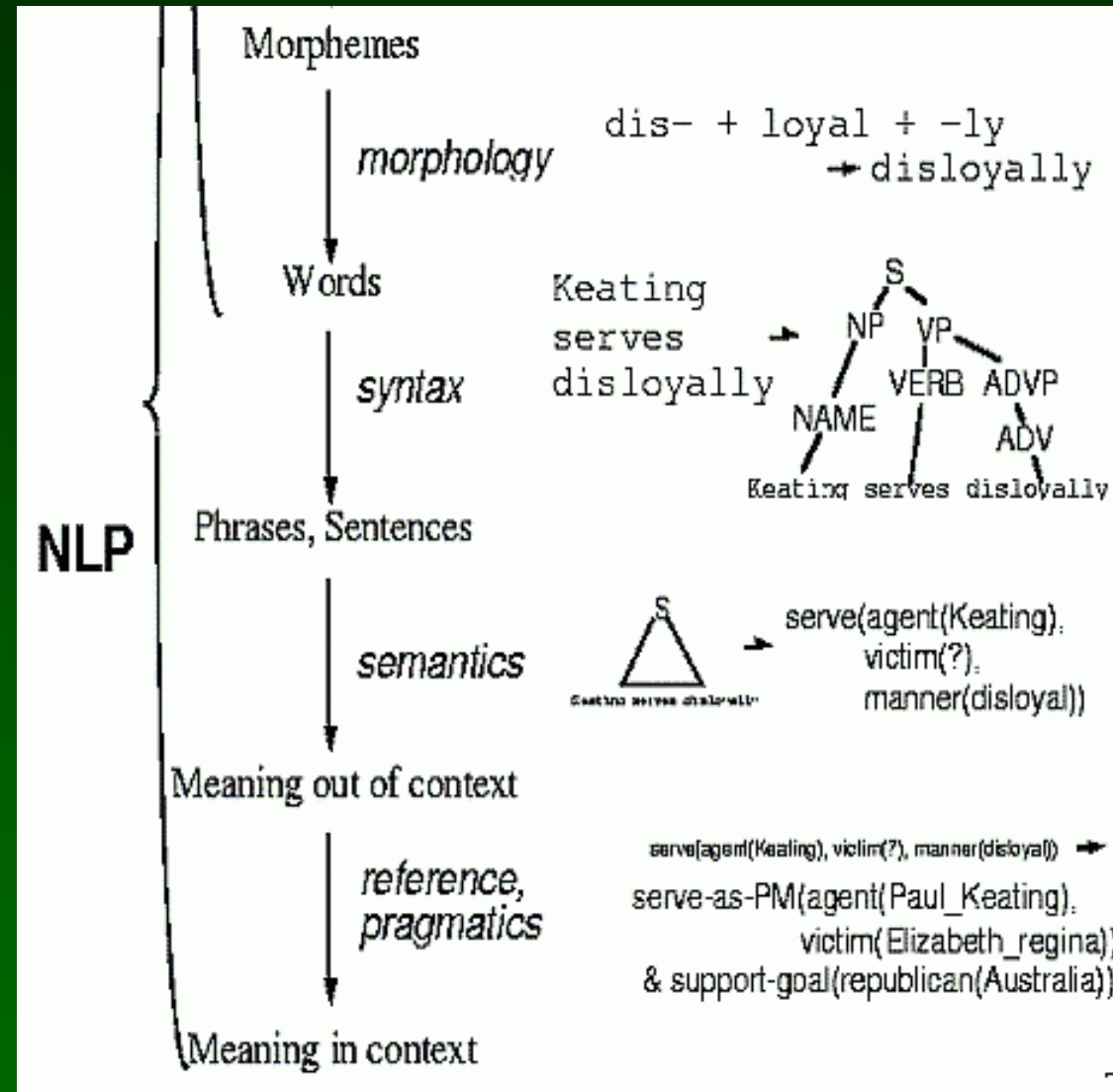
Natural Language Processing (NLP), czyli analiza języka naturalnego, zaczyna się od poziomu

- morfologicznego, obejmuje też poziom
- syntaktyczny,
- Semantyczny,
- pragmatyczny.

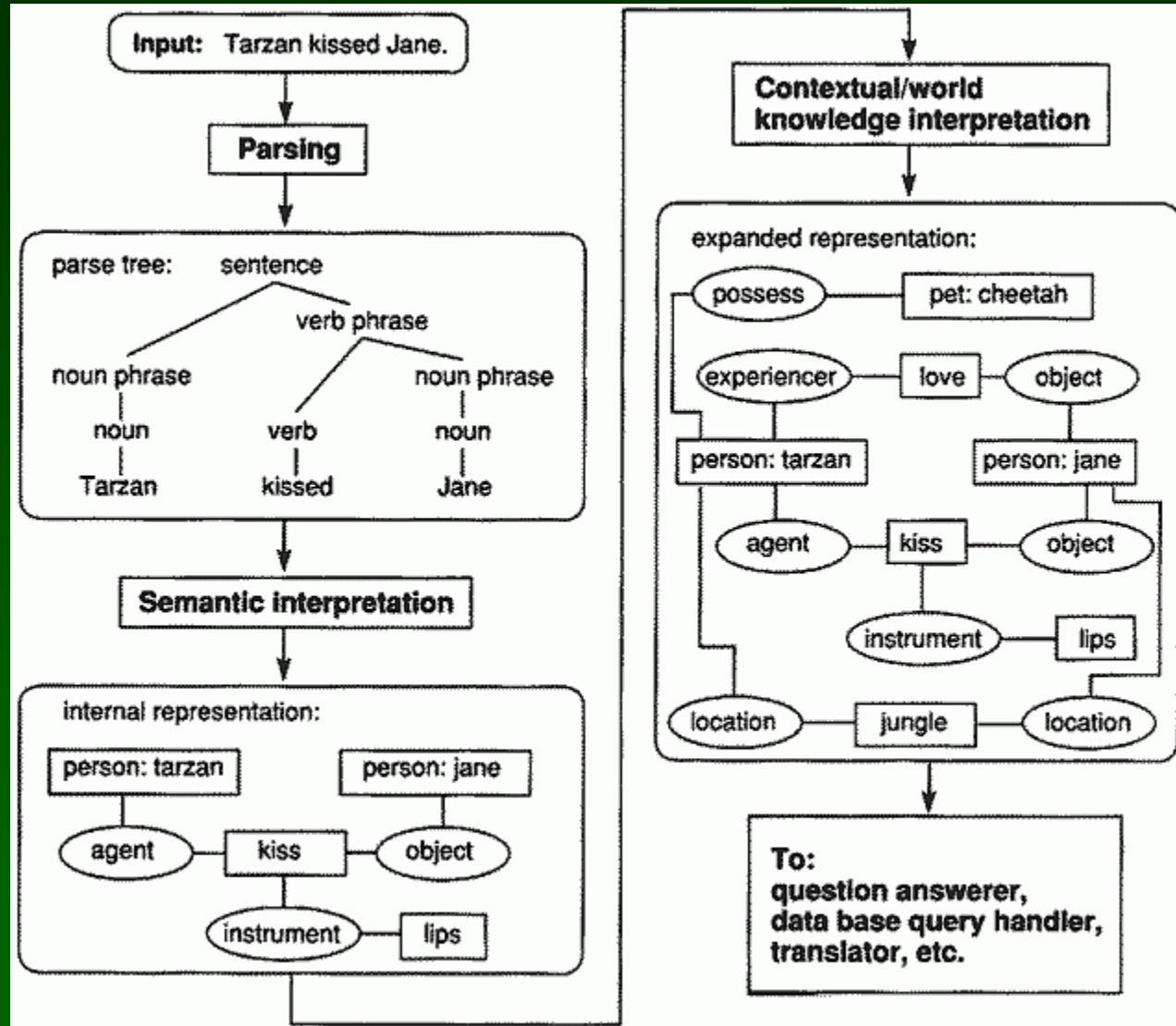
Schemat rozpoznawania mowy



Schemat rozpoznawania



Kroki interpretacyjne



Przykłady programów NLP

Dobrze jest wiedzieć, jakie pomysły wypróbowano. Wczesne programy:

SAD-SAM (R. Lindsay 1963)

Syntactic Appraiser and Diagrammer-Semantic Analyzing Machine, ok. 1700 słów, odpowiada na pytania dotyczące stopnia pokrewieństwa, buduje drzewa genealogiczne, gramatyka bezkontekstowa.

S: Joe bawi się w ogródku ciotki Jane.

Q: Kim jest Joe dla Jane?

BASEBALL (B. Green, 1963)

Baza danych o spotkaniach baseballowych, odpowiadająca na pytania w rodzaju:

Ile gier grali Yankees w lipcu?

Z jakiego miasta jest drużyna Boston?

Przykłady: SIR

SIR (B. Raphael 1968), Semantic Information Retrieval

24 struktury językowe, wyrażające relacje zbiorów, relacje przestrzenne i ilościowe, szablony typu:

* is *, * is part of *, how many * does * have, what is the * of * ?

* = rzeczowniki, lub „wszystkie, dwa, każdy, ... „.

SIR wyciąga proste wnioski z podanych faktów, domagając się ich doprecyzowania:

Każdy chłopiec to człowiek

ROZUMIEM

Palce to część ręki

ROZUMIEM

Człowiek ma 2 ręce

ROZUMIEM

Przykłady: SIR

Ile palców ma John?

POWYŻSZE ZDANIE NIE JEST JEDNOZNACZNE ALE ZAKŁADAM, ŻE ma OZNACZA ma jako części.

NIE WIEM CZY palce TO CZĘŚĆ John

John to chłopiec

ROZUMIEM

Ile palców ma John?

POWYŻSZE ZDANIE NIE JEST JEDNOZNACZNE ALE ZAKŁADAM, ŻE ma OZNACZA ma jako części.

ILE PALCÓW MA RĘKA ?

Każda ręka ma 5 palców.

POWYŻSZE ZDANIE NIE JEST JEDNOZNACZNE ALE ZAKŁADAM, ŻE ma OZNACZA ma jako części.

ODPOWIEDŹ: MA 10

Kto jest prezydentem USA?

NIEZNANA FORMA PYTANIA

Przykłady, STUDENT

STUDENT (D. Bobrow 1968)

Analizuje i rozwiązuje zadania „z treścią” z matematyki, np.:

Jeśli liczba klientów Tomka jest dwa razy większa od kwadratu 20 procent liczby ogłoszeń, które daje, a liczba jego ogłoszeń wynosi 45, ilu Tomek ma klientów?

MTS, Machine Translation System (Wilks 1973)

Korzysta z pośredniej reprezentacji semantycznej, formuł logicznych i wzorców zdań. Ok. 60 elementów podzielonych na 5 klas.

Np klasa *działania* zawiera elementy *przyczyna* i *przemieszczenie*.

„Pić” - działanie ożywionych podmiotów na materiał płynny,

ale: „Samochód pije benzynę”.

Słownik zawiera tłumaczenia wyrazów i stereotypy ich użycia.

Przykłady, LUNAR

LUNAR (Webber 1972) – próbki skał z księżyca

System do wyszukiwania informacji z bazy danych o skałach z wyprawy Apollo-11. Zdania angielskie przekłada na formuły logiczne i używa rachunku predykatów.

Analiza gramatyczna: gramatyki ATN, ok. 3500 słów, oraz reguły semantyczne interpretujące pytania do bazy danych. Radzi sobie z czasami, modalnością i różnymi subtelnościami.

SHRDLU (T. Winograd 1972) – świat klocków, planowanie.

Język aktywuje odpowiednie procedury w słuchaczu.

Reprezentacja proceduralna: wiedza ukryta jest w strukturze programu symulującego robota układającego klocki na stole.

Robot komentuje swoje działania, planuje, ma model swojego mikroświata, odpowiada na pytania.

Nie da się tak działać w szerszym kontekście.

ELIZA

Program ELIZA/Doctor (J. Weizenbaum, MIT 1966).

<http://psych.fullerton.edu/mbirnbaum/psych101/Eliza.htm>

Komputerowy psychoterapeuta ze szkoły Rogers'a (niedyrektywnej); wychwytuje słowa kluczowe, pozwala się wygadać; jest sporo realizacji.

Sekretarka Weizenbauma domagała się prywatnej rozmowy z programem, bo ją „rozumiał”.

Psychiatra dr Colby zasugerował upowszechnienie tego programu w celach leczniczych oraz pisanie programów w celu symulacji pacjentów.

Jest takich programów kilka, ciekawe rezultaty daje konfrontacja komputerowego doktora z pacjentem.

- Jeśli pacjent jest katatoniczny lub w śpiączce to wystarczy program:

Start / Stop

Przykład: ELIZA, naiwne dialogi

Wszyscy mężczyźni są tacy sami.

W jakim sensie

zawsze czegoś od nas chcą.

Czy możesz podać konkretny przykład?

Na przykład mój chłopak mnie tu przysłał.

Twój chłopak cię tu przysłał.

On mówi, że jestem przygnębiona większość czasu.

Przykro mi słyszeć, że jesteś przygnębiona

to prawda. Nie jestem szczęśliwa.

Czy myślisz, że przyjdzie tutaj tobie pomoże?

Na pewno pomoc mi się przyda.

W jaki sposób można by ci pomóc?

Może mogłabym nauczyć się lepiej żyć ze swoją matką.

Opowiedz mi o twojej rodzinie

Program jest bardzo przyjacielski i chętnie słucha zwierzeń a ludzie się na to nabierają.

Margi

MARGI (Meaning, Analysis, Response Generation, Inference) to przykładu intuicyjnego rozumienia języka.

„Koncepcyjna zależność” (conceptual dependency, CD), wewnętrzna reprezentacja znaczenia niezależna od formy.

11 działań, np. transakcja to brać, dawać, przekazać itp.

Związki pomiędzy pojęciami mają postać grafu, kto co komu. Stany obiektów opisane są numerycznie, np. zdrowie (od - 10 martwy do +10), zdrowie umysłowe (+/- 10) i zdrowie fizyczne.

CD używa łańcuchów przyczynowych działań, np.:

- Akcje mogą zmienić stany obiektów.
- Stany mogą umożliwić akcje.
- Stany mogą uniemożliwić akcje.
- Stany mogą zainicjować zdarzenia umysłowe.
- Zdarzenia umysłowe mogą być przyczyną działań.

Margi i SAM/PAM

Informacja w zdaniu jest ujawniana w reprezentacji wewnętrznej.

Np. „Jan je lody łyżeczką” to

„Jan, połyka, lody, łyżeczka, lody na łyżeczce, porusza, usta”.

Pozwala to na parafrazowanie zdań i odpowiadanie na pytania.

SAM (Script Applier Mechanism)

PAM (Plan Applier Mechanism)

Programy demonstrujące użycie scenariuszy i planów w rozumieniu historyjek, oparte na gramatykach CD.

W rozumieniu opowiadań może być ważny kontekst całości.

Scenariusze to stereotypy zachowań, używane do antycypacji zdarzeń.

System potrafi prowadzić sprawne wnioskowanie, np.: z informacji „kotlet był spalony” wnioskuje, że klient był niezadowolony i prawdopodobnie go nie zjadł ani nie dał napiwku. Daje się to zastosować w opisie schematycznych sytuacji.

Problemy NLP

- Problem: brak wiedzy w programach! Trudno jest zrozumieć sens zdania, jeśli się mało wie.
- Reprezentacja znaczenia słów i fraz wymaga reprezentacji wiedzy.
- Konieczna jest nauka znaczenia fraz z przykładów kontekstowych.
- Rozstrzygnięcie, do czego lub kogo odnosi się dany fragment tekstu jest trudne i wymaga zrozumienia kontekstu (problem co-referencji).
- Potrzebne jest dynamiczne dopełnianie fraz na podstawie modelu świata, relacji pomiędzy pojęciami. Można to osiągnąć przez predykcję maskowanych fragmentów tekstu.
- Automatyczna konstrukcja baz danych i uczenie się z tekstów w języku naturalnym wymaga rozumienia tekstu.

Dwa podejścia:

- Budowa wielkich systemów opartych na wiedzy zwerybalizowanej, wykorzystanie metod wnioskowania i klasycznych form reprezentacji wiedzy, np. WATSON, CyC.
- Budowa wielkich sieci neuronowych, które rozpoznają kontekst, dobrze odpowiadają na liczne pytania, ale nie ma w nich głębszego zrozumienia.

Synteza i analiza mowy

Synteza jest stosunkowo prosta, ale stworzenie naturalnie brzmiących głosów udało się dopiero po 2020 roku.

Przykłady mowy syntetycznej.

Zastosowania: przesyłanie informacji tekstowej, synteza po stronie użytkownika, czytanie książek.

Rozpoznawanie mowy:

speaker-dependent, czyli system nauczony rozpoznawania mowy indywidualnej osoby radzi sobie dobrze;

- Na poziomie fonologicznym było to 50-60%, ale nowe modele zagnieżdżania sygnałów w przestrzeniach wektorowych, takie jak Word2Vec, osiągają ponad 90%.
- Na poziomie rozpoznawania słów dla mowy ciągłej i niezależnej od osoby (speaker-independent) komercyjne systemy osiągają 95-98% dokładności, zależnie od warunków. Szumy w pomieszczeniu obniżają dokładność. Dla modeli trenowanych głosem określonej osoby to >98%.
- Testy zrozumiałości mowy robi się na poziomie całych zdań z redundantnym kontekstem.

Praktyczne rezultaty

Dyktowanie tekstu działa w wielu językach w telefonach. WER, czyli Word Error Rate określa się uwzględniając braki i dodatkowe wyrazy:

Word Error Rate = (Substitutions + Insertions + Deletions) / Number of Words

Microsoft w 2017 roku osiągnął błąd na poziomie ekspertów w transkrypcji rozmowy telefonicznej. W obecnie dostępnych programach WER ~5%, ale w rzeczywistych rozmowach przez kiepskie linie telefoniczne to bardziej ponad 10% błędów.

IBM Embedded ViaVoice i ich Super Human Speech Recognition Program, automatycznie generuje podpisy na filmach. Na YouTube to teraz standardowa funkcja.

WATSON API nowe rekordy dokładności, 5.1% błędów dla mowy ciągłej, z użyciem sieci rekurencyjnej LSTM. Ludzie robią w podobnych warunkach około 5% błędów, ale powtarzają, dopytują się, zgadują, więc komunikacja jest sprawna.

IBM WATSON Speech to Text ma liczne zastosowania: podpisy pod wideo, call centers, dyktafony.

W kilku modelach samochodów działają systemy rozpoznawania mowy dla ograniczonej liczby poleceń (<100), pomimo hałasu.

Rozumowanie i rep. lingwistyczne

W 2022 roku pojawiły się duże modele językowe GPT 3 i późniejsze, oparte na transformerach. Wcześniej próbowano wielu pomysłów, np.

- **Metody statystyczne:** Facebook Blender oparty o analizę 1.5 mld dialogów. 49% pytanych woli chatbota od ludzi ... ~10 mld parametrów w sieci NN.
- Inspiracje kognitywne: pamięć rozpoznawcza, semantyczna, epizodyczna pozwalająca tworzyć model dyskursu.
- Podejście koneksjonistyczne, głębokie uczenie i symboliczne do rozumienia, uczenia się i planowania.
- Uczenie się związków przyczynowych.
- Integracja nowej i starej wiedzy.
- Planowanie i argumentacja.
- Modele rozumienia argumentów w tekstach gazetowych.
- Symulacja wyobraźni, procesu marzenia na jawie.
- Odwzorowanie wiedzy na reprezentację tekstową, tworzenie zdań.

Metody sekwencyjne: LSTM

Hochreiter, Schmidhuber, Long short-term memory (1997).

Sieć rekurencyjna (RNN) pozwala na przechowanie wektora w pamięci roboczej.

Trzy bramki:

- o bramka wejściowa: w jakim stopniu wejście wpływa na stan komórki,

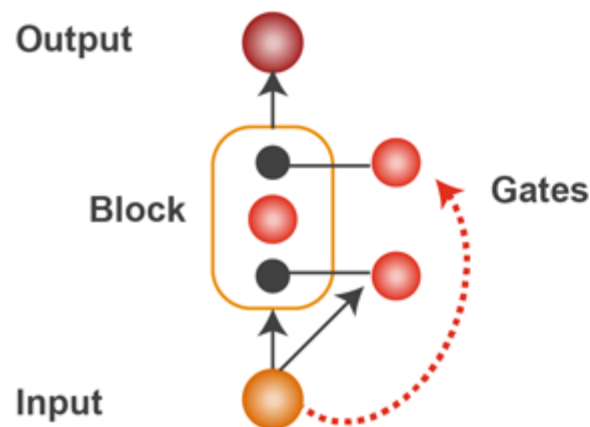
- o bramka zapominania: decyduje o resetowaniu stan komórki

- o bramka wyjściowa: jaką część bieżącego stanu przekazać na wyjście.

Liczne warianty, rozpoznawanie mowy, pisma, muzyki ...

Long short-term memory & gated recurrent unit neural networks

Older forms of RNNs can be lossy. While these older recurrent neural networks only allow small amounts of older information to persist, newer long short-term memory (LSTM) and gated recurrent unit (GRU) neural networks have both long- and short-term memory. In other words, these newer RNNs have greater memory control, allowing previous values to persist or to be reset as necessary for many sequences of steps, avoiding “gradient decay” or eventual degradation of the values passed from step to step. LSTM and GRU networks make this memory control possible with memory blocks and structures called gates that pass or reset values as appropriate.



Source: Genevieve Orr, et al., Williamette University, 1999

Advantages

Long short-term memory and gated recurrent unit neural networks have the same advantages as other recurrent neural networks and are more frequently used than other recurrent neural networks because of their greater memory capabilities.

Use cases

Natural language processing, translation

Linguboty i modele językowe

Technologia: większość botów opierała się na rozszerzeniu techniki szablonów, np. w oparciu o język AIML, często używanego przez chatterboty. Nowsze wykorzystują sieci LMM.

- Amazon: Evi i Alexa od 7/12 po polsku!
- Microsoft Cortana i Apple Siri
- Google Now i Google Assistant i Google Meena
- Samsung Viv i Bixby
- Facebook Blender, bot z analizą emocji, na Github, otwarty projekt.
- Medical prototypes, DARPA's Detection and Computational Analysis of Psychological Signals (DCAPS) project <http://ict.usc.edu/prototypes/current/>
- Leena AI is an AI-powered HR Assistant <https://leena.ai/case-studies>
- Lista polskich botów.
- BotPrize

Linguboty, wirtualni asystenci

Stare boty (chatterbots):

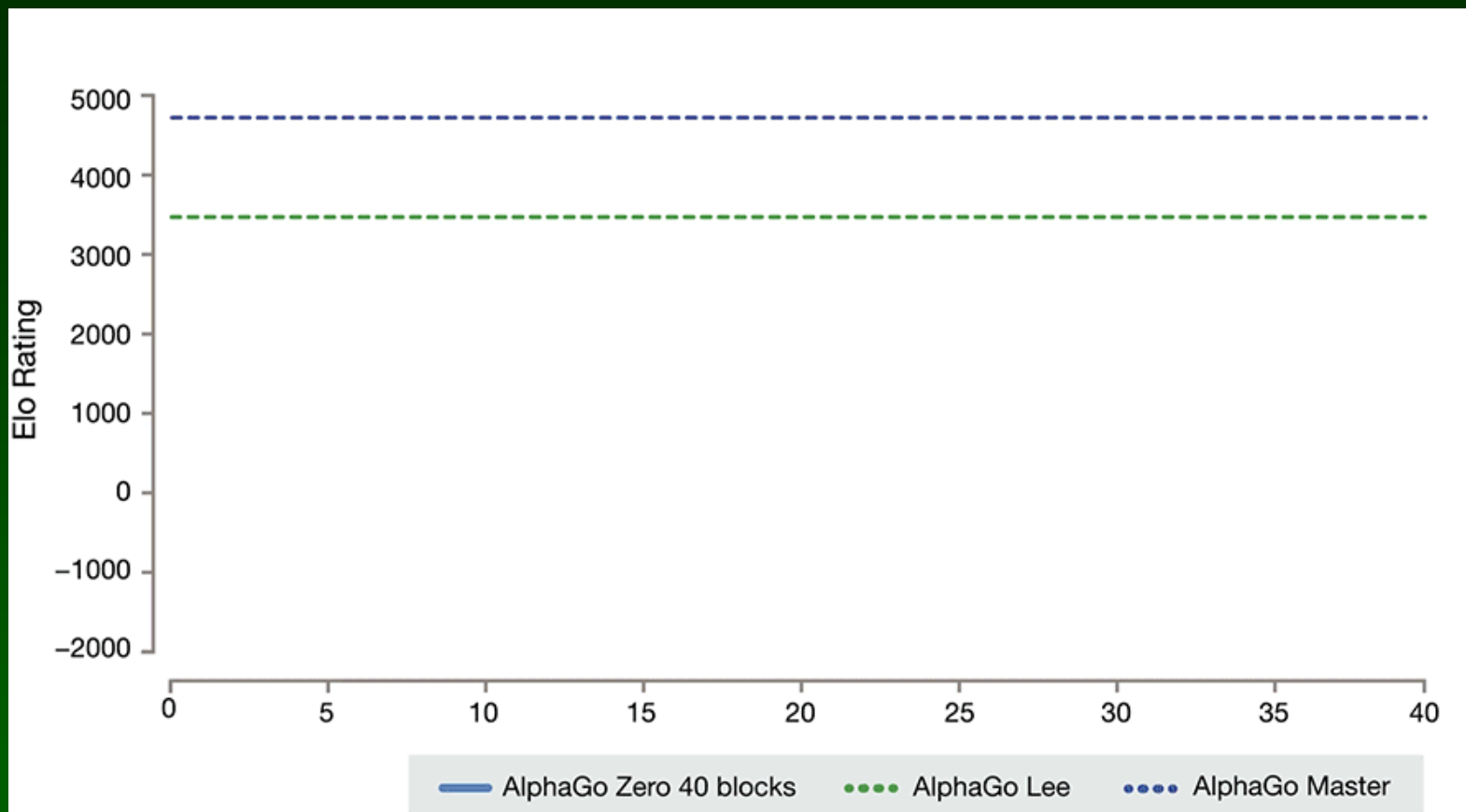
[Agentland](#) | [Chatbots.org](#) (all world) | [Personality forge](#), chatterbot collection | Simon Laven bots, many! |

Zabaware [Ultrahall Assistant](#) to mój ulubiony bot!

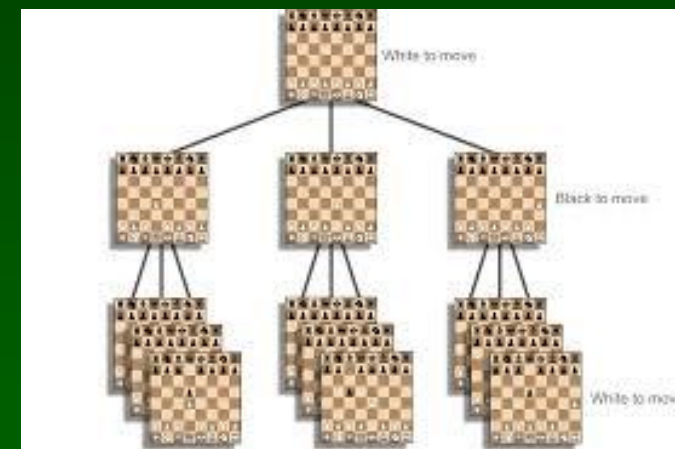


Polskie boty: [katalog](#), [K2bots](#) w biznesie, [voiceboty i chatboty](#),
[Chatboty w Polsce](#) (artykuł)

AlphaGo Zero: nadludzka inteligencja



1997, Deep Blue-Kasparov.
2016, AlphaGo-Lee Sedol
2017, Alpha GoZero
2017, Poker, Dota 2;
2019, Starcraft II,
2022 Stratego, Diplomacy



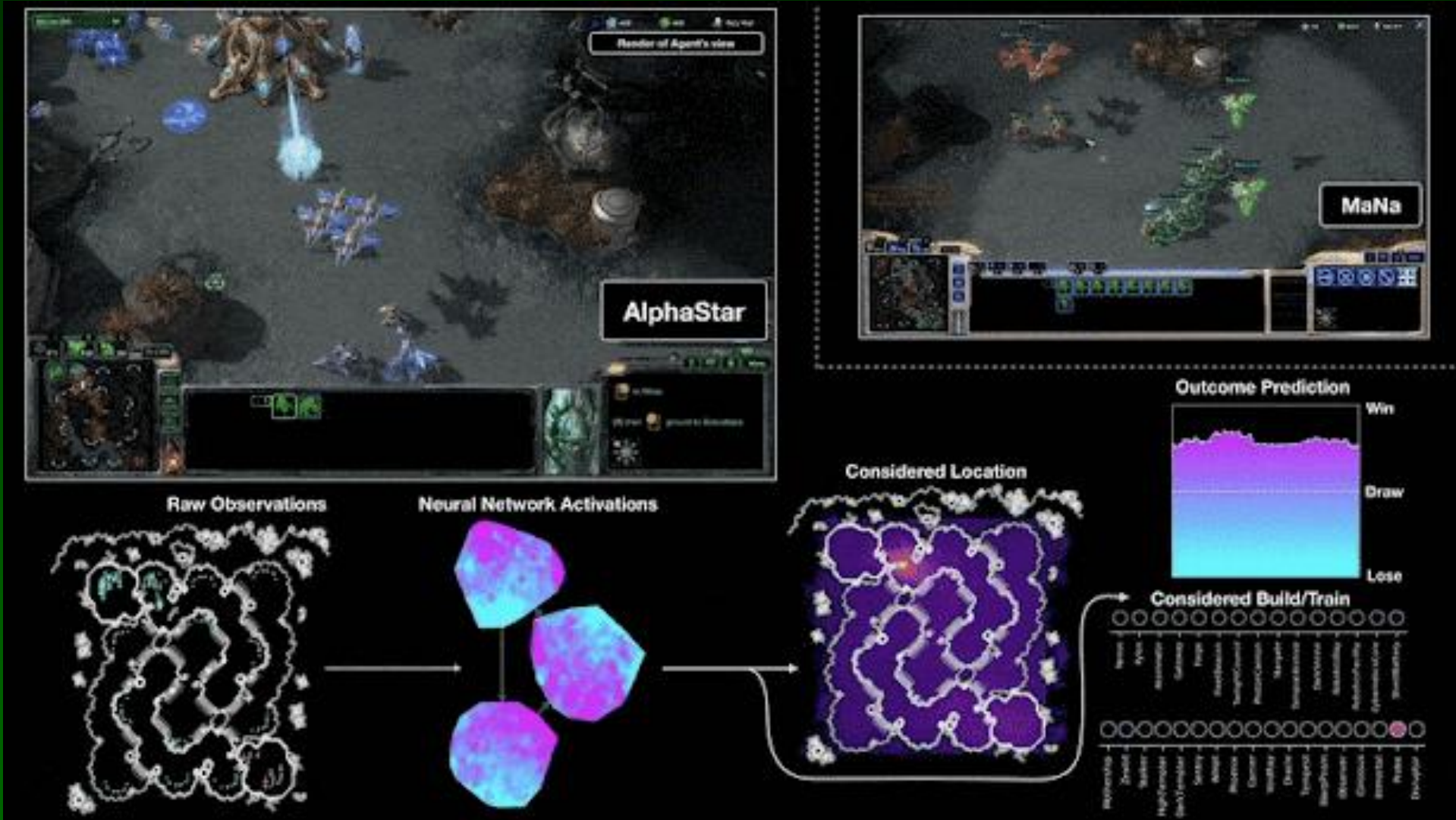
Nadludzki poziom w Go. Najlepsi ludzie 3600-3800 ELO, GoZero 5500 ELO, KataGO ponad 6000 ELO.

MC Tree Search + głęboka ocena wzorców NN jako heurystyka + uczenie ze wzmocnieniem, gra przeciwko własnej kopii. Ludzka wiedza staje się nieistotna, uczenie ludzkich strategii obniża poziom końcowy!

Czysta intuicja: Ruoss ... & Genewein, T. (2024). *Grandmaster-Level Chess Without Search*.

Blitz Chess: transformer model, 270M parametrów, decyzja skojarzeniowa, w **jednym kroku, bez szukania!**

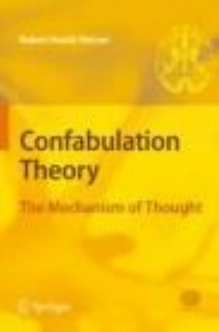
AlphaStar Strategic Game



Deepmind.google alphastar – mastering the real-time strategy game Starcraft-II (2019)
Open, strategic games, like war games.



BERT uczy się języka



Modele językowe: relacje między wektorami(tokenów) w złożonych strukturach sieciowych. W 2018 roku, aby uzyskać uniwersalne „rozumienie języka”, firma Google stworzyła model BERT, Bidirectional Encoder Representations from Transformers, wstępnie wyszkolony na bardzo dużym korpusie tekstowym. To technika uczenia maszynowego oparta na transformerach.

BERT w języku angielskim: dwie sieci, mniejsza o 110 mln parametrów, większy model o 340 mln parametrów w 24 warstwach; przeszkolony na korpusie BooksCorpus zawierającym 800 mln słów oraz Wikipedii zawierającej 2500 mln słów. W 2019 r. BERT działał już w 70 językach.

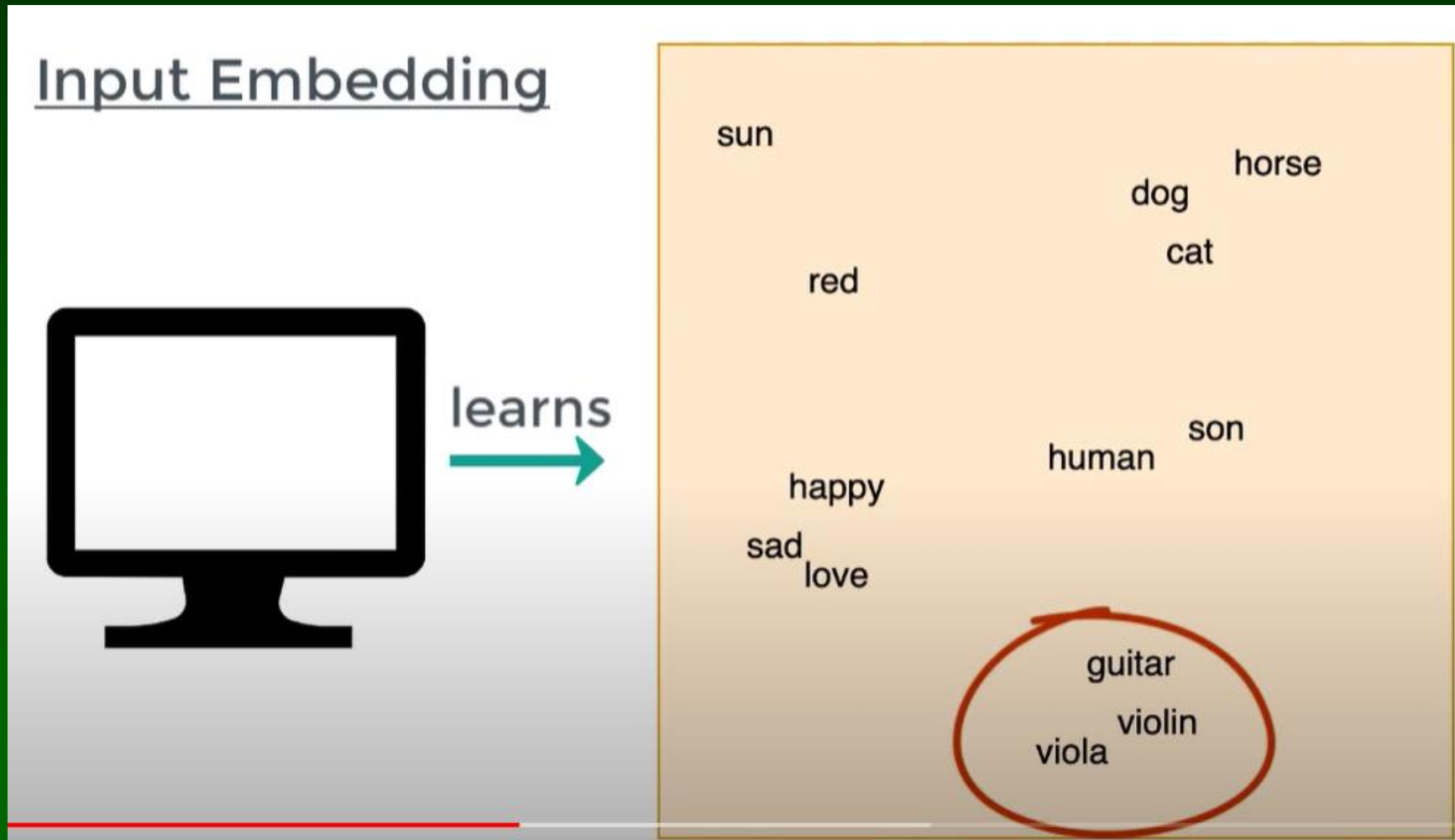
BERT został następnie dostosowany do konkretnych zadań NLP, takich jak odpowiadanie na pytania lub wyszukiwanie informacji semantycznych. Wiele mniejszych, wstępnie wyszkolonych modeli otwartego oprogramowania zostało opublikowanych w repozytorium GitHub.

Sieć uczy się przewidywać zamaskowane słowa (obrazy, sygnały), np:
Dane wejściowe: mężczyzna poszedł do [MASK1]. Kupił [MASK2] mleka.
Etykiety: [MASK1] = sklep; [MASK2] = galon.

Hecht-Nielsen, w Confabulation Theory (2007), proponował przewidywanie kolejnych słów, ale bez dobrego zagnieżdżenia (embedding) i mechanizmu uwagi (relacji słów) to nie działało.

Embeddings

Words => vectors, reflecting their similarity and positions in sentences.



Word2Vec

Jak zamienić dyskretne elementy – słowa – na wektory, które odzwierciedlają sens słów?

<https://learnprompting.thinkific.com/collections>

<https://learnprompting.org/docs/agents/introduction>

<https://evakeiffenheim.substack.com/p/how-to-combine-llms-for-your-breakthrough>

<https://learnprompting.org/docs/agents/mrkl>

Modular Reasoning, Knowledge, and Language or MRKL Systems

LLMs that Reason and Act <https://learnprompting.org/docs/agents/react>

Code as Reasoning <https://learnprompting.org/docs/agents/pal>

Alternatywy: [Glove](#), [FastText](#).

Dobre omówienie: [Advanced Word Embeddings](#)

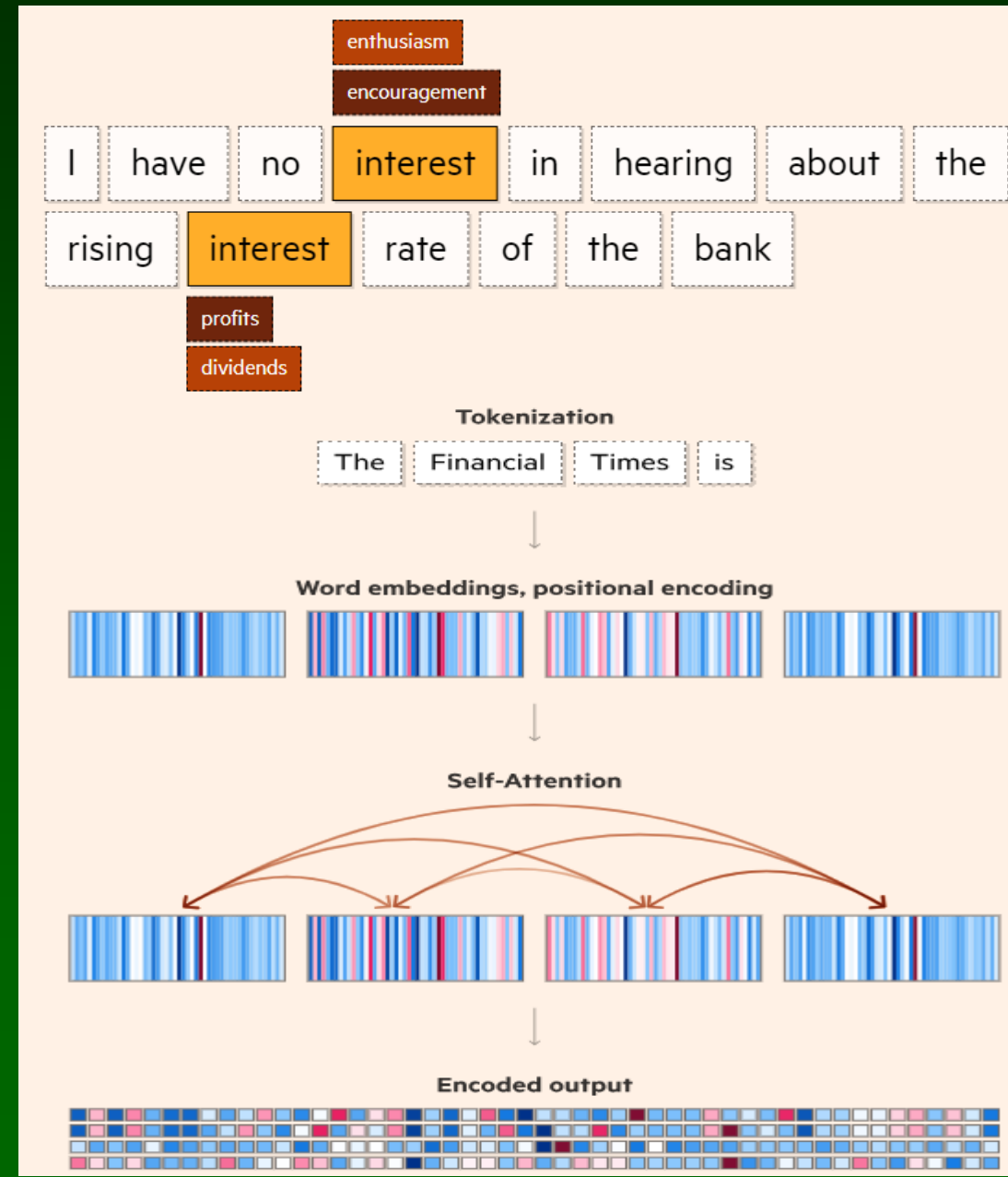
Emeddings and attention

Transformer model published by Google in June 2017 started the generative AI era.

A key concept of the architecture is self-attention to understand relationships between words.

Self-attention links each token in text to other tokens important to understand its meaning.

Matykiewicz P, Pestian J, Duch W, and Johnson N. (2006)
Unambiguous Concept Mapping in Radiology Reports: Graphs of Consistent Concepts, AMIA Ann. Symp Proc. 1024.



Transformers

Attention: given a sequence of tokens (words, image patches), how relevant is each input token to other tokens?

Attention vectors capture context (embedding, semantics)
+ encode relative positions (syntax) of words.

Example:

Input: sentence in English;

Output: sentence in Polish.

Google BERT used this approach.

Generative Pre-trained Transformers or GPTs are now best known models.

[Simple intro on Youtube.](#)

Vaswani et al.(2017).

Attention Is All You Need. arXiv

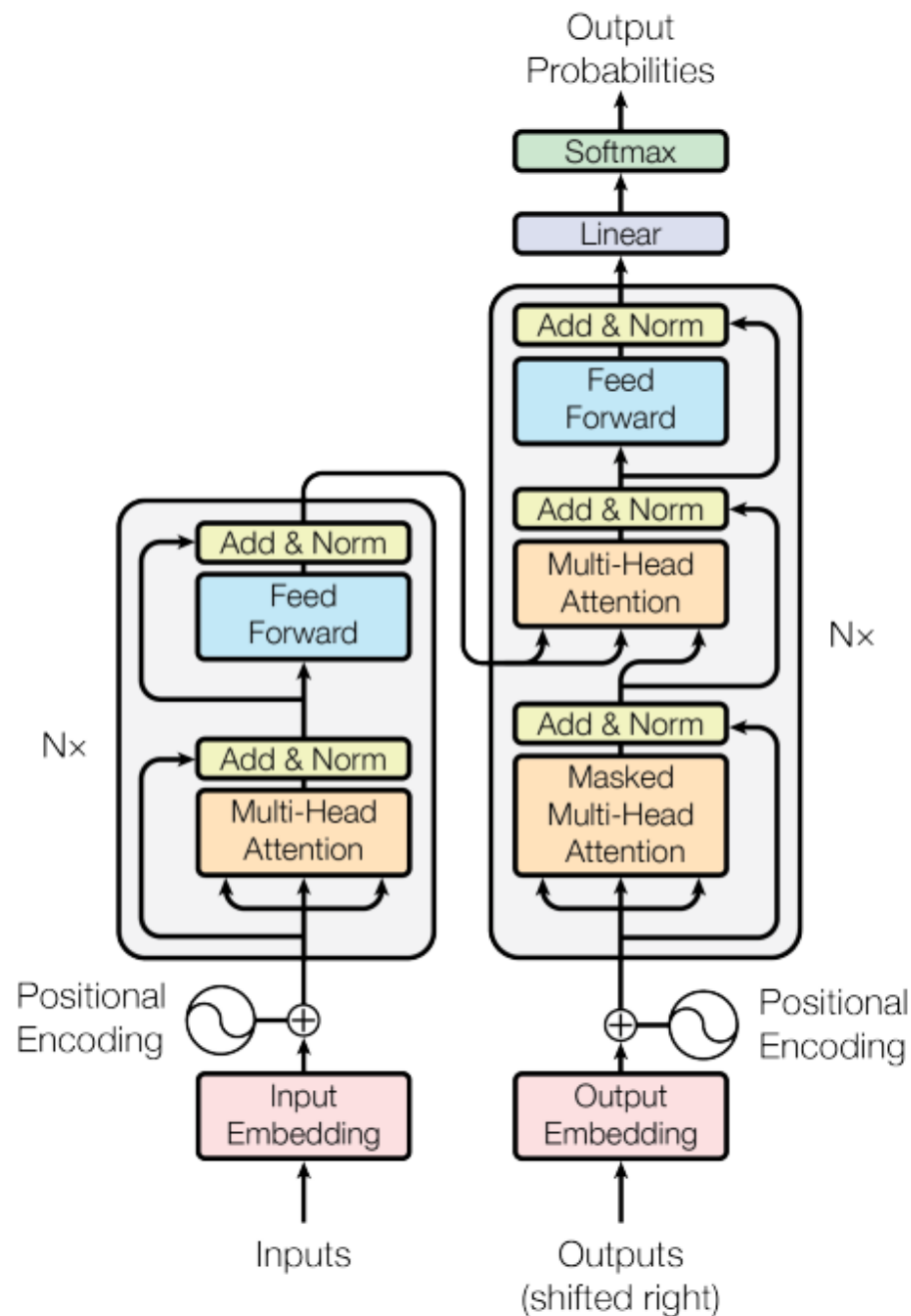
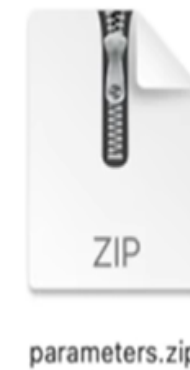


Figure 1: The Transformer - model architecture.

LLM/LMM: kompresja informacji



Trening >10 000 GB tekstów

12 dni pracy 6000 kart GPU V100
2 mln \$, 10^{24} mld mld mld operacji.

parametry $\approx$ 70 GB

Llama 70B, kompresja 142 razy, to umożliwia odległe skojarzenia, które trudno jest odkryć.

Kompresja 8, 4, 2 bity => większa kreatywność, luźne skojarzenia, więcej halucynacji.

Więcej szumu (wyższa temperatura) => większa kreatywność, więcej halucynacji.

Kontekst, RAG, prompty: aktywują interesujące nas obszary wiedzy.

Słowa, obrazy, myśli pobudzają mózgi i LLMy

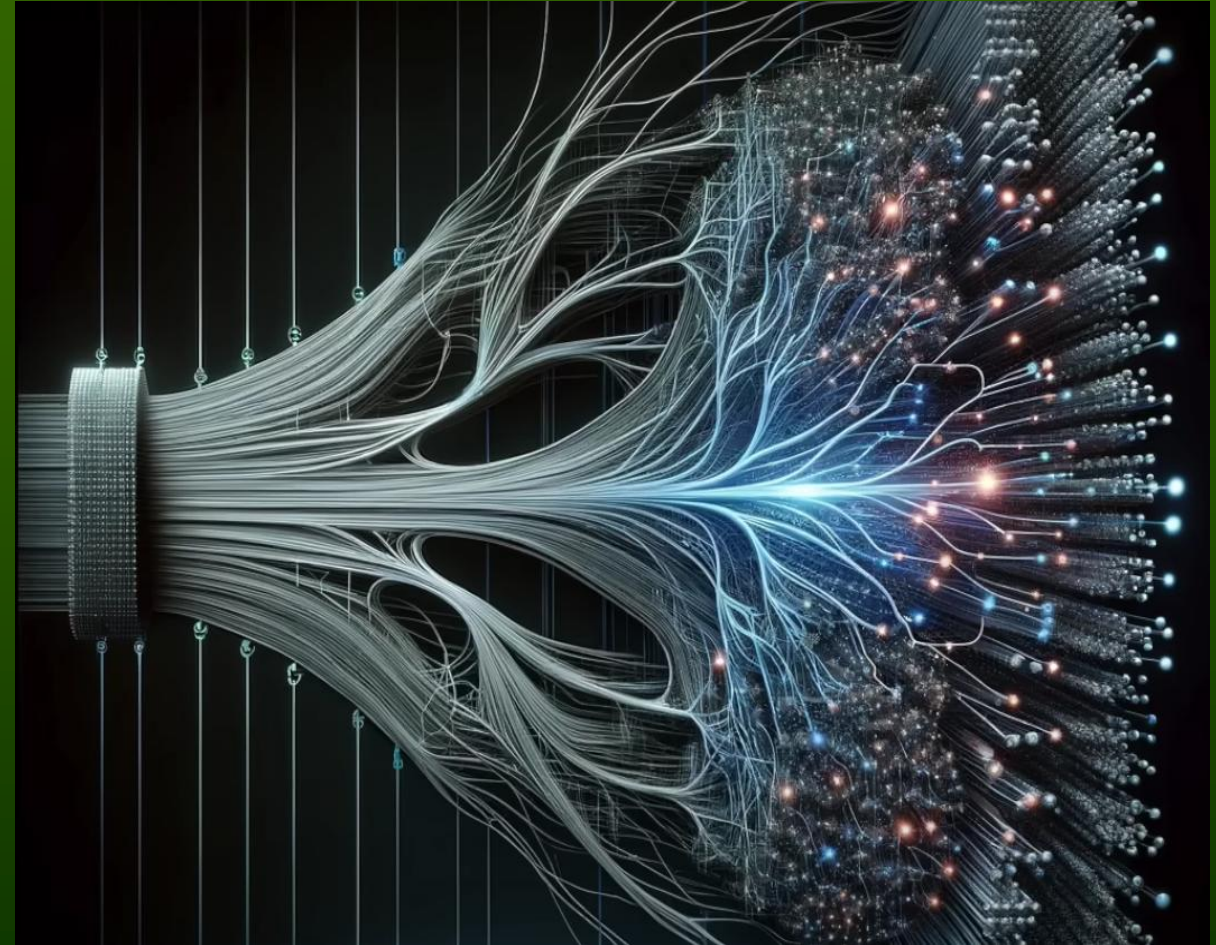
GPT (2017) = Generative Pre-trained Transformers, czyli generatywne, wstępnie nauczone transformery.

Sieci rozchodzących się aktywacji, transformacji.

Dane wejściowe => tokeny => osadzanie (embedding) w przestrzeni wektorowej tak, by zachować podobieństwo znaczenia (semantyki) w różnych kontekstach.

Zwracaj uwagę na powiązania między wszystkimi tokenami, dodaj taką informację do ich reprezentacji, użyj takich wektorów by nauczyć sieć odpowiadać na pytania. Wybierz podgrafy skojarzonych tokenów.

LLM visualization, <https://bbycroft.net/llm>



Duch et al., Towards Understanding of Natural Language: Neurocognitive Inspirations. 2007

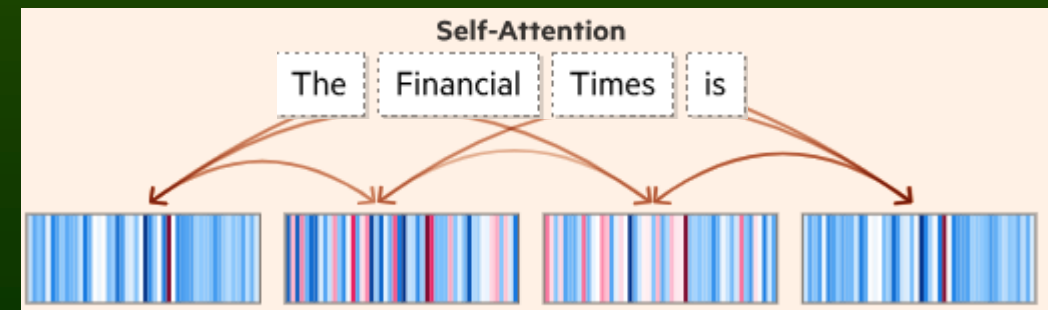
Jak działają LLM?

How transformers work (The Financial Times + visual storytelling)

- Transformery: w 2017 roku Google zaczął rewolucję w AI przedstawiając nowe podejście do sieci neuronowych. Publikacja Attention is all you need, pokazała jak budować programy analizujące teksty w oparciu o mechanizm uwagi, łączący różne słowa.
- Wizualizacja na stronach FT pokazuje jak działają transformery na prostych przykładach, dlatego pozwoliło to chatbotom tak dobrze opanować język.
- LLM mapują biliony słów na wielkie tablice liczb, osadzając tokeny w wielowymiarowych przestrzeniach wektorowych, nadając im sens, kodując cechy i relacje. To pozwala przewidzieć następne słowo w dowolnej sekwencji, ale też wiarygodność całego zdania.
- Algorytmy są dość proste, liczy się wielkość sieci i zbiorów treningowych.

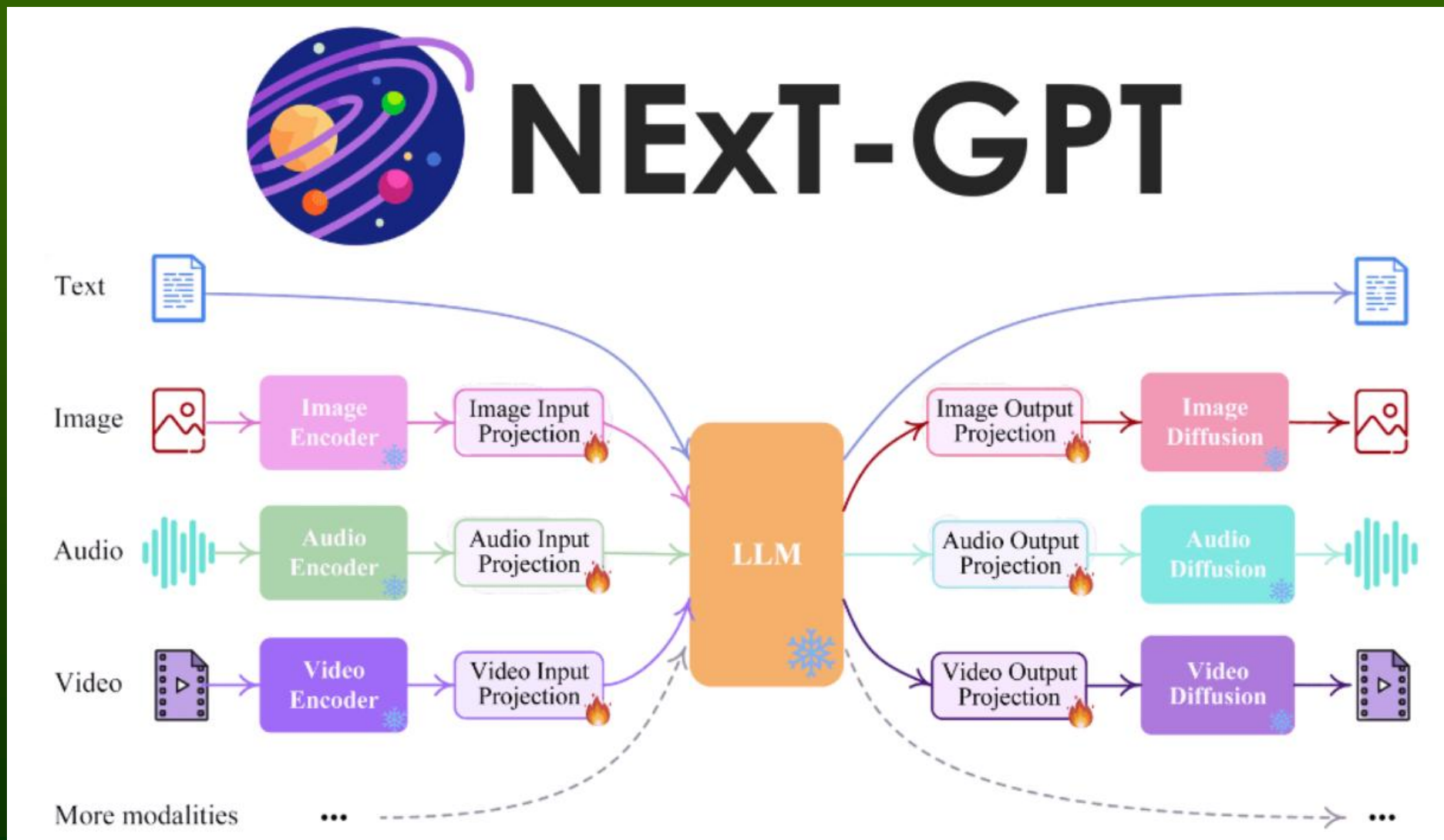
Język jest kluczem do wszystkiego:
obrazy, sygnały, ruch robotów, DNA, białka ...

- Andrej Karpathy: Intro to Large Language Models



GPT skojarzy wszystko

Substytucja zmysłów: cokolwiek wrzucimy do mózgu będzie wykorzystane.



Camera

Tongue Display



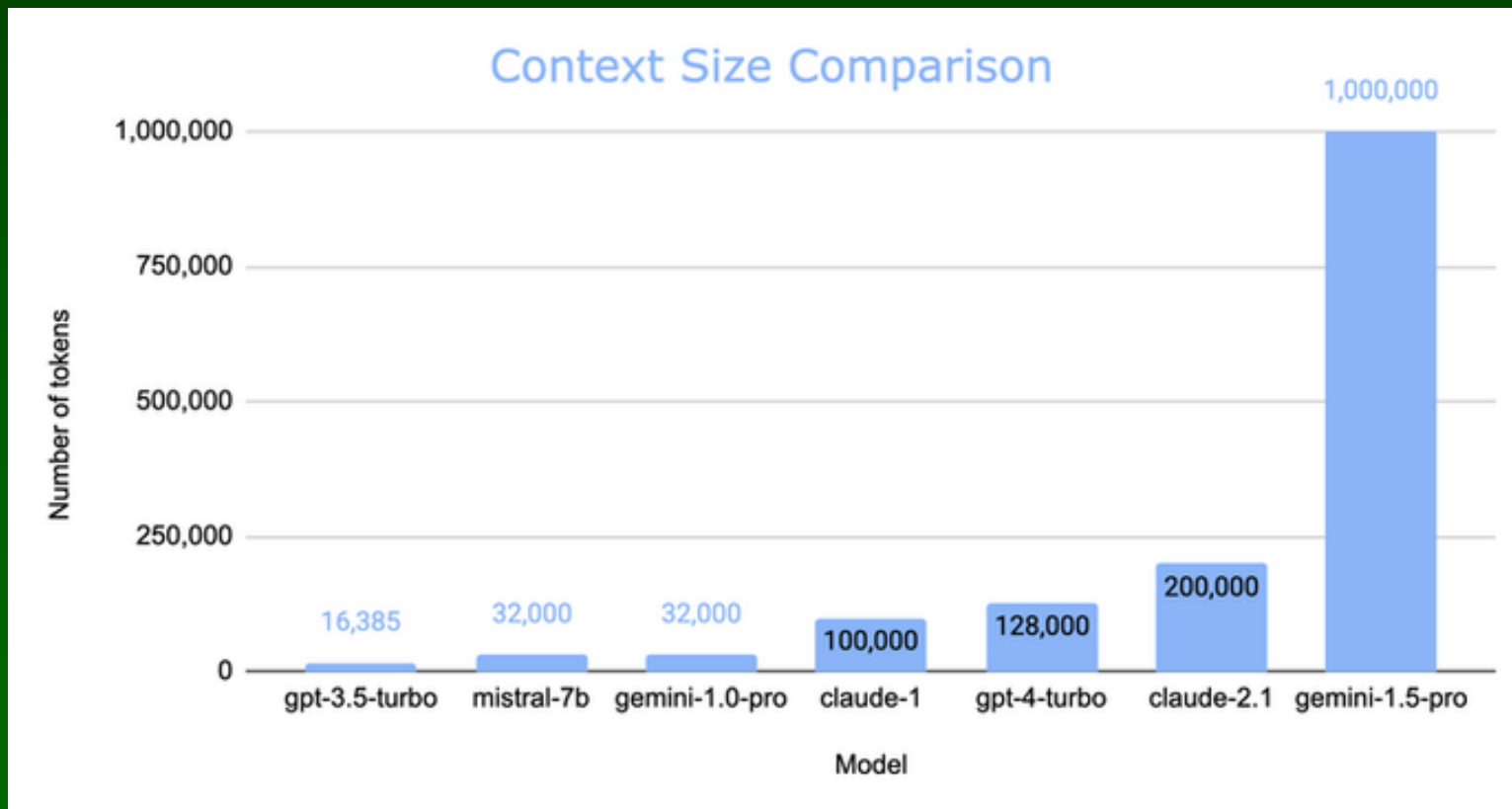
Source: BrainPort



Kontekst

Modele LLM są trenowane na wybranym zbiorze tekstów, modele LMM na danych obrazowych, wideo, sygnałach – to tworzy pamięć długotrwałą (ale nie wierną).

Prompt dostarcza pytanie i kontekst, dokładne instrukcje czego oczekujemy od danego modelu. W niektórych modelach kontekst może osiągnąć ponad 100.000 tokenów, np. wiele tomów akt. Google Notebook LM pozwala na 2.5 mln, [Llama Scout 10 mln](#).



Polskie projekty



Stworzenie dużego modelu fundacyjnego jest bardzo drogie, ale są otwarte modele, które można łatwiej douczyć.

Projekt PLLuM (Polish Large Language Model), OPI, NASK, IPI PAN, Pol Wrocławska, Uniw. Łódzki.

Planowane jest wykorzystanie w administracji publicznej i sektorze prywatnym, np. w postaci prototypowego inteligentnego asystenta.

BIELIK, projekt budowany oddolnie, efekt pracy społeczności entuzjastów sztucznej inteligencji.

Qra to odpowiednik otwartych narzędzi Mety czy Mistral AI. Lepiej rozumie pytania i treści w języku polskim i lepiej sama tworzy spójne teksty.

LLama2 90B posłużyła jako start dla projektu **Qra**. Politechnika Gdańska i

AI Lab z (OPI) – PIB, opracowały polskojęzyczne generatywne neuronowe modele językowe na bazie terabajta danych tekstowych wyłącznie w języku polskim.

PLAMA to projekt ogłoszony 25.04.2024.

Kompleksowa Lista Ewaluacji polskich modeli Językowych **KLEJ**.

Świetne kursy

- [UWaterloo CS 886 papers/slides/videos !](#) - Recent Advances on Foundation Models.
- [Stanford CS25-Transformers United, video](#)
- [Princeton COS 484 NLP Course \(2025\)](#)
- [Princeton COS 597R Deep Dive into Large Language Models \(Fall 2024\)](#)
- [Maxime Labonne LLM-course + Colab notebooks](#)
- [Darmowy kurs "Illustrated transformer"](#)
- [Stanford CS324 - Large Language Models \(2022\)](#)
- [Neurips2022 Tutorial-Foundational Robustness of Foundation Models](#)
- [ICML2022-"Big Model" Era: Techniques and Systems \(Berkeley\)](#)

Inne linki

- [Andrej Karpathy wyjaśnia](#) |
 - [Let's build GPT: from scratch, in code, spelled out](#)
 - [GPT Tokenizer](#)
 - [Why you should work on AI AGENTS](#)
- [LLM Powered Autonomous Agents](#)
- [Umar Jamil Series, kilka modeli](#) - edukacyjne wideo o nowych modelach (Titans).
- [Alexander Rush projekty OpenSource.](#)
- [A Visual Guide to Transformers, Mamba and State Space Models \(Maarten Grootendorst\)](#)
- [Create GPT in 60 Lines of NumPy](#)
- [femtoGPT, minimal GPT implementation](#) Pure Rust

Książki, prompty

- Prompt engineering examples
- **LLM Books**
 - Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs - it comes with a GitHub repository that showcases a lot of the functionality
 - Build a Large Language Model (From Scratch) - A guide to building your own working LLM.
 - BUILD GPT: HOW AI WORKS - explains how to code a Generative Pre-trained Transformer, or GPT, from scratch.
 - Hands-On Large Language Models: Language Understanding and Generation - Explore the world of Large Language Models with over 275 custom made figures in this illustrated guide!

Książki, prompty

- Prompt engineering examples
- **LLM Books**
 - Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs - it comes with a GitHub repository that showcases a lot of the functionality
 - Build a Large Language Model (From Scratch) - A guide to building your own working LLM.
 - BUILD GPT: HOW AI WORKS - explains how to code a Generative Pre-trained Transformer, or GPT, from scratch.
 - Hands-On Large Language Models: Language Understanding and Generation - Explore the world of Large Language Models with over 275 custom made figures in this illustrated guide!