

Generativní modely

Aleš Horák

E-mail: hales@fi.muni.cz
<http://nlp.fi.muni.cz/uui/>

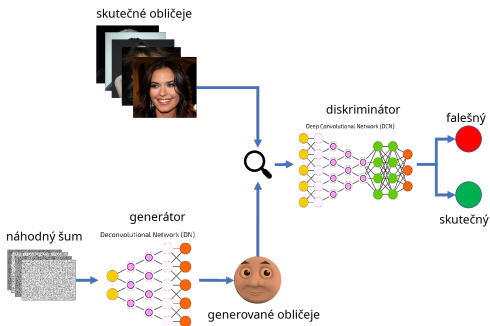
Obsah:

- Generování obrazu
- Generování textu
- Asistenční modely typu ChatGPT
- Metodologie tvorby otázek
- PA026 – Projekt z umělé inteligence

Generování obrazu

Generative Adversarial Network (GAN)

- *generující soupeřící síť*
- 2 části – **generátor** a **diskriminátor** trénované současně **bez dohledu**
- **generátor** – vytváří **simulované** vstupy podle zadání, snaží se zmást diskriminátor
- **diskriminátor** – učí se rozpoznávat **skutečné** vstupy od **podvržených**



(Missinato, 2020)

Difuzní modely

- **třénování** na sadě obrázků
- postupné přidávání **šumu** do obrázku
- **generování** – obrácení postupu, odšumování (*denoising*)
- současné generátory **Dall-E 3**, **Midjourney**, **Stable Diffusion** jsou založené na difuzních modelech

dopředný difuzní proces

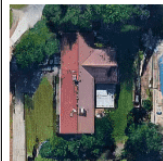
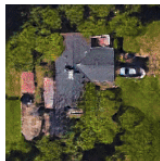
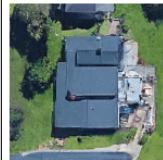


zpětný generativní proces odšumování

(Vahdat and Kreis, 2022)

porovnání GAN a difuzních modelů

- oba přístupy umí generovat velice **kvalitní** obrazové výstupy
- GAN jsou **méně** výpočetně náročné
- difuzní modely jsou **stabilnější** v generování a generují **realističtější** detaily



Aplikace generování obrazu

Aplikace GAN a difuzních modelů:

- **obraz:**

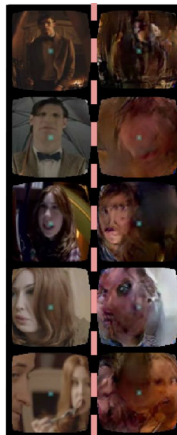
- **StyleGAN** – od Nvidia, generované snímky lidí podle atributů (držení těla, brýle, pohlaví, vlasy, ...)
- **superrezoluce** – generovaný obraz ve větším rozlišení
- **Brain2Pix** – generování obrazu toho, co vidí mozek, podle mozkové aktivity
- **Ganilla** – generování uměleckého obrazu podle fotky
- **DeepFake** – přenesení vlastností z jednoho obrazu/video do druhého

- **text:**

- **DALL-E** (OpenAI) – vytváří obrázky podle textového zadání
- **SentiGAN** – generuje text se zadaným sentimentem

- **zvuk:**

- **CereVoice Me** – vytvoří hlas podle nahrávek
- **DeepComposer** – vytváří orchestrální skladbu podle jednoduché melodie



Obsah

- 1 Generování obrazu
 - Difuzní modely
 - Aplikace generování obrazu
- 2 Generování textu
 - Jazykové modely
- 3 Asistenční modely typu ChatGPT
 - Předtrénování
 - Vyladění (fine-tuning)
 - Modelování odměn
 - Zpětnovazební učení
- 4 Metodologie tvorby otázek
 - Prompt Engineering
 - Myšlenkový postup
 - Informace z vyhledávání
- 5 PA026 – Projekt z umělé inteligence

Jazykové modely

Jazykový model

<i>Okna jsou skleněné výplně stavebních</i>	{	<i>otvorů</i>	0.08
		<i>materiálů</i>	0.05
		<i>ploch</i>	0.05
		...	
		<i>sutin</i>	0.01
		<i>spoření</i>	0.01
		...	

$$\arg \max_{w_i} P(w_i | w_1 w_2 \dots w_{i-1})$$

Jazykové modely– znalosti

Schopnost doplňování je možné využít jako **znalostní funkce** ...

Hvězdárna Brno se nachází na ... [fakt]

Tašku jsem si nechal ... práci. [syntax]

Mezi oblíbené domácí mazlíčky patří ... [téma]

Prvočíselná řada začíná čísla 2, 3, 5, 7, 11, 13, ... [aritmetika]

Výrobky této firmy prostě zbožňuju. Každý z nich je úplně ... [sentiment]

Kolotoč nemůže bez hřídele fungovat. Proto ... musíme pravidelně mazat.
[anafora]

Jazykové modely

Jaká je tedy cesta od

Hvězdárna Brno se nachází na ...



Dobrý den,
Co pro vás dnes můžu udělat?

Nápady na knihu



Nákupní seznam



Příručka začátečníka



Reklamní slogan



Sem zadejte pokyn



Obsah

- 1 Generování obrazu
 - Difuzní modely
 - Aplikace generování obrazu
- 2 Generování textu
 - Jazykové modely
- 3 **Asistenční modely typu ChatGPT**
 - Předtrénování
 - Vyladění (fine-tuning)
 - Modelování odměn
 - Zpětnovazební učení
- 4 Metodologie tvorby otázek
 - Prompt Engineering
 - Myšlenkový postup
 - Informace z vyhledávání
- 5 PA026 – Projekt z umělé inteligence

Asistenční modely typu ChatGPT

Chat Generative Pre-trained Transformer, ChatGPT

4 fáze trénování:

- předtrénování (*pre-training*)
- vyladění (*fine-tuning*)
- modelování odměn (*reward modeling*)
- zpětnovazební učení (*reinforcement learning*)

Transformer dekodér

První úspěšné NLP aplikace **transformer** modelu – **enkodér** (BERT, RoBERTa, ...) a **seq2seq** (enkodér+dekodér – BART, T5, ...)

Transformer dekodér

První úspěšné NLP aplikace **transformer** modelu – **enkodér** (BERT, RoBERTa, ...) a **seq2seq** (enkodér+dekodér – BART, T5, ...)

Zjednodušení – použití jen **dekodéru**, model **GPT**

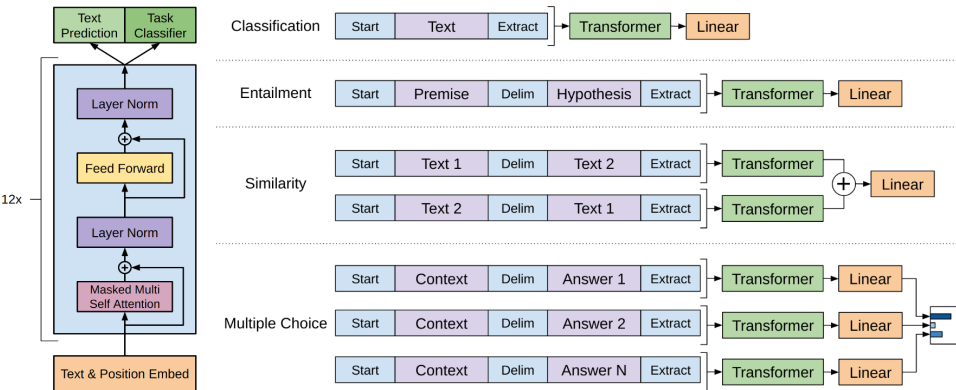
úkol

$$in_1, in_2, \dots, in_n \mapsto out_1, out_2, \dots, out_m$$

se převede na generování sekvence

$$in_1, in_2, \dots, in_n, \langle sep \rangle, out_1, out_2, \dots, out_m$$

Generative Pre-trained Transformer



(Radford et al, 2018), GPT-1

poslední vrstvy (lineární, softmax):

- **generování** (*Text Prediction*) = **předtrénování** (následující slovo)
- **klasifikace** (*Task Classifier*) = **vyladění** (třída podle úlohy)

Předtrénování

data pro trénování:

- dokumenty z internetu (včetně kódů)
 - velké množství, nízká kvalita
- doplněné o vybrané kolekce
 - knihy, učebnice
- převedené na pod-slovní jednotky (*tokens*, algoritmus Byte Pair Encoding) \mapsto čísla

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Předtrénování

Provozovat tratě jen v nezávislé trakci je výhodné tam, kde jsou slabší přepravní proudy a jezdí zde méně vlaků s menší obsazeností nebo nízkou tonáží. Zavádění elektrického pohonu je totiž spojeno s nemalými investičními náklady a zvýšenými provozními náklady.



Provozovat tratě jen v nezávislé trakci je výhodné tam, kde jsou slabší přepravní proudy a jezdí zde méně vlaků s menší obsazeností nebo nízkou tonáží. Zavádění elektrického pohonu je totiž spojeno s nemalými investičními náklady a zvýšenými provozními náklady.



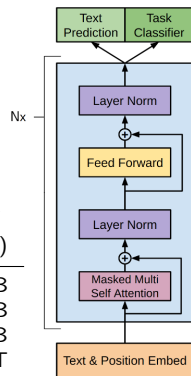
261 znaků, 144 tokenů (GPT-3)

token IDs = [15946, 8590, 709, 265, 491, 265, 128, 249, 474, 268, 410, 497, 89, 6557, 85, 3044, 2634, 1291, 74, 979, 11223, 410, 127, 121, 2065, 77, 2634, 21885, 11, 479, 2934, 44804, 280, 38677, 32790, 8836, 279, 129, 247, 538, 4108, 77, 8836, 6613, 88, 257, 474, 8471, 67, 8836, 1976, 2934, 285, 35942, 128, 249, 410, 75, 461, 129, 107, 264, 1450, 32790, 8836, 10201, 1031, 268, 455, 8836, ..., 899, 8590, 77, 8836, 11632, 299, 6557, 41582, 4597, 13]

Předtrénování

velikosti modelů:

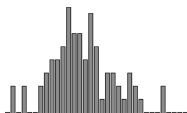
	počet parametrů	počet vrstev	dimenze vektorů	počet hlav	vstupní kontext	trénovací data (tok.)
GPT-1	117 M	12	768	12	512	20 B
GPT-2	1.5 B	48	1600	12	1024	300 B
GPT-3	175 B	96	12288	12	2048	500 B
GPT-4	1.8 T	120	20000	12	32768	13 T



Předtrénování

trénování:

- klasický jazykový model – predikuje **následující slovo**
- využívá k tomu **skrytou reprezentaci** (výstup n -té vrstvy dekodéru):
 - posledního slova – **hladové** (*greedy*) generování
 - posledních b slov – **paprskové** (*beam*) hledání
- reprezentace je vypočítaná ze **všech předchozích slov**



50 257 čísel (pravděpodobnosti dalšího tokenu)
pro trénování, na tokenu 581/we: 621/do

Transformer dekodér

4827	665	581	621	395	481	30
What	can	we	do	for	you	?

Předtrénování

Trénovací data (Shakespeare)

First Citizen:

We cannot, sir, we are undone already.

MENENIUS:

I tell you, friends, most charitable care

Have the patricians of you. For your wants,

Your suffering in this dearth, you may as well

Strike at the heaven with your staves as lift them

Předtrénování

Trénovací data (Shakespeare)

First Citizen:

We cannot, sir, we are undone already.

MENENIUS:

I tell you, friends, most charitable care

Have the patricians of you. For your wants,

Your suffering in this dearth, you may as well

Strike at the heaven with your staves as lift them

Generování po inicializaci

```
z'v}yy_RMV(7ea AOCEi2tfEi lermh'  
'88]gLNSSx|6Mj"i1wdcf,  
WezVII<4x?OBHS7D-}.8wCkGFgB(KC-  
h`Ywa.QhjPo,3C.dA!3;_)!AKa.eOMI  
Iz(DqAfE8.}nm32<Z2ma1,6DAP  
xOrA"jA[V;yhD]<g?BjKXbuptt|W:RT8,ti"  
'mnGs]MG8saNr3"u7tAftthQBt
```

Předtrénování

Trénovací data (Shakespeare)

First Citizen:

We cannot, sir, we are undone already.

MENENIUS:

I tell you, friends, most charitable care

Have the patricians of you. For your wants,

Your suffering in this dearth, you may as well

Strike at the heaven with your staves as lift them

Generování po inicializaci

```
z'v}yy_RMV(7ea AOCEi2tfEi lermh'  
'88]gLNSSx|6Mj"i1wdcf,  
WezVII<4x?OBHS7D-}.8wCkGFgB(KC-  
h`Ywa.QhjPo,3C.dA!3;_)!AKa.eOMI  
Iz(DqAfE8.}nm32<Z2ma1,6DAP  
xOrA"jA[V;yhD]<g?BjKXbuptt|W:RT8,ti"  
'mnGs]MG8saNr3"u7tAftthQBt
```

Generování po 250 krocích trénování

ONom hende beer'TIAFRO.

Rome thecoramerert BENRABENBUR. Nore se. he

lloed hears hy pid gof wiere the the paron deread boan:

ins wtherk hof at f o otherira coust Soot, Hyou

sealler sheron mer w f shathe thatchie anden wer by

he thew bat

Předtrénování

Trénovací data (Shakespeare)

First Citizen:

We cannot, sir, we are undone already.

MENENIUS:

I tell you, friends, most charitable care

Have the patricians of you. For your wants,

Your suffering in this dearth, you may as well

Strike at the heaven with your staves as lift them

Generování po inicializaci

```
z'v}yy_RMV(7ea AOCEi2tfEi lermh'
'88]gLNSSx|6Mj"i1wdcf,
WezVII<4x?OBHS7D-}.8wCkGFgB(KC-
h`Ywa.QhjPo,3C.dA!3;_]!AKa.eOMI
Iz(DqAfE8.}nm32<Z2ma1,6DAP
xOrA"jA[V;yhD]<g?BjKXbuptt|W:RT8,ti"
'mnGs]MG8saNr3"u7tAftthQBt
```

Generování po 250 krocích trénování

ONom hende beer'TIAFRO.

Rome thecoramerert BENRABENBUR. Nore se. he

lloed hears hy pid gof wiere the the paron deread boan:

ins wtherk hof at f o otherira coust Soot, Hyou

sealler sheron mer w f shathe thatchie anden wer by

he thew bat

Generování po 500 krocích trénování

For but te aser if the coouldlavilcoon Creater?

RANTEBR. In fease. Youll doverss, your fill will welt

yxether Ind comestand ins, therk hop at far on trimle

OnD Sould; maringeed her sheron mertsef andeand

datke foard and, bule thise and meardest mor your Or,

Předtrénování

Trénovací data (Shakespeare)

First Citizen:

We cannot, sir, we are undone already.

MENENIUS:

I tell you, friends, most charitable care

Have the patricians of you. For your wants,

Your suffering in this dearth, you may as well

Strike at the heaven with your staves as lift them

Generování po 250 krocích trénování

ONom hende beer'TIAFRO.

Rome thecoramerert BENRABENBUR. Nore se. he

lloed hears hy pid gof wiere the the paron deraed boan:

ins wtherk hof at f o otherira coust Soot, Hyou

sealler sheron mer w f shathe thatchie anden wer by

he thew bat

Generování po 500 krocích trénování

For but te aser if the coouldlavlilcoon Creater?

RANTEBR. In fease. Youll doverss, your fill will welt

yxether Ind comestand ins, therk hop at far on trimle

OnD Sould; maringeed her sheron mertsef andeand

datke foard and, bule thise and meardest mor your Or,

Generování po inicializaci

z'v}yy_RMV(7ea AOCEi2tfEi lermh'

'88]gLNSSx|6Mj"i1wdcf,

WezVII<4x?OBHS7D-}.8wCkGFgB(KC-

h`Ywa.QhjPo,3C.dA!3;_]!AKa.eOMI

Iz(DqAfE8.}nm32<Z2ma1,6DAP

xOrA"jA[V;yhD]<g?BjKXbuptt|W:RT8,ti"

'mnGs]MG8saNr3"u7tAftthQBT

Generování po 5 000 krocích trénování

Hor. I have been me, thereof my life, and he concludes him.

These offended his soul mine of a form that country,

And he any instruction of an have, convention'd a heart,

Caius, her charges, by affraighted daughterly de-

Předtrénování

Trénovací data (Shakespeare)

First Citizen:
 We cannot, sir, we are undone already.
 MENENIUS:
 I tell you, friends, most charitable care
 Have the patricians of you. For your wants,
 Your suffering in this dearth, you may as well
 Strike at the heaven with your staves as lift them

Generování po 250 krocích trénování

ONom hende beer'TIAFRO.
 Rome thecoramerert BENRABENBUR. Nore se. he
 llod hears hy pid gof wiere the the paron deraed boan:
 ins wtherk hof at f o otherira coust Soot, Hyou
 seealler sheron mer w f shathe thatchie anden wer by
 he thew bat

Generování po 500 krocích trénování

For but te aser if the coouldlavilcoon Creater?
 RANTEBR. In fease. Youll dovers, your fill will welt
 yetther Ind comestand ins, therk hop at far on trimle
 Ond Sould; maringeed her sheron mertsef andeand
 datke foard and, bule thise and meardest mor your Or,

Generování po inicializaci

z'v}yy_RMV(7ea AOCEi2tfEi lermh'
 '88]gLNSSx|6Mj"i1wdcf,
 WezVII<4x?OBHS7D-}.8wCkGFgB(KC-
 h`Ywa.QhjPo,3C.dA!3;_]!AKa.eOMI
 lz(DqAfE8.}nm32<Z2ma1,6DAP
 xOrA"jA[V;yhD]<g?BjKXbuptt|W:RT8,ti"
 'mnGs]MG8saNr3"u7tAftthQBt

Generování po 5 000 krocích trénování

Hor. I have been me, thereof my life, and he concludes him.
 These offended his soul mine of a form that country,
 And he any instruction of an have, convention'd a heart,
 Caius, her charges, by affraided daughterly de-

Generování po 30 000 krocích trénování

Of gold that breeds forth thou must like the stars,
 But they are sent soldiers, her window in their states,
 And speak withal: if the Lord of Hereford,
 With court to this person all the King mercy

(GPT from scratch, NYT, 2023)

Předtrénovaný model není asistent

- předtrénovaný/**základní** (*foundation/base*) **model neodpovídá** na otázky
- pouze **pokračuje** v započatém dokumentu

Write a tagline for an icecream shop.

Write a description for your favorite restaurant CHEVY'S, include food that has each of their five senses. Do mention the price range and atmosphere. Soft background music, good lighting. Describe the place and pictures in words. Write a catchy title for a newspaper ad.

Předtrénovaný model není asistent

- předtrénovaný/**základní** (*foundation/base*) **model neodpovídá** na otázky
- pouze **pokračuje** v započatém dokumentu

Write a tagline for an icecream shop.

Write a description for your favorite restaurant CHEVY'S, include food that has each of their five senses. Do mention the price range and atmosphere. Soft background music, good lighting. Describe the place and pictures in words. Write a catchy title for a newspaper ad.

trik na generování odpovědi:

Here is a tagline for an icecream shop:

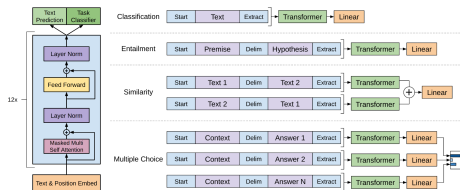
"How 'bout a treat?"

The name needs to create a strange sense of mystery.

"How 'bout a treat?" is very strange and makes the reader think what kind of treat exactly. The strategy is working because . . .

Vyladění (fine-tuning)

- 2.fáze vývoje **asistenčního** modelu
- základní model vyladěný pro úlohu **odpovídání na otázky**
- ručně anotovaná **sada** pro vyladění:
 - **ideální odpovědi** na otázky
 - **vysoká kvalita**
 - **malý rozsah** desítky tisíc otázek a odpovědí



Can you write a short introduction about the relevance of the term "monopsony" in economics? Please use examples related to potential monopsonies in the labour market and cite relevant research.

"Monopsony" refers to a market structure where there is only one buyer for a particular good or service. In economics, this term is particularly relevant in the labor market, where a monopsony employer has significant power over the wages and working conditions of their employees. The presence of a monopsony can result in lower wages and reduced employment opportunities for workers, as the employer has little incentive to increase wages or provide better working conditions. Recent research has identified potential monopsonies in industries such as retail and fast food, where a few large companies control a significant portion of the market (Bivens & Mishel, 2013). In these industries, ... Further research is needed to understand the extent and impact of monopsonies on the economy and to develop policies to address this issue. References: Bivens, J., & Mishel, L. (2013). The Pay of Corporate Executives and Financial Professionals as Evidence of Rents in Top 1 Percent Incomes. *Journal of Economic Perspectives*, 27(3), 57-78.

Modelování odměn

- 3.fáze vývoje **asistenčního** modelu
- anotátoři **porovnají** kvalitu odpovědí

Napiš Python funkci, která zkontroluje, jestli je zadaný řetězec palindrom.

Samozřejmě, můžu ti s tím pomoci. Zde je jednoduchá Python funkce, která zkontroluje, zda je zadaný řetězec palindrom:

```
def je_palindrom(retezec):
    retezec = retezec.lower() # Převodíme všechny písmena na malá
    cisty_retezec = "".join(c for c in retezec if c.isalnum())
    return cisty_retezec == cisty_retezec[::-1] # Porovnání řetězce se svým obrácením

# Testování
retezec = input("Zadej řetězec: ")
if je_palindrom(retezec):
    print("Zadaný řetězec je palindrom.")
else:
    print("Zadaný řetězec není palindrom.")
```

Tato funkce nejprve převede všechna písmena v řetězci na malá písmena a odstraní veškeré nealphanumerické znaky. Poté porovná upravený řetězec se svým obrácením a vrátí True, pokud jsou stejné, jinak vrátí False.

Napiš Python funkci, která zkontroluje, jestli je zadaný řetězec palindrom.

Samozřejmě! Zde je jednoduchá Python funkce, která zkontroluje, zda je zadaný řetězec palindrom:

```
def je_palindrom(retezec):
    retezec = retezec.lower() # Převodíme všechny písmena na malá
    return retezec == retezec[::-1] # Porovnání řetězce se svým obrácením

# Příklad použití
vstup = input("Zadejte řetězec: ")
if je_palindrom(vstup):
    print("Zadaný řetězec je palindrom.")
else:
    print("Zadaný řetězec není palindrom.")
```

Tato funkce nejprve převede vstupní řetězec na malá písmena pomocí metody lower(), aby nedošlo ke zvýšení nebo snížení písmen, což by mohlo vést k nesprávným výsledkům. Poté porovná řetězec se svou reverzí pomocí operátoru řezu [::-1]. Funkce vrátí True, pokud je řetězec palindromem, a False, pokud není.

Napiš Python funkci, která zkontroluje, jestli je zadaný řetězec palindrom.

Jistě, zde je jednoduchá Python funkce, která zkontroluje, zda je zadaný řetězec palindrom:

```
def je_palindrom(retezec):
    # Převodíme řetězec na malá písmena a odstraníme mezery
    retezec = retezec.lower().replace(" ", "")

    # Porovnání řetězce s jeho obrácením
    return retezec == retezec[::-1]

# Příklad použití
print(je_palindrom("radar")) # True
print(je_palindrom("A man a plan a canal Panama")) # True
print(je_palindrom("python")) # False
```

Tato funkce nebere v úvahu interpunkční znaménka nebo mezerové znaky při kontrolním procesu.

Modelování odměn

- 3.fáze vývoje **asistenčního** modelu
- anotátoři **porovnají kvalitu** odpovědí

Napiš Python funkci, která zkontroluje, jestli je zadaný řetězec palindrom.

Samozřejmě, můžu ti s tím pomoci. Zde je jednoduchá Python funkce, která zkontroluje, zda je zadaný řetězec palindrom:

```
def je_palindrom(retezec):
    retezec = retezec.lower() # Převodíme všechna písmena na malá
    cisty_retezec = ''.join(c for c in retezec if c.isalnum()) # Odstraníme
    return cisty_retezec == cisty_retezec[::-1] # Porovnání řetězce se svou reverzí

# Testování
retezec = input("Zadej řetězec: ")
if je_palindrom(retezec):
    print("Zadaný řetězec je palindrom.")
else:
    print("Zadaný řetězec není palindrom.")
```

Tato funkce nejprve převede všechna písmena v řetězci na malá písmena a odstraní veškeré nealphanumerické znaky. Poté porovná upravený řetězec se svým obrácením a vrátí True, pokud jsou stejné, jinak vrátí False.

Napiš Python funkci, která zkontroluje, jestli je zadaný řetězec palindrom.

Samozřejmě! Zde je jednoduchá Python funkce, která zkontroluje, zda je zadaný řetězec palindrom:

```
def je_palindrom(retezec):
    retezec = retezec.lower() # Převodíme všechna písmena na malá
    return retezec == retezec[::-1] # Porovnání řetězce se svou reverzí

# Příklad použití
vstup = input("Zadejte řetězec: ")
if je_palindrom(vstup):
    print("Zadaný řetězec je palindrom.")
else:
    print("Zadaný řetězec není palindrom.")
```

Tato funkce nejprve převede vstupní řetězec na malá písmena pomocí metody lower(), aby nedošlo ke zvýšení nebo snížení písmen, což by mohlo vést k nesprávným výsledkům. Poté porovná řetězec se svou reverzí pomocí operátoru řezu [::-1]. Funkce vrátí True, pokud je řetězec palindromem, a False, pokud není.

Napiš Python funkci, která zkontroluje, jestli je zadaný řetězec palindrom.

Jistě, zde je jednoduchá Python funkce, která zkontroluje, zda je zadaný řetězec palindrom:

```
def je_palindrom(retezec):
    # Převodění řetězce na malá písmena a odstranění mezerníků a interpunkčních znamének
    retezec = retezec.lower().replace(" ", "").replace(".", "").replace(",", "")

    # Porovnání řetězce se svou reverzí
    return retezec == retezec[::-1]

# Příklad použití
print(je_palindrom("radar")) # True
print(je_palindrom("A man a plan a canal Panama")) # True
print(je_palindrom("python")) # False
```

Tato funkce nebere v úvahu interpunkční znaménka nebo mezerové znaky při kontrolním procesu.

Modelování odměn

- další vyladění modelu
- vstup: otázka, odpověď a speciální token |odměna|
- trénuje se pouze výstup na tokenu |odměna|
- speciální **trénovací funkce** – reprodukuje shodu mezi navrženou odměnou a pořadím od anotátorů
- výsledek = model schopný měřit kvalitu odpovědi

otázka 	odpověď 1 odměna	→ 0.2
otázka 	odpověď 2 odměna	→ 1.2
otázka 	odpověď 3 ... odměna	→ -0.5

Zpětnovazební učení

Reinforcement Learning with Human Feedback, RLHF

- 4.fáze vývoje **asistenčního** modelu
- **úprava vah** pro generování vyladěného modelu z fáze 2
- generuje odpovědi, které **maximalizují** odměnu
- vstup: **otázka**
- vyladěný model generuje **odpověď**
- model pro měření kvality odpovědi (fáze 3) **stanoví** odměnu
- trénuje se **pouze** část **odpověď**
- pravděpodobnosti slov odpovědí jsou **vážené** normalizovanou **odměnou**
- výsledek = **asistenční model** typu ChatGPT

otázka |...| |...|

odpověď 1 |...| |...| |odměna|

skóre **0.2**

otázka |...| |...|

odpověď 2 |...| |...| |...| |odměna|

skóre **1.0**

otázka |...| |...|

odpověď 3 |...| |odměna|

skóre **-1.2**

Zpětnovazební učení

proč je potřeba **RLHF** a nestačí **vyladění**?

- RLHF **funguje lépe**
- **porovnávat** kvalitu je **jednodušší** než kvalitně **generovat**

Vytvoř rýmovanou básničku o koťatech

V zahradě se koťátka,
honí kolem vrátka.
Pruhované, strakaté,
všechny trochu neposedné.

Jedno skáče na lavičku,
druhé honí svoji myšku.
Třetí spí pod sluníčkem,
vrtí ouškem, pije mlíčko.

Vytvoř rýmovanou básničku o koťatech

Kořata si hrají v trávě,
skákají, běhají, nic je nezastaví.
Malé tlapičky, ocásky se třepotají,
a v očích jim jiskřičky veselí září.

Srdíčka jak z hebkého peří,
mňoukají tiše, když se vzbudí z postele.
Líně se protahují, pak zas běhají,
v očích tajemství, co jen kořata znají.

Vytvoř rýmovanou básničku o koťatech

V rohu světa, kde je klid,
spí si koťe, sladký cit.
Tlapičky má jako z mechu,
v kožišku má vůni dechu.

Očka jako drahokamy,
čertík v něm i klidné rány.
Společnost ti dělá rád,
věrný je jak žádný zvířecí kamarád.

Asistenční modely

Rank* (UB) ▲	Rank (StyleCtrl) ▲	Model ▲	Arena Score ▲	95% CI ▲	Votes ▲	Organization ▲	License ▲
1	1	ChatGPT-4o-latest (2024-11-20)	1366	+3/-4	19888	OpenAI	Proprietary
1	3	Gemini-Exp-1121	1364	+7/-4	13583	Google	Proprietary
3	5	Gemini-Exp-1114	1347	+5/-4	16208	Google	Proprietary
4	2	o1-preview	1334	+4/-4	29747	OpenAI	Proprietary
5	7	o1-mini	1307	+3/-4	35691	OpenAI	Proprietary
5	5	Gemini-1.5-Pro-002	1301	+3/-3	31240	Google	Proprietary
7	10	Grok-2-08-13	1289	+3/-3	54702	xAI	Proprietary
7	12	Yi-Lightning	1287	+3/-4	29201	01 AI	Proprietary
7	6	GPT-4o-2024-05-13	1285	+2/-2	115598	OpenAI	Proprietary
7	3	Claude 3.5 Sonnet (20241022)	1283	+3/-3	33295	Anthropic	Proprietary
10	16	Athene-v2-Chat-72B	1276	+5/-5	7159	NexusFlow	NexusFlow

<https://lmarena.ai/?leaderboard>, 4.12.2024

Obsah

- 1 Generování obrazu
 - Difuzní modely
 - Aplikace generování obrazu
- 2 Generování textu
 - Jazykové modely
- 3 Asistenční modely typu ChatGPT
 - Předtrénování
 - Vyladění (fine-tuning)
 - Modelování odměn
 - Zpětnovazební učení
- 4 Metodologie tvorby otázek
 - Prompt Engineering
 - Myšlenkový postup
 - Informace z vyhledávání
- 5 PA026 – Projekt z umělé inteligence

Prompt Engineering

- od určité velikosti modelu (cca GPT-2) \mapsto **úprava otázky** (*prompt*) může **nahradit vyladění bez příkladů** (*zero-shot*)

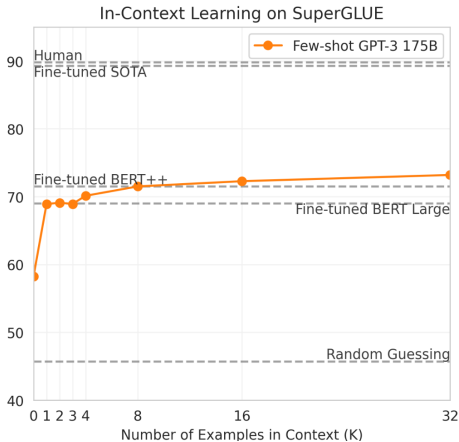
Translate English to French:
cheese =>

jeden příklad (*one-shot*)

Translate English to French:
sea otter => loutre de mer
cheese =>

málo příkladů (*few-shot*)

Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe
peluche cheese =>



(Brown et al, 2020)

Jak efektivně tvořit otázku?

metodologie tvorby kontextu a otázky = **promptové inženýrství** (*prompt engineering*)

- uvedení **příkladů** (*few-shot learning*)
- doplnění **detailů**
- **myšlenkový postup** (*chain of thought*)
- informace z **vyhledávání** (*retrieval-augmented generation, RAG*)

Doplnění detailů

- model negeneruje **správné** odpovědi, ale odpovědi z **trénování**
- když chceme **správnou** odpověď, musíme si o ni říct:
 - „Jsi přední odborník na ...“
 - „Máš IQ 130.“
 - „Ujisti se, že máš **správnou odpověď**.“
 - „Postupuj logicky.“
- detailní otázka s kontextem má obsahovat prvky:
 - **role** odpovídajícího
 - **cíl** otázky
 - **posluchači** – komu je odpověď určena
 - **tématické** zaměření – styl, tón, ...
 - **struktura** a formát odpovědi

Slido

Jsi učitel dějepisu na střední škole. Navrhni pro žáky 8.třídy poutavou aktivitu ve třídě týkající se vytváření společné digitální časové osy. Aktivita by měla zahrnovat vyprávění příběhů a technologie. Časová dotace je 3–4 vyučovací hodiny (každá 45 minut). Formátuj odpověď jako osnovu.

Myšlenkový postup

Slon africký je ...**1,5krát**... těžší než slon indický

jak postupuje **člověk**?

- **rozhodne** se porovnat váhu slona afrického a indického
- **podívá** se na Wikipedii – slon africký **6 000 kg**, slon indický **4 000 kg**
- **spočítá** poměr $6\,000/4\,000 = 1,5$
- **vyjádří** v textu, **formuluje**

jak postupuje jazykový **model**?

- **generuje** slova jedno po druhém
- na každém slově tráví **stejný čas**
- **nekontroluje** správnost, jen **vztahy** mezi slovy
- využívá obrovskou **paměť** těchto vztahů

Myšlenkový postup

chain of thought:

- model potřebuje **víc** tokenů, aby mohl “přemýšlet”
- v otázce **požádat**, aby v odpovědi postupoval **krok za krokem**
- výsledek je pak **přesnější**

Na zastávce z autobusu vystoupilo 44 cestujících. Po nástupu dalších cestujících přibylo v autobusu 13 cestujících. Kolik cestujících nastoupilo na zastávce?

Počet cestujících, kteří nastoupili na zastávce, lze vypočítat odečtením počtu cestujících, kteří vystoupili, od celkového počtu cestujících na zastávce po příchodu.

Počet cestujících, kteří nastoupili = $44 + 13 - 44 = 13$

Takže na zastávce nastoupilo 13 cestujících.

Myšlenkový postup

chain of thought:

- model potřebuje **víc** tokenů, aby mohl “přemýšlet”
- v otázce **požádat**, aby v odpovědi postupoval **krok za krokem**
- výsledek je pak **přesnější**

Na zastávce z autobusu vystoupilo 44 cestujících. Po nástupu dalších cestujících přibylo v autobusu 13 cestujících. Kolik cestujících nastoupilo na zastávce?

Počet cestujících, kteří nastoupili na zastávce, lze vypočítat odečtením počtu cestujících, kteří vystoupili, od celkového počtu cestujících na zastávce po příchodu.

$$\text{Počet cestujících, kteří nastoupili} = 44 + 13 - 44 = 13$$

Takže na zastávce nastoupilo 13 cestujících.

Na zastávce z autobusu vystoupilo 44 cestujících. Po nástupu dalších cestujících přibylo v autobusu 13 cestujících. Kolik cestujících nastoupilo na zastávce? **Vyřeš správně slovní úlohu. Postupuj přitom krok za krokem.**

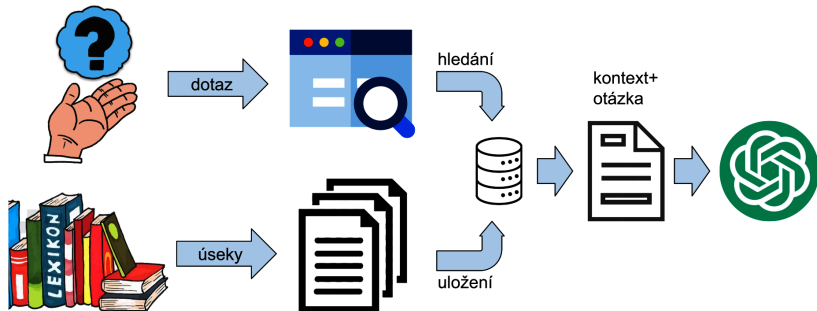
1. Po vystoupení cestujících klesl počet cestujících v autobusu o 44.
 2. Po nástupu nových cestujících přibylo v autobusu 13 cestujících.
 3. Takže na zastávce muselo nastoupit o 44 + 13 cestujících více, než kolik jich vystoupilo.
 4. Výsledný počet cestujících, kteří na zastávce nastoupili, tedy vypočítáme sčítáním: $44 + 13 = 57$.
- Odpověď: Nastoupilo 57 cestujících.

Informace z vyhledávání

někdy je potřeba **vyhledávat** odpovědi z **databáze** nebo sady dokumentů

Retrieval-Augmented Generation, RAG

- **rozdělte** dokumenty do krátkých úseků textu
- uložte vektorové reprezentace úseků do **vektorové databáze** (např. pomocí **LlamaIndex**)
- po dotazu **vyhledejte** text v DB podle **shody s vektorem** otázky
- **přidejte** vyhledaný úsek textu jako **kontext** otázky



Metodologie tvorby otázek – doporučení

- zadávejte **detailní otázky** s kontextem, relevantními informacemi a instrukcemi
- relevantní informace můžete **dohledat** podle dotazu
- **experimentujte** s různými formami otázky
- **základní** model má vyšší perplexitu než asistenční model, generuje **rozmanitější** texty
- přemýšlejte o **etice** – zaujatost dat (*bias*), možné zneužití pro šíření dezinformací nebo škodlivého obsahu
- odpovědi vždy **ověřujte**

Obsah

- 1 Generování obrazu
 - Difuzní modely
 - Aplikace generování obrazu
- 2 Generování textu
 - Jazykové modely
- 3 Asistenční modely typu ChatGPT
 - Předtrénování
 - Vyladění (fine-tuning)
 - Modelování odměn
 - Zpětnovazební učení
- 4 Metodologie tvorby otázek
 - Prompt Engineering
 - Myšlenkový postup
 - Informace z vyhledávání
- 5 PA026 – Projekt z umělé inteligence

PA026 – Projekt z umělé inteligence

- **navazuje** na předmět *PB016 Úvod do umělé inteligence*
- volba programovacího jazyka **není omezena**
- **samostatná** volba **tématu** v rozsahu ≥ 1 semestru
- předmět probíhá jako **prezentace** a **konzultace**
- zajímavé výsledky (<http://nlp.fi.muni.cz/aiproject/>)
 - projekt **elnet** – > 5 let spolupráce na grantových projektech simulace elektrorozvodných sítí
 - projekt **plagiaty_z_webu** – vyhledávání shod s dokumenty na celém webu
 - projekt **robot_johnny_5** – sestavení a “oživení” robota – mobilního počítače
 - robot **Karel Pepper** – <https://nlp.fi.muni.cz/projects/pepper>

