

# Modele językowe. Wprowadzenie

Paweł Rychlikowski

Instytut Informatyki UWr

8 października 2024

# Wykłady

- Zapraszam w każdy wtorek (niestety rano)!
- Mile widziane przerywanie pytaniami, itd.

Wykład jest prowadzony pierwszy raz, aczkolwiek jakieś części treści będą brane z innych wykładów

# Zajęcia pomocnicze

- Mamy ćwiczenia i pracownie, standardowo w rytmie:  
 $5 * (P + P + C)$
- Na pierwsze 3 pracownie będą pełne listy zadań, od czwartej zaczynamy myśleć o projekcie (aczkolwiek w P4 jakieś zadania jeszcze będą, ale mniej)

## Projekt

- Projekt będzie dość wysoko punktowany, ale formalnie nieobowiązkowy.
- Można go będzie realizować w zespole
- Częścią trudności projektu jest wymyślenie tematu, ale bez obaw: będą różne pomysły na zajęciach pomocniczych.

Treścią pracowni są indywidualne konsultacje:

- dotyczące zadań, które Student(ka) wykonał(a)
- oraz tych, z którymi ma kłopot

# Pracownia

Kilka drobiazgów:

- Sprawdzanie może być selektywne – prowadzący nie musi oglądać wszystkich zadań (ale za wszystkie 'zadeklarowane' powinien przyznać punkty)
- Student ma prawo zadawać różne pytania, prosić o wskazówki, o pomoc przy debugowaniu. itp
- Prowadzący decyduje, jak wygląda opowiadanie o zadaniu. Może na przykład poprosić o zreferowanie rozwiązania, albo zadać pytanie o jakiś fragment, albo przeczytać kod i stwierdzić, że wszystko jasne, albo ...

## Prośba od prowadzących

Opowiadając o zadaniu nie trzeba mówić o wszystkich funkcjach. Znaczną część rutynowych rzeczy można (należy) pomijać

- Wszystkie programy będzie można wykonać w Pythonie
- Używanie innego języka nie jest zabronione, ale może być istotnie mniej wygodne

# Pierwsza pracownia

- Zasadniczo powinno się dać ją rozwiązać za pomocą informacji z dzisiejszego wykładu
- Można też spytać o dowolne zadanie, również dzisiaj po wykładzie

# Ćwiczenia

- System deklarycyjny (przed zajęciami mówimy, co umiemy i za te zadania dostajemy punkty)
- Dodatkowa zachęta dla rozwiązujących: bonus +2 za 50%



# Punkty, zaliczenia, egzamin

- Za prezentację (i wcześniejsze napisanie) programów studenci dostają punkty
- Podobnie za projekt.
- Podobnie za deklaracje zadań na ćwiczeniach

**Wszystkie punkty są równe!**

Sumujemy je i wyznaczamy ocenę (szczegóły będą w regulaminie)

DEVELOPER NEWS

# CHATGPT CHANGES EVERYTHING




**CHATGPT AND  
GENERATIVE AI:  
THE FUTURE OF HUMAN  
DECISION MAKING**



WITH THE CODE BY



It's No Big Deal,  
but  ChatGPT  
Changes Everything



## ChatGPT 4 - Redefining the AI Landscape

SIT TIGHT.  
**CHATGPT IS ABOUT TO**  
CHANGE EVERYTHING IN YOUR CLASSROOM

[teachergoals.com/blog](https://teachergoals.com/blog)



## ChatGPT AI

Transforming the Modern  
Business Landscape



# O czym będzie przedmiot (b. ogólnie)

- Czym są modele językowe i jak ich używać.
- W jaki sposób sieci neuronowe są używane w modelowaniu języka.
- Jak poszerzać zakres zastosowań LM-ów
- Czy i gdzie są niebezpieczeństwa i ograniczenia LM-ów

## Large (?) Language Models

Będziemy mówić również o rzeczach związanych z naprawdę dużymi modelami językowymi, ale na pracowni i w projekcie zatrzymamy się na mniejszej skali.

# Definicja modelu językowego

Zamiast jednej definicji podamy cztery prawie-definicje, patrzące na model językowy na czterech różnych poziomach:

- **Poziom 0:** aplikacja-chatbota, (ChatGPT, Copilot, ...)
- **Poziom 1:** API, procedura generująca napisy, karmiona innymi napisami
- **Poziom 2:** model językowy w wąskim sensie: rozkład prawdopodobieństwa kolejnego tokena (warunkowany ciągiem poprzednich tokenów)
- **Poziom 3:** sieć neuronowa obliczająca rozkład z poziomu 2

Jak za chwilę zobaczymy, te perspektywy trochę na siebie nachodzą.

Nasz wykład będzie operował głównie na poziomach 1-3, na początku będziemy pomijać poziom 3.

## Definicja

**Molel językowy (P0)** jest aplikacją, która posługuje się interfejsem chatbota tekstowego i wykorzystuje jako istotną część swojego działania modele językowe na dalszych poziomach.

## Definicja

**Molel językowy (P0)** jest aplikacją, która posługuje się interfejsem chatbota tekstowego i wykorzystuje jako istotną część swojego działania modele językowe na dalszych poziomach.

- Oczywistym przykładem jest ChatGPT
- Wydaje się, że w wielu miejscach jest to **naprawdę** użyteczne



Przygotowuję wykład dotyczący modeli językowych. Wymyśliłem, że zamiast podać jedną definicję, podam 4 na różnych poziomach szczegółowości. Poziom 0 to poziom aplikacji (takich jak ChatGPT), poziom 3 – to konkretne sieci neuronowe (na przykład transformer). Czy wg Ciebie powinny być poziomy 1 i 2?



Świetny pomysł na uporządkowanie definicji modeli językowych w sposób dostosowany do różnych poziomów szczegółowości! Poziomy 1 i 2 mogą służyć jako mosty między bardzo ogólnym podejściem aplikacyjnym (Poziom 0) a technicznymi szczegółami związanymi z architekturą modeli neuronowych (Poziom 3). Oto, jak możesz je zdefiniować:

### Poziom 1: Abstrakcja algorytmiczna — modelowanie języka naturalnego

Na tym poziomie można opisać model językowy jako system, który przewiduje lub generuje tekst na podstawie danych wejściowych, opierając się na prawdopodobieństwie wystąpienia sekwencji słów. Ten poziom skupia się na koncepcji samego **modelowania języka**, bez konieczności wchodzenia w detale techniczne.

Przykład:

- Model językowy to algorytm, który „uczy się” przewidywać kolejne słowa w zdaniach na podstawie wzorców w danych tekstowych...  
↓  
...bazuje na prawdopodobieństwach statystycznych i

# Przykładowa książka na poziomie 0



Oczywiście takich książek jest (będzie) dużo więcej.



# Jak pisać do ChataGPT

Porady od p. Valentyny:

- **Jasność** – używaj prostych zdań i instrukcji.
- **Spójność** – stosuj krótkie teksty, w razie czego dzieląc tekst na kilka instrukcji
- **Skupienie** – treść zapytania musi być mocno skoncentrowana na konkretnym temacie (żeby nie uzyskać zbyt ogólnej odpowiedzi)
- **Niezmiennność** – zachowaj stały ton i język podczas konwersacji
- **Zmiana roli** – poleć CGPT by zachowywał się jak ktoś lub coś

Nie, nie musimy ich się uczyć na pamięć!

# Poziom 1

## Definicja

**Model językowy (P1)**, jest procedurą, która na wejściu otrzymuje tekst i która generuje tekst związany z wejściem. Elementem tej generacji jest **próbkowanie** z modelu językowego (poziom 2)

- Na tym poziomie mamy API, a nie aplikację.
- Obliczenia są niedeterministyczne (tzn. nie można tu myśleć o funkcji)
- Podstawowym trybem jest tu tryb **dokończ tekst** (tekst będący argumentem)
- Można o też tym myśleć jako o interpreterze wysokopoziomowego języka, mocno bazującym na języku naturalnym.

ChatGPT nazwał ten poziom: **Abstrakcją algorytmiczną**

# Stochastic Parrot



# Papuga, która mówi po polsku

Zajmiemy się modelem **papuGaPT2**, wytrenowanym na polskich tekstach odpowiedniku GPT-2.

Pełen kod programu, który uruchomimy:

```
1 from transformers import pipeline, set_seed
2
3 generator = pipeline('text-generation', model='flax-community/papuGaPT2')
4
5 print ('Model loaded')
6 last_prompt = 'Dzisiaj na obiad zjemy kartofelki z'
7
8 while True:
9     prompt = input().strip()
10    if not prompt:
11        prompt = last_prompt
12    g = generator(prompt,
13        pad_token_id=generator.tokenizer.eos_token_id)[0]['generated_text']
14
15    print (g)
16    print (50 * '=')
17    print ()
18    last_prompt = prompt
```

# Poziom 1 (cd)

## Definicja

**Prompt (zachęta, monit)**, to po prostu tekst  $x$ , który podajemy do API

## Uwaga

Podstawową techniką na poziomie 1 jest konstruowanie promptów (będących czasem wynikiem pewnych obliczeń, o różnym stopniu skomplikowania).

# Poziom 1 (cd)

Najprostszy wariant:

```
prompt = system_prompt + user_prompt
```

Czasem można to interpretować tak:

- `system_prompt` – opis działania do wykonania, w języku angielskim (polskim, niemieckim, ...). Czyli taki jakby `program`
- `user_prompt` – dane wejściowe

# Co jeszcze się dzieje na tym poziomie?

Mnóstwo rzeczy:

- Wielokrotne wywoływanie tego samego modelu (po co?)
- Dialog między **instancjami** (tak jakby) tego samego modelu
  - ▶ Instancje różnią się tylko `system_prompt-em`
- **Few-shots learning**

O niektórych powiemy sobie teraz, o innych później.

# Poziom 1 vs Poziom 0 (ChatGPT)

Zastanówmy się nad tym, co wprowadza poziom 0 dla ChataGPT

- **System prompt** (oczywiście nie znamy, ale to powinno wyglądać jakoś tak):

Jesteś uczynnym i starającym się być jak najbardziej użytecznym botem, nazywającym się ChatGPT, który uprzejmie wykonuje polecenia użytkownika i odpowiada na jego pytania. Wypowiedzi użytkownika zaczynają się od znacznika **[user:]**, twoje wypowiedzi zaczynają się od znacznika **[bot:]**. Każdą wypowiedź kończysz znacznikiem **[eot]**

- Co jeszcze, oprócz ostatniej wypowiedzi użytkownika jest włączane do prompta? Cała historia rozmowy, czy jakaś jej część? A może dla długich dialogów streszczać początek?
- Czy coś robimy z wygenerowaną odpowiedzią? Na pewno usuwamy znacznik **[eot]**, ale może coś jeszcze?



# Standard Machine Learning



# One-shot Machine Learning



# In-Context Learning

## No Prompt

Zero-shot  
(0s)

skicts = sticks

1-shot  
(1s)

chiar = chair  
skicts = sticks

Few-shot  
(FS)

chiar = chair  
[...]  
pciinc = picnic  
skicts = sticks

## Prompt

Please unscramble the letters into  
a word, and write that word:

skicts = sticks

Please unscramble the letters into  
a word, and write that word:

chiar = chair  
skicts = sticks

Please unscramble the letters into  
a word, and write that word:

chiar = chair  
[...]  
pciinc = picnic  
skicts = sticks

# Inne przykłady

Przykłady zadań, które nadają się do in-context-learning:

- Tłumaczenie słów
- Klasyfikacja krótkich tekstów
- Odpowiadanie na pytania
- Tłumaczenie zapytania po angielsku na SQL
- ...

# Zero-shot learning

## Sample Hotel Review

Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax.

We can get the model to classify the sentiment of this text by taking the text and appending an incomplete statement to the review like `In short, our stay was:`

## A prompt consisting of a review plus an incomplete statement

Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax. In short, our stay was

## Zero-shot learning (2)

### Hotel Review Completions

Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relaxed. In short our stay was

... not a pleasant one. The staff at the front desk were not welcoming or friendly, and seemed disinterested in providing good customer service.

... uncomfortable and not worth the price we paid. We will not be returning to this hotel.

## Zero-shot learning (3)

### Hotel Review Completions

Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relaxed. In short our stay was

... not a pleasant one. The staff at the front desk were not welcoming or friendly, and seemed disinterested in providing good customer service.

... uncomfortable and not worth the price we paid. We will not be returning to this hotel.

## Poziom 2

### Definicja

**Model językowy (P2)** jest rozkładem prawdopodobieństwa

$$P(w_n | w_1 \dots w_{n-1})$$

gdzie  $w_i$  są kolejnymi **tokenami** tekstu.

### Uwaga

Do pracy na tym poziomie konieczna jest zatem **tokenizacja**, czyli funkcja zamieniająca napis na ciąg tokenów.

**Pierwsze przybliżenie:** `txt.split()`



## Poziom 2 (cd)

- Prawdopodobieństwo sekwencji tokenów można obliczyć następująco:

$$P(w_1 \dots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \dots P(w_n|w_1 \dots w_{n-1})$$

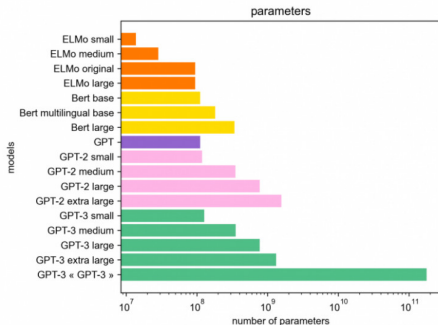
- Tak zdefiniowany model prawdopodobieństwa daje możliwość próbkowania tekstu

### Uwaga

Rozkład prawdopodobieństwa wyznaczany jest na podstawie **korpusu** – dużego zbioru tekstów.

ChatGPT (prawie) nazwał ten poziom: **Abstrakcją stochastyczną**

# Wielkości korpusów



More recent models: PaLM (540B), OPT (175B), BLOOM (176B)...

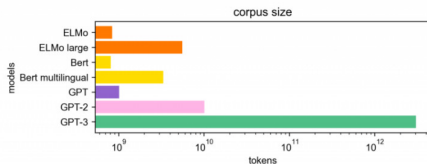


Image source: <https://hellofuture.orange.com/en/the-gpt-3-language-model-revolution-or-evolution/>

# Wykorzystanie prawdopodobieństwa zdania

Zdanie bardziej prawdopodobne = zdanie bardziej naturalne



*Sfklkljf fskjhfkjsh kjfs fs kjhkjhs fsjhfkshkjfh*

**Low** probability



*ChatGPT is all you need*

**high** probability