FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Bakalářská práce

Využití AI při tvorbě no-code a low-code aplikací

Studijní program: B0714A270001 Mechatronika
Autor práce: **Jan Kluz**
Vedoucí práce: Ing. Roman Špánek, Ph.D.
Ústav mechatroniky a technické informatiky

Liberec 2024

FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Zadání bakalářské práce

Využití AI při tvorbě no-code a low-code aplikací

Jméno a příjmení:	**Jan Kluz**
Osobní číslo:	M21000033
Studijní program:	B0714A270001 Mechatronika
Zadávající katedra:	Ústav mechatroniky a technické informatiky
Akademický rok:	2023/2024

Zásady pro vypracování:

1. Seznamte se s tématy velkých jazykových modelů, základy neuronových sítí, architekturou transformerů.
2. Nastudujte současný stav vývoje vybraných nástrojů souhrnně označovaných jako AI, jako jsou například: ChatGPT, Google Bard, Microsoft Azure chatbot, HuggingChat, GitHub Copilot a dalších.
3. Připravte detailní srovnání výhod a nevýhod využití AI při tvorbě aplikací za pomoci low-code či no-code.
4. Připravte srovnání aktuálních možností návrhu vlastního řešení založeného na AI a to i bez nutnosti napojení na serverové řešení třetích stran.
5. Připravte pilotní implementaci komplexního chatbota založeného na umělé inteligenci, který využije velké jazykové modely, a to pomocí softwaru umožňujícího low-code a no-code aplikace.

Rozsah grafických prací:	dle potřeby dokumentace
Rozsah pracovní zprávy:	30 až 40 stran
Forma zpracování práce:	tištěná/elektronická
Jazyk práce:	čeština

Seznam odborné literatury:

[1] PAVLÍČEK, Antonín a SYROVÁTKOVÁ, Jana. Základy moderní informatiky. [Průhonice]: Professional Publishing, 2022. ISBN 978-80-88260-59-2.

[2] SHNEIDERMAN, Ben. Human-centered AI. Oxford: Oxford University Press, [2022]. ISBN 978-0-19-284529-0.

[3] PAN, Chao. Deep learning fundamentals: an introduction for beginners. [Wilmington, Delaware]: AI Sciences, 2018. ISBN 978-1-7212-3088-4.

Vedoucí práce:	Ing. Roman Špánek, Ph.D. Ústav mechatroniky a technické informatiky
Datum zadání práce:	12. října 2023
Předpokládaný termín odevzdání:	14. května 2024

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

doc. Ing. Josef Černohorský, Ph.D.
vedoucí ústavu

V Liberci dne 12. října 2023

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

25. března 2024 Jan Kluz

Využití AI při tvorbě no-code a low code aplikací

Abstrakt

Tato bakalářská práce se v teoretické části podrobně zabývá umělou inteligencí. Poskytuje detailní informace o umělé inteligenci, zahrnuje pohled do strojového učení a nabízí detailní informace o hlubokém učení, neuronových sítích a transformerech. Dále práce obsahuje rozbor a srovnání různých jazykových modelů, které jsou v současnosti dostupné na trhu. Hlavní téma práce se zaměřuje na spojení umělé inteligence s platformami pro low-code a no-code programování. V rešeršní části je zdůrazněna motivace a významnost těchto platforem, které by měly být brány v potaz. V praktické části jsou prezentovány dva hlavní úkoly: první se týká analýzy a praktické aplikace vytváření offline chatbota, který nabízí výhody umělé inteligence bez rizika úniku dat, jelikož data zůstávají uložena na vlastním počítači a není potřeba internetového připojení. Druhý úkol se věnuje vytvoření vlastního AI chatbota s využitím low-code a no-code platforem, kde byla zvolena AI persona komunikující na Instagramu pod jménem autora této práce a jako zdroj informací využívá obsah této bakalářské práce. Hlavní část práce zabývající se umělou inteligencí je realizována na platformě s asistenty od OpenAI. Komunikace mezi platformami je zprostředkována prostřednictvím Make a komunikace s Instagramem je realizována přes aplikaci ManyChat.

Klíčová slova: Umělá inteligence, velké jazykové modely, low-code a no-code programování, AI persona, transformerové architektury, ManyChat, Make.

Using AI in development of no-code and low-code applications

Abstract

This bachelor thesis provides a detailed theoretical examination of artificial intelligence. It offers in-depth insights into artificial intelligence, includes perspectives on machine learning, and provides detailed information on deep learning, neural networks, and transformers. Additionally, the thesis contains an analysis and comparison of various language models currently available in the market. The main theme of the thesis focuses on integrating artificial intelligence with platforms for low-code and no-code programming. The literature review section emphasizes the motivation and significance of these platforms, which should be considered. The practical part presents two main tasks: the first concerns the analysis and practical application of creating an offline chatbot, which offers the advantages of artificial intelligence without the risk of data leakage, as the data remain stored on the owner's computer and no internet connection is needed. The second task involves creating a custom AI chatbot using low-code no-code platforms, where an AI persona communicating on Instagram under the name of this thesis' author was chosen, and the content of this bachelor thesis is used as the information source. The main part of the thesis, dealing with artificial intelligence, is implemented on a platform with assistants from OpenAI. Communication between platforms is mediated through Make, and communication with Instagram is facilitated through the ManyChat application.

Keywords: Artificial Intelligence, Large Language Models, Low-code and No-code Programming, AI Persona, Transformer Architectures, ManyChat, Make.

Obsah

Seznam obrázků

Seznam zkratek

TUL	Technická univerzita v Liberci
FM	Fakulta mechatroniky, informatiky a mezioborových studií
AI	Artificial Intelligence
ML	Machine Learning
LLM	Large Language Models
LCNC	Low-Code No-Code
RNN	Recurrent Neural Network

Úvod

V dnešním neustále se měnícím technologickém světě je umělá inteligence (AI) na úplném vrcholu zájmu. AI se stala klíčovou součástí mnoha aplikací a systémů, které zautomatizovávají a zefektivňují procesy napříč různými obory. Tato bakalářská práce se zaměřuje na propojení umělé inteligence s platformami pro low-code a no-code programování, což představuje revoluční přístup k vývoji softwaru. Tento přístup nabízí nespočet výhod, zejména umožňuje uživatelům bez programátorských dovedností vyvíjet základní aplikace a řešení použitelná v reálném světě.

Motivace pro zvolení tohoto tématu vychází z rostoucího významu AI v moderní společnosti a jejího potenciálu radikálně změnit způsoby, jakými jsou softwarové aplikace vytvářeny a distribuovány. Cílem této práce je poskytnout ucelený přehled o integraci umělé inteligence s low-code a no-code platformami a prozkoumat, jak mohou tyto technologie spolupracovat při tvorbě nových, inovativních řešení.

Pro realizaci tohoto cíle byla práce rozdělena do dvou hlavních částí: teoretické rešerše a praktické aplikace. V teoretické části je uveden podrobný přehled základních principů AI, včetně strojového učení, hlubokého učení, neuronových sítí a architektur transformerů. Dále práce obsahuje analýzu a porovnání různých jazykových modelů, které jsou v současnosti dostupné na trhu.

Praktická část se soustředí na dva hlavní úkoly: první z nich je analýza a demonstrace vytváření offline chatbota, který nabízí výhody AI, ale eliminuje rizika spojená s únikem dat, protože data zůstávají uložena na lokálním počítači a nepotřebují internetové připojení. Druhý úkol popisuje vývoj vlastního AI chatbota s využitím low-code a no-code platforem, který komunikuje na Instagramu pod jménem autora a využívá obsah této práce jako zdroj informací.

Celková struktura práce je navržena tak, aby postupně vedla čtenáře od teoretických základů AI, přes důkladnou analýzu trhu s jazykovými modely, představení světa low-code a no-code programování, až po praktické implementace využívající tyto technologie. Každá kapitola je pečlivě strukturována tak, aby poskytla komplexní pohled na specifické aspekty tématu, což umožňuje čtenářům lépe porozumět materiálu a jeho aplikaci v reálném světě.

1 Základy umělé inteligence: Definice, klíčové koncepty a historický vývoj

Umělá inteligence - artificial intelligence (AI) - je schopnost strojů napodobovat lidské schopnosti, jako je uvažování, učení se, plánování nebo kreativita. Umělá inteligence umožňuje technickým systémům reagovat na vněmy z jejich prostředí, řešit problémy a dosahovat určitých cílů. Zabudovaný počítač přijímá data - která byla již připravena, nebo jsou sbírána pomocí vlastních senzorů a kamer - ty následně vyhodnotí a reaguje na ně. Systémy umělé inteligence jsou schopné pracovat samostatně a také měnit a přizpůsobovat své jednání na základě vyhodnocení efektů předchozích akcí.[24]

Umělá inteligence (AI) se dá považovat za klíčovou součást strojového učení (ML), specializující se na identifikaci statistických vzorců v rozsáhlých datasetech vytvořených lidmi. Velké jazykové modely (LLMs), reprezentující AI, byly vyvinuty prostřednictvím učení se na bilionech slov po několik měsíců, což vyžadovalo enormní výpočetní kapacitu. Tyto modely se odlišují od tradičního programování, které vyžaduje specifickou syntaxi a interakci s knihovnami. Na rozdíl od nich, LLMs zpracovávají přirozený jazyk ve formě promptů {5.1}, které se vkládají do takzvaného kontextového okna. Toto okno slouží jako paměť a jeho velikost, která se liší model od modelu, je jedním z hlavních faktorů pro výběr vhodného modelu pro specifické použití. Modely predikují další slova nebo písmena a reagují na dotazy očekávanými odpověďmi. Tento proces je známý jako „Completition“.

1.1 Rozdělení Umělé Inteligence: ANI versus AGI

Umělá inteligence (AI) se dělí na dvě základní kategorie, které se od sebe zásadně liší. První kategorií je tzv. ANI (Artificial Narrow Intelligence), neboli úzká umělá inteligence, která je navržena pro konkrétní úlohy nebo činnosti. V posledních letech jsme byli svědky významného pokroku v této oblasti, ať už jde o inteligentní asistenty jako Alexa nebo Siri, autonomní vozidla nebo AI systémy využívané ve výrobě a zemědělství. Tyto systémy mohou být mimořádně cenné, pokud jsou správně nasazeny ve vhodných aplikacích.

Druhá kategorie, AGI (Artificial General Intelligence), představuje vizi umělé inteligence s obecnými kognitivními schopnostmi srovnatelnými nebo převyšující-

mi lidské. AGI by bylo schopné vykonávat jakoukoliv intelektuální úlohu, kterou je schopen vykonat člověk, a představuje konečný cíl výzkumu v oblasti AI. K dosažení tohoto cíle je však ještě potřeba překonat mnoho vědeckých a technologických výzev.[1]

1.2 Historie a vývoj AI

Vývoj AI byl inspirován nedostatky rekurentních neuronových sítí (RNN), které se zabývaly pouze omezeným kontextem, což omezovalo jejich schopnost správně interpretovat význam slov a vět. Jazyk a text jsou velice komplexní; například jedno slovo může mít více významů, a proto je pro správnou interpretaci nutná znalost kontextu celé věty. Publikace „Attention is All You Need“ [31] od Googlu a Univerzity v Torontu z roku 2017 představila architekturu transformerů, která radikálně změnila způsob, jakým AI rozumí a generuje text, umožňující efektivnější zpracování dat a poskytující modelům schopnost soustředit se na relevantní části textu.

1.3 Strojové učení

Strojové učení (ML) je základním pilířem oblasti umělé inteligence (AI) a jeho esencí je schopnost učit se z dat. Nejběžnější formou strojového učení je supervizované učení, kde model zpracovává vstupní data (A) a učí se na základě těchto dat produkovat požadovaný výstup (B). Strojové učení nachází uplatnění v široké škále aplikací - od rozpoznávání řeči, přes automatické překladače, on-line reklamu, autonomní vozidla, po vizuální inspekci v průmyslových procesech. Tyto aplikace vycházejí z principu mapování vstupů na výstupy (A->B mapping), čímž model získává schopnost předpovídat nebo kategorizovat data na základě naučených vzorců. Příkladem může být filtr spamu v e-mailových schránkách, kde je model naučen rozlišovat mezi spamem a legitimními e-maily a podle toho je kategorizuje.

Rozmach strojového učení v posledních letech byl umožněn díky exponenciálnímu nárůstu dat, zvýšení dostupného výpočetního výkonu a vývoji neuronových sítí, které umožňují efektivnější a přesnější zpracování těchto dat. Velikost a složitost neuronových sítí jsou přímo úměrné jejich schopnosti naučit se z dat a dosáhnout vyšší výkonnosti. Tento pokrok vedl k rozšíření možných aplikací ML a k větší užitečnosti pro firmy i jednotlivce. Pro maximální účinnost strojového učení je nezbytné disponovat rozsáhlými datovými sadami (tzv. „big data“) a dostatečnými výpočetními zdroji pro trénink rozsáhlých neuronových sítí. [19].

1.4 Data - základní stavební prvek

Klíčovým prvek AI a tedy i ML jsou data. Data mohou představovat širokou škálu informací, od numerických hodnot, až po obrazy a texty, organizované do struktury známé jako dataset. Tyto datasety jsou základem pro trénink a vývoj AI modelů,

neboť obsahují vstupní (A) a výstupní (B) prvky, mezi kterými se model učí rozpoznávat vzorce a vztahy.

Získání kvalitních dat je klíčovým krokem pro úspěšný vývoj AI modelů. Data mohou být získávána různými metodami, včetně metody ručního označování, kde člověk ručně určí kategorie nebo hodnoty pro konkrétní prvky dat. Tento proces je však časově náročný a vyžaduje značné množství dat pro dosažení spolehlivých výsledků. Další metodou je automatické shromažďování dat z pozorování, například zákaznického chování v on-line obchodech nebo operací strojů ve výrobním procesu. Další data lze získat stahováním z internetu nebo přímo od klientů či uživatelů.

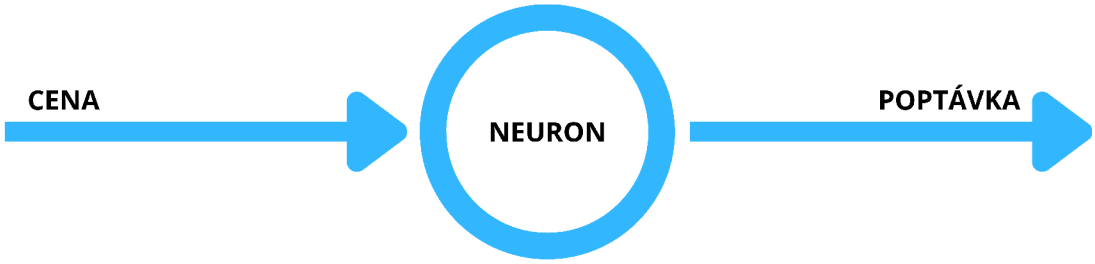
Při práci s daty je důležité zvážit jejich relevanci a kvalitu. Data často obsahují chyby, jako jsou nesmyslné nebo chybějící hodnoty, které mohou způsobit problémy při vývoji a tréninku modelů. Proto je nezbytné věnovat pozornost předzpracování dat, jejich čištění a validaci, aby se zajistila co nejvyšší kvalita datasetů použitých pro trénink AI modelů [16].

1.5 Hluboké učení a neuronové sítě

Deep learning, neboli hluboké učení, a neuronové sítě, klíčové komponenty v oblastech AI a ML, jsou inspirovány funkcí a strukturou lidského mozku. Tyto systémy se skládají z vrstev neuronů schopných zpracovávat rozsáhlá množství dat, čímž identifikují vzorce a vztahy mezi vstupy (A) a výstupy (B). Tento proces umožňuje modelování složitých vztahů a provádění přesných předpovědí.

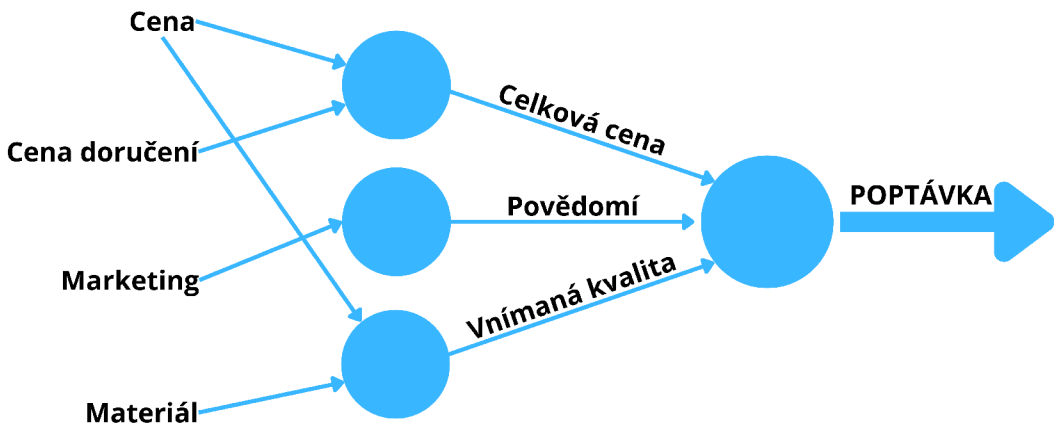
Neuronové sítě zahrnují několik vrstev: vstupní vrstvu pro příjem dat, jednu nebo více skrytých vrstev pro jejich zpracování a výstupní vrstvu pro generování výsledků. Váhové koeficienty mezi neurony určují důležitost vstupů pro výstupy.

Ilustrace funkčnosti neuronové sítě může být zřetelná na příkladu zkoumajícím vztah mezi cenou a poptávkou. V tomto modelu {1.1}, kde A reprezentuje cenu a B poptávku, je vytvořen dataset s těmito dvěma proměnnými. Do tohoto kontextu je umístěn neuron, který na základě známého vztahu mezi cenou (A) a poptávkou (B) z tréninkových dat dokáže pro nové hodnoty ceny předpovědět odpovídající úroveň poptávky.



Obrázek 1.1: Neuron

Tento umělý neuron, základní stavební jednotka neuronových sítí, vypočítává funkci reprezentující vztah mezi vstupem a výstupem. I když je zde uvedená síť sestavena pouze z jednoho neuronu, reálné neuronové sítě obsahují mnohem více neuronů spojených různými způsoby, což umožňuje zpracovávat složitější vztahy a vstupy {1.2}, jako by zde byly například náklady na dopravu, cena marketingu nebo použitý materiál, které ovlivní konečnou poptávku a i jednotlivé vztahy mezi sebou určitým způsobem, které by se tyto neuronové sítě naučily.[17]

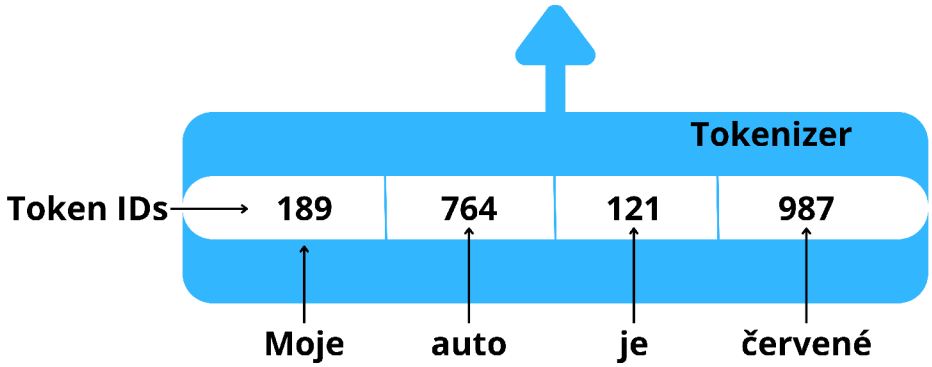


Obrázek 1.2: Neuronová síť

Tento proces učení a adaptace v neuronových sítích je klíčem k jejich schopnosti předpovídat a modelovat s vysokou přesností. I přes velký potenciál neuronových sítí je jejich efektivita omezena kvalitou a rozsahem dostupných tréninkových dat.

1.6 Architektura transformerů

Architektura transformerů je strukturována do dvou hlavních komponent: enkodéru a dekodéru. Jednou ze základních vlastností transformerů je, že dokáží pracovat výhradně s číselnými hodnotami. To představuje podstatný předpoklad pro jejich fungování, kdy je nutné převést slova na čísla, což se realizuje procesem zvaným tokenizace. Během tohoto procesu se slova nebo jejich části transformují na unikátní identifikátory, tzv. tokeny, které jsou poté reprezentovány vektorově.



Obrázek 1.3: Tokenizer

Následně jsou tyto tokeny umístěny do vektorové databáze, kde každý token získává svůj vektorový ekvivalent. Tyto vektory poté vstupují buď do enkodéru nebo dekodéru, kde je mechanismus „self-attention“ zodpovědný za přiřazení váhy jednotlivým slovům v závislosti na jejich kontextu[30].

Využití transformerů v oblasti rozsáhlých jazykových modelů představovalo významný pokrok oproti předchozím modelům založeným na Rekurentních neuronových sítích (RNN). Otevřely se nové možnosti pro umělou inteligenci, umožňující modelům chápat kontext nejen celých vět, ale klidně i celých textů, a nikoli jen vztahy mezi sousedícími slovy jako tomu bylo právě u RNN. Díky tomu začaly modely zohledňovat relevanci mezi všemi slovy v textu. Jednou z klíčových inovací transformerů je vytvoření tzv. „attention maps“, které vizualizují vzájemné vztahy a důležitost mezi slovy, umožňující modelu soustředit se na klíčové části textu, což je proces známý jako „self-attention“ [4].

Architektura transformerů s „multi-headed self-attention“ mechanismem umožňuje paralelní zpracování textu, čímž zvyšuje efektivitu a přesnost modelů. Každá „hlava“ v tomto systému se specializuje na odlišné jazykové aspekty, což modelům umožňuje dosáhnout hlubšího porozumění textu. Výstupy z těchto procesů jsou pak transformovány do softmax vrstvy, která určuje pravděpodobnost následujícího slova v sekvenci, což umožňuje generování soudržného a relevantního textu [13].

Tato pokročilá struktura a metody posunuly možnosti velkých jazykových modelů daleko za hranice původních omezení a poskytly nový rozměr tomu, co je umělá inteligence schopna dosáhnout.

V rámci transformerů může být architektura upravena pro specifické účely, obsahující buď pouze enkodér, dekodér, nebo oboje. Model jako například BERT, který využívá pouze enkodér, se ukazuje jako vhodný pro úlohy vyžadující hluboké porozumění kontextu vstupu, jako je analýza sentimentu nebo extrakce informací. Enkodér zpracovává vstupní text a jeho kontext do vektorové reprezentace, zatímco dekodér je odpovědný za generování výstupu na základě této analýzy [30].

1.6.1 Příklad fungování transformeru

Jako ilustrační příklad, jak transformery fungují v praxi, můžeme uvést překlad věty „Mám rád strojové učení“ do angličtiny. Proces začíná tokenizací vstupního textu, kde každé slovo je převedeno na token pomocí stejného tokenizéru, který byl použit během trénování modelu. Například, spojení „mám rád“ může být převedeno na token [1234], kde „1234“ je unikátní identifikátor tohoto slova ve vektorové databázi modelu. Tento identifikátor nese v sobě informaci nejen o slově samotném, ale také o jeho významu a použití v různých kontextech. Každý token je reprezentován vektorem, který je výsledkem trénování modelu na obrovských textových datasetech. Tyto vektory jsou umístěny ve vektorové databázi, kde každý vektor může být představován jako bod v mnohorozměrném prostoru. V tomto prostoru mohou být osy reprezentovány různými lingvistickými charakteristikami slova, jako je syntaxe, sémantika nebo kontextová příslušnost. V našem příkladu může bod pro token [1234] („mám rád“) být umístěn na ose X, která představuje hodnotu pro dané spojení, a na ose Y je reprezentace pro náš úkol, tedy překlad - spojením těchto dvou hodnot poté dostaneme vektor [XXXX], který po tokenizaci představuje spojení „I love“, ještě před tím je ale vektor pro token „1234“ vstupem pro enkodér. V enkodéru jsou jednotlivým slovům přiřazeny váhy, které odrážejí jejich důležitost v kontextu celé věty. Tato vážená slova jsou poté předána do dekodéru.

Dekodér na základě počátečního signálu začíná sekvenčně předpovídat další slova ve větě, vždy s ohledem na kontext poskytnutý enkodérem. Pro výběr slov může být použita metoda „greedy selection“, která volí slovo s nejvyšší pravděpodobností, nebo metoda „random weighted sampling“, která vybírá slova na základě jejich pravděpodobnostní váhy, což může přinést více variant do překladu a zamezit opakování stejných frází. Po dokončení generování sekvenčně vybraných tokenů nastává proces detokenizace, což vede k finálnímu překladu věty „I love machine learning“.

Tento příklad ukazuje, jak transformery zpracovávají a překládají text a demonstrující jejich schopnost učit se z kontextu a efektivně predikovat další slova. Proces zahrnuje složité kroky jako tokenizaci, vážení slov, predikci na základě kontextu

a nakonec detokenizaci, což ilustruje vyspělost a komplexnost transformerů v aplikaci překladu textu[30].

1.7 Velké jazykové modely (LLMs)

LLMs jsou to, co v dnešní době známe jako například ChatGPT. Tento typ umělé inteligence, který je reprezentován rozhraním, se kterým můžeme interagovat, umožňuje široké spektrum aplikací od psaní esejí, přes vytváření shrnutí, překlady, psaní kódu, až po vyhledávání specifických informací v textech.

1.7.1 Trénovaní velkých jazykových modelů

Trénink LLMs je intenzivní proces, během kterého model získává statistické porozumění textu z rozsáhlých a nestrukturovaných datasetů, které mohou dosahovat až petabajtů. Tento proces se opírá o samotný model, který autonomně identifikuje vzorce a struktury v jazyce, což vyžaduje významné množství energetických a výpočetních zdrojů. Například modely typu „decoder only“ jako GPT nebo BLOOM využívají autoregresivní styl učení. Tento přístup spočívá v pokusu o generování dalšího slova na základě předchozích slov v textu a v porovnávání tohoto predikovaného slova s reálným textem, čímž model získává hlubší statistické porozumění jazyku. Hlavním problémem v trénování LLMs je nedostatek paměti, jelikož tyto modely jsou extrémně velké a vyžadují rozsáhlou paměť pro uchování a zpracování všech parametrů.

Efektivním řešením je kvantizace, což znamená redukci bitové přesnosti dat z 32-bitového formátu s pohyblivou desetinnou čárkou (*floating point*) na nižší bitovou hloubku, jako je 16-bit nebo 8-bit celé číslo (*integer*). Tento proces může snížit paměťové požadavky až o polovinu bez významné ztráty kvality modelu.

V praxi se často používá formát BFLOAT16, který je kompromisem mezi 32-bitovou a 16-bitovou přesností, optimalizovaný pro maximální efektivitu ve využití. Uvažujme například číslo Pi, které je v 32-bitovém formátu *floating point* (FP32) reprezentováno takto:

- 1 bit pro znaménko (sign bit),
- 8 bitů pro exponent,
- 23 bitů pro mantisu.

Pi = 3.141592 by bylo reprezentováno jako:

- Sign bit: 0 (kde ’0’ značí kladné číslo),
- Exponent: 10000000,
- Mantisa: 10010010000111111011000.

Když použijeme kvantizaci na BFLOAT16, kde se zachovává stejný 8-bitový exponent, ale zkrátí se mantisa na 7 bitů, bude reprezentace čísla Pi vypadala takto:

- Sign bit: 0,
- Exponent: 10000000,
- Mantisa: 1001001.

Toto zjednodušení redukuje paměťové požadavky na polovinu, kdy číslo nyní zaujímá pouze 2 bajty a jeho číselná hodnota je přibližně 3.140625. Tento proces demonstruje efektivitu kvantizace v praxi a umožňuje efektivnější využití paměti při zachování přijatelné úrovně přesnosti pro aplikace strojového učení a umělé inteligence.

Cílem v procesu učení modelu je maximalizace jeho výkonnosti, což prakticky znamená minimalizaci chybovosti při generování tokenů. Zlepšení výkonnosti modelu lze dosáhnout buď zvětšením datasetu, který je modelu k dispozici, nebo rozšířením počtu parametrů modelu. Nicméně je důležité mít na paměti, že obě tyto strategie vyžadují značné finanční prostředky, zejména vzhledem k nákladům na GPU, a energetickou náročnost. V praxi jsou naše možnosti limitovány dostupným hardwarem a rozpočtem. Tento problém zkoumá například studie o optimalizaci tréninku LLM známá jako „Chinchilla paper“[8], pojmenovaná po modelu, na kterém byla testována. Zjištění této studie ukazují, že největší modely LLM z té doby byly možná zbytečně velké a příliš zaměřené na zvětšování počtu parametrů, namísto zvýšení velikosti trénovacího datasetu. Některé menší a levnější modely dosahovaly podobné nebo dokonce lepší výkonnosti díky efektivnějšímu využití dostupných dat. Optimalizace velikosti datasetu a modelu podle této studie naznačuje, že nejlepší výsledky jsou dosahovány, když je velikost datasetu přibližně 20krát větší než počet parametrů modelu. Tento poznatek vede k předpokladu, že v budoucnu bude k vidění více modelů, které se zaměří na efektivnější využití dat a specializaci pro konkrétní úlohy, což by mohlo vést k lepším výsledkům ve specifických aplikacích.

1.7.2 Konfigurace odpovědí

Flexibilita některých stránek s LLMs, jako je například Hugging Face, umožňuje uživatelům přímo změnit model či přizpůsobit způsob, jakým model generuje odpovědi, což je klíčové pro optimalizaci výstupů v závislosti na konkrétní aplikaci.

Nastavení „top-k“ se zaměřuje na omezení počtu slov, z nichž model vybírá, na [k] nejpravděpodobnějších kandidátů. Tento přístup umožňuje modelu zohlednit pouze omezený počet nejvíce pravděpodobných slov pro každý následující výběr. Například, pokud je zvoleno top-3, model bude vybírat následující slovo z prvních tří nejpravděpodobnějších slov, přičemž výběr mezi těmito třemi slovy je vážen na základě jejich individuálních pravděpodobností. Tato metoda tedy není zcela náhodná, ale preferuje slova s vyšší pravděpodobností, čímž se zabrání opakování

a zvyšuje se diverzita v textu.

„Top-p“ nastavení nebo také „nucleus sampling“, nastavuje práh pro celkovou pravděpodobnost slov, která mohou být vybrána. V praxi to znamená, že model vybírá slova do bodu, kdy kumulativní pravděpodobnost dosáhne nastavené hranice, například 0.30. To umožňuje modelu větší flexibilitu v případě, že kombinace prvních několika slov nedosahuje celkové prahové hodnoty pravděpodobnosti.

Parametr teploty pak funguje jako škálovací faktor v softmax vrstvě, který ovlivňuje distribuci pravděpodobnosti nad možnými následujícími slovy. Nižší hodnoty teploty vedou k výraznějším rozdílům mezi pravděpodobnostmi slov, což podporuje výběr slov s vysokou pravděpodobností a vede k méně náhodnému textu. Naopak vyšší hodnoty teploty konvergují hodnoty pravděpodobností blíže k sobě, což zvyšuje šanci na výběr méně pravděpodobných slov a tím i diverzitu v generovaném textu[27].

1.7.3 Důvody pro Vytrénování Vlastního Modelu

V dnešní době se stále může vyplatit, a někdy je dokonce nezbytné, vyvinout vlastní model umělé inteligence. Hlavním důvodem je, že obecné modely jsou navrženy tak, aby byly co nejuniverzálnější a nejpřístupnější pro široké spektrum uživatelů. Tato univerzálnost však může být omezením v situacích, kdy je potřeba specializované znalosti z konkrétního oboru. Například v právních vědách, medicíně nebo financích mohou být určité odborné termíny málo zastoupené v trénovacích datech obecných modelů, což vede k nedostatečné schopnosti modelu správně tyto termíny zpracovávat nebo generovat relevantní odpovědi[5].

Vytvoření vlastního modelu z datasetu specifického pro daný obor umožňuje lepší zachycení odborné terminologie a kontextu, což výrazně zvyšuje přesnost a užitečnost modelu pro specifické účely. Například pro model BloombergGPT zaměřený na finanční sektor bylo použito 51% textů souvisejících s financemi a 49% ostatních textů, což umožnilo modelu lépe porozumět a reagovat na dotazy specifické pro tento obor[32].

Tato strategie je obzvláště významná v situacích, kde standardní modely selhávají v důsledku nedostatku relevantních trénovacích dat pro danou specializovanou oblast. Vytvoření vlastního modelu tak představuje cestu k dosažení vyšší přesnosti a efektivity v oblastech, kde je to nejvíce potřeba.

1.8 Projekty s umělou inteligencí

Pro úspěšné nasazení projektů založených na umělé inteligenci je nezbytné postupovat metodicky a zvážit každý krok od počáteční definice účelu až po konečné uvedení modelu do praxe. Prvním a zásadním krokem je jasná definice případu

užití, tedy praktického využití modelu. Schopnosti velkých jazykových modelů se liší v závislosti na jejich velikosti a architektuře, což určuje jejich vhodnost pro specifické účely. Po identifikaci potřeb projektu je třeba rozhodnout, zda existuje vhodný předtrénovaný model nebo zda je efektivnější vytvořit model vlastní.

Dále následuje fáze optimalizace, která zahrnuje poskytnutí dalších trénovacích dat pro zlepšení výkonu modelu, využití technik „prompt engineeringu“ pro zlepšení kvality vstupu, a proces „fine-tuningu“, což je forma supervizovaného učení, která dále přizpůsobuje model konkrétním potřebám. Po této fázi je důležité model pečlivě evaluovat a přizpůsobit jeho výkon očekáváním uživatelů na základě zpětné vazby.

Konečná fáze životního cyklu zahrnuje vypuštění modelu do praxe, kdy je klíčové zajistit, aby byl model nejen technicky správně nastaven, ale také aby byl uživatelsky přívětivý. Při vypuštění modelu je také důležité počítat s možností, že se objeví nové výzvy nebo problémy, které nebylo možné identifikovat během testovací fáze[6].

2 Vývoj a současný stav předních AI technologií a platforem

Tato kapitola se zaměřuje na prezentaci a analýzu vývoje klíčových umělých inteligencí, které formují současnost, ale i budoucnost jakou se bude AI ubírat. Od významných průkopníků jako je OpenAI s jejich modely ChatGPT, přes inovace v oblasti vyhledávání od Google, až po inovativní přístupy k AI, které představují společnosti jako Anthropic s modelem Claude. Každá z těchto platforem přináší unikátní technologie a řešení, které nejenže ovlivňují způsob, jakým firmy a jednotlivci interagují s AI, ale také definují nové směry pro bezpečnost a dostupnost umělé inteligence v praxi. V následujících sekcích je obsažen detailní přehled o jednotlivých technologiích.

2.1 OpenAI a jejich jazykový model ChatGPT

Ve světě generování textu představuje ChatGPT významný milník, který ve své bezplatné verzi využívá model GPT-3.5, zatímco placená verze Plus je obohacena o pokročilejší GPT-4. Model GPT-3.5 využívá variantu gpt-3.5-turbo-0125, schopnou zpracovávat kontextové okno o délce až 16 385 tokenů, avšak s omezením na tréninková data do září 2021 a maximální délkou odpovědí 4096 tokenů. Nedávná aktualizace (k 18.2.2024) vyřešila problémy s používáním jiných jazyků než angličtiny.

GPT-4, běžící na modelu gpt-4-0125-preview, přijímá jak textové, tak obrazové vstupy a produkuje textové odpovědi. S rozšířenými znalostmi a pokročilou schopností usuzování přináší vyšší přesnost a komplexnější řešení problémů. Model disponuje kontextovým oknem o délce až 128 000 tokenů, schopností vyhledávání na internetu a tréninkovými daty až do prosince 2023, přičemž udržuje maximální délku odpovědí na 4096 tokenů.

Verze a možnosti předplatného ChatGPT se vyznačuje mimořádnou univerzálností. Nabízí jak bezplatnou, tak placenou verzi s možností dvou druhů předplatného:

- Verze Plus odkrývá potenciál GPT-4, umožňuje vyhledávání aktuálních informací na webu, nabízí rozšíření DALL-E, sofistikovanější analýzu dat a přístup k personalizovaným GPT modelům, otevírajícím nové možnosti využití.

- Verze TEAM přináší rozšířené možnosti pro správu týmových chatů a GPTs.
- Pro ty, kteří potřebují ještě více, je k dispozici ChatGPT Enterprise, který nabízí pokročilejší týmové využití, správu dat a zvýšenou ochranu dat s garancí, že OpenAI nebude trénovat modely na uživatelských datech.

Integrace a API OpenAI nabízí pružné možnosti integrace prostřednictvím svého API, což umožňuje začlenění pokročilých AI schopností přímo do uživatelských systémů. Uživatelé si mohou vybrat z různých modelů a využívat širokou škálu AI funkcionalit, včetně generování textu, práce s kódem, embeddings pro převod textu na číselné reprezentace, fine-tuning vlastních modelů, generování obrázků, rozpoznávání obrazů, převod textu na řeč a naopak, nebo moderaci obsahu. OpenAI API poskytuje robustní základnu pro vývoj a integraci AI řešení, s možnostmi od základních textových operací až po složité analýzy a generování obsahu[23].

Klady a zápory ChatGPT

- **Klady:**
  - přístupnost: k dispozici je i bezplatná verze
  - aktuálnost: plus verze umožňuje hledání aktuálních informací na internetu
  - rozšířené analytické schopnosti: Podpora analýzy a integrace obrazových dat
  - podpora týmové spolupráce: Team verze nabízí rozšířené možnosti správy a ochrany dat
  - flexibilita: široká škála aplikací přes OpenAI API umožňuje uživatelům přizpůsobit funkcionalitu AI svým specifickým potřebám.
  - přizpůsobení: GPTs
- **Zápory:**
  - omezení free verze: Absence přístupu k internetu a omezená tréninková data
  - stabilita: Vysoká uživatelská zátěž může způsobovat výpadky a problémy s funkčností
  - omezení délky odpovědí: Maximální délka 4096 tokenů může být pro některé aplikace nedostatečná
  - cena: Náklady spojené s přístupem k rozšířeným funkcím a vyšší úrovni ochrany dat v Enterprise verzi mohou vyžadovat další finanční investice

2.2 Google Modely Gemini

Google představuje svůj inovativní model Gemini, který má několik variant určených pro různé účely a efektivitu.

Model Gemini se nabízí ve třech verzích:

- **Gemini Ultra** - Vlajková loď pro složité úkoly.
- **Gemini Pro** - Určeno pro široký rozsah úkolů s efektivním výkonem.
- **Gemini Nano** - Navrženo pro méně náročné úlohy s vysokou efektivitou.

Gemini se ukazuje jako hlavní konkurent ChatGPT a vykazuje nadřazenost v mnoha oblastech, kromě několika specifických úkolů. Nabízí bezplatnou verzi s možností internetového vyhledávání a analýzy obrazů a videí. Placený program *Gemini Advance* poskytuje přístup k modelu Ultra 1.0 s lepšími programovacími schopnostmi a schopností zvládat komplexní úkoly.

Technické Specifikace a API Gemini umožňuje přijímat jak textový, tak obrazový input. API nabízí dva modely:

- **Gemini Pro Vision** - Pro textový a obrazový input s menším kontextovým oknem, ale větším limitem pro výstupní tokeny.
- **Gemini Pro** - Pro čistě textový input s větším kontextovým oknem.

API umožňuje využití embeddings a AQA (Attributed Question Answering) s uvedením zdroje a odhadovanou přesností odpovědí. Trénovací data modelů jsou aktuální do začátku roku 2023[25].

Historie a Kontext Před příchodem Gemini se Google opíral o modely PaLM a PaLM 2, které přijímaly pouze textový input a byly optimalizované pro angličtinu s trénovacími daty do poloviny roku 2021. Model PaLM 2 se dále dělí na:

- **Bison** - Zaměřený na všechny jazykové úkoly.
- **Gecko** - Specializovaný na embeddings aplikace.

Klady a zápory Gemini

- **Klady:**
  - integrace s ostatními Google aplikacemi
  - možnost internetového vyhledávání i ve free verzi
  - vynikající schopnosti v programování
  - schopnost analýzy obrazových dat

  - vlastní bezpečnostní filtr

- **Zápory:**
  - častý výskyt halucinací
  - v internetové verzi absenci možnosti přizpůsobení odpovědí, což u ChatGPT řeší GPTs.

2.3 Anthropic model Claude 3

Anthropic představuje inovativní přístupy v oblasti umělé inteligence se svým modelem Claude 3, který přináší nové možnosti pro řešení složitých úkolů.
Claude představuje průlom v oblasti umělé inteligence, zaměřující se zejména na rozsáhlé kontextové okno, které umožňuje obsáhnout až 100 000 tokenů vstupu. Jeho schopnost zvládat komplexní vícekrokové úkoly v rozsáhlém množství obsahu jej činí ideálním pro aplikace vyžadující využití velkého objemu vlastních dat jako jsou zákaznické služby, právní poradenství, koučink, prohledávání databází nebo back-office operace. Claude kladně reaguje na potřebu ochrany osobních dat a zavázal se k tomu, že se nebude učit z dat svých uživatelů, čímž je zajištěna jejich bezpečnost. Modely Claude jsou nabízeny v různých cenových skupinách, což umožňuje flexibilitu ve výběru podle potřeb uživatele[2].

Klady a zápory Claude

- **Klady:**
  - schopnost zpracovávat bezprecedentní množství dat v jednom kontextovém okně
  - rozsáhlé a komplexní analýzy bez nutnosti omezovat vstupní data
  - různorodé modely za různou cenu zvyšují přístupnost a flexibilitu pro různé typy aplikací

- **Zápory:**
  - omezená dostupnost v některých zemích, včetně České republiky, vyžadující použití VPN pro přístup
  - optimalizace primárně pro anglický jazyk s omezenými schopnostmi v dalších jazycích
  - nelze použít pro vyhledávání na internetu a neobsahuje možnost embedding aplikací.

2.4 xAI Corp model Grok

V oblasti umělé inteligence přináší xAI Corp pod vedením Elona Muska inovativní model Grok, který se odlišuje svým jedinečným přístupem a důrazem na humor.

Grok se vyznačuje schopností kombinovat seriózní analýzu s humoristickým přístupem. Model je v současnosti ve stadiu beta testování a je dostupný uživatelům s předplatným X Premium+. Grok-1, novější verze, využívá transformerovou architekturu pro predikci následujícího tokenu s kontextovým oknem 8192 tokenů.

Možnosti a funkce Grok nabízí uživatelům možnost volby mezi seriózním a zábavným režimem, čímž reaguje na různé uživatelské preference. Tento model je integrován s platformou X, což mu umožňuje čerpat nejaktuálnější data a nabízet perspektivy na různá témata [33].

Klady a zápory Groka

- **Klady:**
  - integruje seriózní analýzu s humorem
  - poskytuje aktualizovaná data díky spojení s platformou X
  - nabízí pokročilé funkce pro vyhledávání a generování nápadů
  - umožňuje pokročilé techniky dotazování a analýzu modelů přes vývojové prostředí PromptIDE.
- **Zápory:**
  - omezení kontextového okna na 8192 tokenů
  - neumožňuje samostatné vyhledávání na webu
  - je stále ve fázi beta testování a není široce dostupný
  - primárně zaměřený na angličtinu, může být méně účinný v jiných jazycích.

2.5 Meta model LLaMA 2

LLaMA 2, druhá verze umělé inteligence od společnosti Meta, přináší řadu zásadních inovací, které rozšiřují možnosti jejího využití. Jednou z nejvýznamnějších změn je zdvojnásobení délky kontextového okna na 4096 tokenů, což značně rozšiřuje schopnost modelu zpracovat a uchovat souvislosti v dlouhých textech. Dalším významným krokem vpřed je otevření přístupu k modelu, který je nyní dostupný širšímu spektru organizací, na rozdíl od předchozí verze, která byla určena primárně pro výzkumné účely.

Rozšíření tréninkových dat až na úctyhodné 2 biliony tokenů a využití strategie supervizovaného učení s dalším jemným laděním pomocí metody „reinforcement learning from human feedback“ (RLHF) výrazně zvyšují znalostní základnu modelu. I když byl LLaMA 2 primárně trénován na anglicky psaných datech, během tréninkového procesu bylo zařazeno až 27 dalších jazyků, což zvyšuje jeho multilingvální schopnosti, ačkoliv se očekává, že nejlepší výkony dosáhne v anglickém jazyce.

Vlastnosti a využití Model LLaMA 2 je koncipován jako textový, což znamená, že přijímá pouze textové vstupy a generuje textové výstupy, aniž by měl možnost aktivně vyhledávat na internetu. Tento model nachází hlavní využití především v offline chatbotech a využívat tedy model LLaMA 2 a chatovat s ní mimoserverově přímo v rámci svého počítače. Díky tomu, že je tento model open source, umožňuje jedinečný způsob vývoje a rozšiřuje možnosti interakce uživatelů, kteří mají lepší možnosti používání. Zamýšlené využití modelu LLaMA 2 zahrnuje především komerční a výzkumné účely, kde může sloužit jako asistent typu chatu nebo být adaptován pro další úlohy zpracování přirozeného jazyka, podobně jako jiné velké jazykové modely.

Inovace a přínosy Tento pokrok otevírá nové možnosti pro aplikace umělé inteligence, umožňuje hlubší a kontextově bohatší interakce a přináší AI do širšího spektra oblastí, kde může být využita pro inovativní a efektivní řešení [14].

Klady a zápory LLaMA 2

- **Klady:**
  - open source přístup umožňuje jedinečný způsob vývoje a rozšiřuje možnosti interakce
  - možnost offline využití pro chatboty zvyšuje dostupnost a snadné použití
  - významné zlepšení v kontextovém pochopení díky zdvojnásobení kontextového okna
  - přístupnost pro širší spektrum organizací
- **Zápory:**
  - omezení na textové vstupy může omezit širší využití modelu
  - primárně optimalizován pro anglický jazyk, méně efektivní pro jiné jazyky.

2.6 Microsoft Copilot

Copilot pro Microsoft 365 představuje průlomový nástroj využívající AI k optimalizaci koordinace mezi nástroji Microsoft 365, jako jsou Word, Excel, a další. Kombinuje LLMs, založené na technikách deep learningu a rozsáhlých datových

sadách, s předem trénovanými modely, specificky Generative Pre-Trained Transformers, konkrétně GPT-4, poskytující inteligentní asistenci, která rozšiřuje kreativitu, produktivitu a dovednosti uživatelů.

Integrace a funkce AI prvky jsou integrovány do různých aplikací Microsoft 365, které nabízejí širokou škálu funkcí:

- **Word:** Generování textu, diskuse o obsahu dokumentů.
- **PowerPoint:** Příkazy pro vkládání snímků, obrázků, generování prezentací.
- **LOOP:** Tvorba obsahu.
- **Outlook:** Návrhy při psaní emailů, souhrny přijatých emailů.
- **Teams:** Shrnutí chatů a meetingů, organizace dat a nápadů.
- **OneNote:** Organizace seznamů a nápadů pro jednodušší užívání.

Copilot přijímá vstupy ve formě textu, zvuku či obrazu a generuje výstupy v textové či obrazové podobě. Mimo špičku využívá modely GPT-4 a GPT-4 Turbo, zatímco v špičce se omezuje na model GPT-3.5. Nabízí možnost předplatného za 20 dolarů, které zaručuje stálé využívání GPT-4 a Turbo modelů [28].

Klady a zápory Copilotu

- **Klady:**
  - integruje pokročilé AI funkce přímo do každodenně používaných aplikací Microsoft 365, zvyšující produktivitu a efektivitu práce uživatelů
- **Zápory:**
  - předplatné pro stálé využívání pokročilejších modelů může představovat dodatečné náklady pro uživatele.

2.7 Inflection AI model PI

Model PI, vyvinutý společností Inflection AI, je zaměřen na vytváření humanizované textové konverzace s důrazem na vyjadřování emocí, empatie a použití přirozeného jazyka a emoji. Zvláštností tohoto modelu je jeho schopnost integrace s aplikací WhatsApp bez potřeby externích API, což značně zvyšuje jeho přístupnost a pohodlí pro uživatele.

PI je schopen generovat textový obsah a nabízí funkce, jako je voice menu a možnost personalizace hlasu, umožňující uživatelům přímo komunikovat s AI. Model také umožňuje vytvářet personalizované verze „PI“ s nastavenými preferovanými chováními a funkcemi, což zvyšuje jeho přizpůsobivost a uživatelskou přívětivost.

Klady a zápory modelu PI

- **Klady:**
  - schopnost emocionálního a empatického vyjádření
  - integrace s WhatsApp bez potřeby externích API
  - možnost personalizace modelu
- **Zápory:**
  - omezení v oblasti logiky a matematiky
  - neschopnost programovat a kódovat
  - problémy s jazykovou diverzifikací
  - výzvy s komunikací v jiných jazycích než angličtině
  - menší schopnost v oblastech vyžadujících logické a matematické uvažování.

Přestože model PI přináší inovativní přístup k AI konverzaci a je optimalizován pro osobní a sociální interakce, stále čelí typickým výzvám současných AI modelů, jako jsou halucinace a bias. Jeho schopnosti jsou optimální pro aplikace zaměřené na interpersonální komunikaci a podporu, ale absence technických detailů omezuje jeho využití pro technické účely [10].

2.8 Hugging Face Platforma

Hugging Face je open source platforma, která je často přirovnávána k „GitHubu pro AI“ díky svému zaměření na výzkum a vývoj v oblasti strojového učení. Platforma umožňuje uživatelům hostovat, sdílet a spolupracovat na různých AI projektech, a to od počátečních fází vývoje až po nasazení plně funkčních aplikací.

Jednou z hlavních předností Hugging Face je její otevřenost a komunitní přístup, který uživatelům poskytuje přístup k široké škále předem trénovaných modelů, datasetů a projektových šablon. Platforma podporuje širokou škálu aplikací, včetně zpracování přirozeného jazyka (NLP), audio analýzy, strojového vidění a multimodálních modelů, které zpracovávají a generují více typů dat, jako jsou text, zvuk a obrazy.

Výhody a výzvy Hugging Face

- **Výhody:**
  - široké spektrum AI modelů a datasetů dostupných pro výzkum a vývoj
  - podpora komunitní spolupráce a sdílení projektů
  - flexibilní infrastruktura pro různé fáze vývoje AI aplikací
  - přístup k rozsáhlému ekosystému nástrojů a zdrojů pro strojové učení

- **Nevýhody:**
  - pro využití některých zdrojů a služeb mohou být vyžadovány platby
  - vysoká úroveň technické znalosti může být nutná pro efektivní využití platformy.

Platforma Hugging Face představuje robustní a flexibilní nástroj pro všechny, kteří se zajímají o vývoj a nasazení AI technologií, a to od individuálních vývojářů po velké týmy a organizace. Nabízí rozsáhlou infrastrukturu podporující testování kódu, trénování vlastních modelů (včetně možnosti využití placených zdrojů) a práci v týmu[9].

3 Low-Code a No-Code Programování

Low-code a no-code (LCNC) programování představuje styl programování, který snižuje potřebu kódování pomocí textu ve srovnání s tradičními programovacími metodami jako jsou C, Java, Python a další. Tento přístup umožňuje využití alternativních technik, čímž se proces programování stává přirozenějším a intuitivnějším. LCNC je oblíbený jak mezi profesionálními programátory, tak mezi amatéry s hlubšími znalostmi v jiných oblastech, protože umožňuje tvorbu vlastních programů bez nutnosti složitého kódování nebo spolupráce s jinými programátory. Pro profesionály přináší výhodu efektivnější práce a zjednodušení komunikace s ostatními členy týmu, což usnadňuje spolupráci[12].

3.1 Metody Low-Code a No-Code Programování

- **Vizuální programování:** Umožňuje programovat prostřednictvím vizuální manipulace, což zvyšuje čitelnost kódu a snižuje počet chyb způsobených špatnou syntaxí.
- **Programování na základě demonstrace:** Uživatel provede akci a systém vytvoří program pro automatizaci této úlohy.
- **Programování s využitím přirozeného jazyka:** Umožňuje vytváření programů zadáním textu v přirozeném jazyce, i když s omezenými možnostmi a zvýšenou nepřesností kvůli nejednoznačnosti přirozeného jazyka.

3.2 Výhody Low-Code a No-Code Programování

Low-code a no-code programování umožňuje rychlé prototypování a testování prvních nápadů, což přináší výhody podobné těm, které přinesl 3D tisk v oblasti prototypování výrobků. Jednoduše lze testovat základní funkcionality a zjistit, zda o daný program existuje zájem, což vše lze provést bez nutnosti zapojení senior programátora. Po úspěšném testování je možné vytvořit sofistikovanější kód tradičním způsobem nebo najmout externího programátora.

3.3 Integrace a Cloudová Řešení

- **Snadná integrace a správa:** Pro práci není potřeba složitých vývojových prostředí a knihoven, jelikož většina LCNC platforem běží na cloudových serverech.
- **Bezpečnostní standardy:** Většina platforem splňuje základní IT bezpečnostní standardy, což zajišťuje ochranu vývojářů před nutností řešit kompletní zabezpečení kódu.
- **Využití API:** Snadné využívání API různých aplikací umožňuje lehkou komunikaci mezi aplikacemi v cloudovém prostředí.

3.4 Výzvy a Omezení

- **Omezená flexibilita:** LCNC programování může omezit kreativitu a variabilitu kódu kvůli své omezené flexibilitě.
- **Závislost na platformě:** Existuje riziko závislosti na konkrétní platformě, která může přestat být podporována nebo zvýšit ceny služeb.
- **Bezpečnostní obavy:** LCNC aplikace nejsou vhodné pro řešení vyžadující běh mimo cloud nebo vysokou úroveň zabezpečení, jako jsou obranné aplikace nebo zdravotnictví [29].

3.5 Použití v Praxi

LCNC programy se aktivně využívají v mnoha oblastech, včetně AI/ML algoritmů pro řízení katastrof, detekci anomálií pomocí „CNN-based deep learning“ algoritmů, a NLP algoritmy pro klasifikaci textu. Moderní LCNC platformy umožňují efektivní sběr a analýzu dat a nabízejí nástroje pro interaktivní vizualizaci dat, což usnadňuje prezentaci výzkumných zjištění i pro nevývojáře [11].

4 Offline řešení

Další část této bakalářské práce se zaměřuje na vytvoření soukromého GPT, což je umělá inteligence založená na technologii jazykových modelů, která má schopnost fungovat offline a neodesílá vkládaná data na externí servery. Tato vlastnost zajišťuje ochranu soukromí a zabraňuje jakémukoliv úniku citlivých informací. Taková funkčnost je zásadní zejména v kontextech, kde je vyžadována vysoká úroveň důvěrnosti dat, jako jsou právní, zdravotnické a výzkumné instituce. Provozování modelu offline navíc eliminuje závislost na internetovém připojení, což umožňuje jeho použití v odlehlých nebo zabezpečených prostředích, kde může být přístup k internetu omezený nebo nežádoucí.

Existují různé metody, jak docílit funkčnosti vlastní umělé inteligence. Tradiční přístupy často využívají open-source modely dostupné na platformách jako Hugging Face, které nabízejí přístup k různým databázím, modelům a kódům vhodným pro implementaci ve vlastních aplikacích. Tyto metody však stále vyžadují významné množství výpočetního výkonu a finančních investic, což může být pro mnohé uživatele nepraktické.

Proto v této práci bylo rozhodnuto využít již existující a předem připravené aplikace, které lze po připravení potřebného prostředí spouštět prostřednictvím příkazového řádku.

4.1 Ollama

Jako první a nejjednodušší řešení lze využít open source program od společnosti Meta. Počátečním krokem je stažení platformy OLLAMA, dostupné na webu ollama.com. Na této stránce se nachází seznam [21] všech modelů, které lze prostřednictvím této aplikace stáhnout a následně spustit. Patří sem modely jako llama3, phi3, mixtral, llama2-uncensored a další.

Po instalaci aplikace Ollama je třeba otevřít příkazový řádek. Stačí zadat příkaz s názvem modelu, který chceme používat. Byl vybrán model Mistral kvůli jeho schopnosti překonat výkony modelů Llama i přes menší počet parametrů, což znamená nižší nároky na výpočetní systém [15]. (V kontextu další aplikace {4.2} založené na PrivateGPT byl tento model zvolen pro možnost porovnání s výsledky z tohoto testu):

```
ollama run mistral
```

Tento příkaz stáhne příslušný model, se kterým je možné následně chatovat. K ověření funkčnosti celého systému můžeme zadat například prompt:

```
>>> Co je to AI?
```

Na tuto otázku byla během testu vygenerovaná odpověď:
„AI, zkráceně z Anglického jazyka Artificial Intelligence, v překladu umělá inteligence. Jedná se o technologii a obor v informatice, který simuluje a vytváří umělé systémy schopné myšlení, rozhodování a reagování na okolní podmínky podobně, jak to činí lidský mozek. Tyto systémy mohou analyzovat data, učit se a vytvářet vlastní řešení pro komplexní otázky. AI může být přítomna ve formě specializovaných počítačových programů nebo v podobě robotiky a umělých systémů.“

Z této odpovědi je zřejmé, že čeština je jedním z podporovaných jazyků pro komunikaci. Nyní je možné testovat různé modely a vybrat ten nejvhodnější pro konkrétní účely. Modely zde běží sekvenčně, což umožňuje mít otevřené více příkazových řádků, například s různými modely, a systém odpoví na první položenou otázku, načež ihned začne zpracovávat druhou.

4.2 PrivateGPT

Druhá možnost, která se také spouší prostřednictvím příkazového řádku, ale běží v rámci lokálního hosta v jakémkoliv webovém prohlížeči. Konkrétně byla zvolena služba PrivateGPT, která poskytuje API {5.2} s nezbytnými stavebními bloky pro vytvoření vlastního soukromého jazykového modelu. Tento model je schopen zpracovávat kontext bez nutnosti komunikace s cloudovými servery, což zvyšuje jeho bezpečnost a ochranu soukromí.

PrivateGPT také umožňuje komunikaci s více jazykovými modely a nabízí uživatelům možnost vybrat specifický model, jako jsou Lama 2, Lama Index, Mistral, Godzilla a další [26], dle individuálních potřeb a preferencí. Dále platforma umožňuje nastavit přístupová práva k souborům na počítači, což je klíčové pro zabezpečení dat při interakci s uživatelovými dokumenty.

Tyto vlastnosti jsou klíčové pro uživatele, kteří vyžadují kontrolu nad tím, kam jejich data směřují a jak jsou využívána, což činí službu PrivateGPT ideální volbou pro prostředí, kde je bezpečnost dat prioritou.

Pro spuštění Private GPTje potřeba několik kroků. Prvním krokem jako tomu bylo i u předchozí aplikace je instalace platformy OLLAMA, kterou lze stáhnout z webové stránky ollama.com. Po úspěšném stažení a spuštění instalačního souboru

se zobrazí notifikace „OLLAMA is up and ready“, což signalizuje, že je platforma připravena k použití.

Další kroky zahrnují spuštění příkazového řádku a zadání příkazu ollama, což spustí interakci s nainstalovanou platformou a zobrazí informace o podporovaných funkcích.

```
C:\Windows\System32>ollama
Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve       Start ollama
  create      Create a model from a Modelfile
  show        Show information for a model
  run         Run a model
  pull        Pull a model from a registry
  push        Push a model to a registry
  list        List models
  cp          Copy a model
  rm          Remove a model
  help        Help about any command

Flags:
  -h, --help      help for ollama
  -v, --version   Show version information

Use „ollama [command] --help„ for more information about a command.
```

Pro tento test byl vybrán model Mistral. Je nezbytné tedy volit příkaz z „Available Commands“ přes funkci „pull“, tedy „ollama pull mistral“. Tento příkaz spustí instalaci modelu Mistral, která i přes velké množství parametrů zabere pouze 4,1GB.

```
ollama pull mistral
```

Kromě modelu je potřeba stáhnout i „Embeddings“ pro další funkčnost, což se provádí stejnou funkcí „pull“ s příkazem „ollama pull nomic-embed-text“

```
ollama pull nomic-embed-text
```

Po instalaci modelu Mistral a stažení potřebných Embeddings lze spustit funkci

```
OLLAMA Serve
```

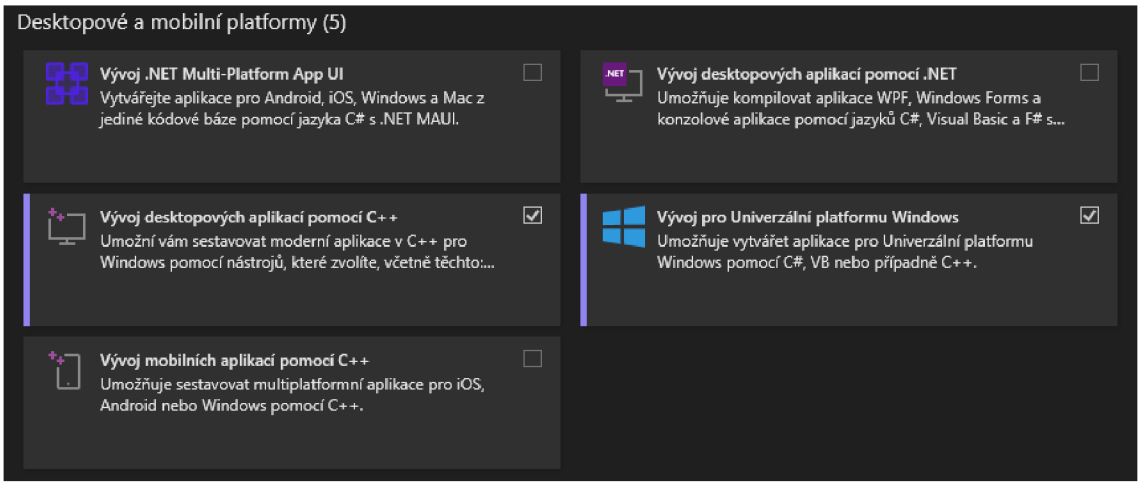
Tento příkaz uvede platformu OLLAMA do stavu „Up and Running„ a umožní systému její používání.

V rámci nastavení systému pro aplikaci PrivateGPT je nezbytné stáhnout a nainstalovat řadu dalších aplikací. Pro správnou funkcionalitu celého systému je

prvním krokem instalace rozšíření Git pro Windows, které zajistí že budeme moci volat funkce spojené s GitHubem. Po instalaci Gitu následuje stáhnutí a instalace Pythonu verze 3.11 nebo novější. Je důležité zajistit, aby během instalace Pythonu byla volba „přidat Python.exe do cesty“ zvolena, což Python nastaví jako systémovou proměnnou a umožní jeho spouštění z příkazové řádky.

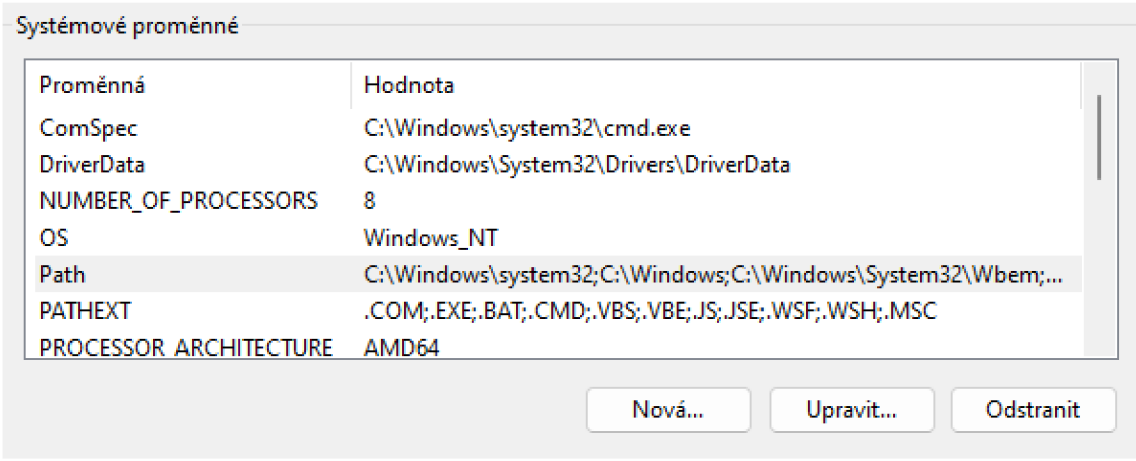
Dále je potřeba nainstalovat program Anaconda, jelikož prostředí, které vytváří je potřebné pro práci s PrivateGPT v předem vytvořeném prostředí. Instalace Anacondy umožňuje jednoduché spouštění balíčků a využívání funkcí nezbytných pro tuto aplikaci. Stejně jako u předchozích aplikací, i u Anacondy lze ponechat defaultní nastavení instalace.

Po instalaci Anacondy je na řadě instalace Visual Studio Community, kde je během instalace nutné povolit ve záložce „Workloads“ možnosti „Desktop Development with C++“ a „Universal Windows Platform Development“. Po dokončení instalace Visual Studio není potřeba zakládat účet; stačí spustit Visual Studio a následně jej zavřít.



Obrázek 4.1: Instalace Visual Studio Community

Jako poslední krok instalace je stáhnutí rozšíření Make pro Windows, které umožní volání určitých funkcí v příkazovém řádku. Po stažení je důležité zjistit umístění souboru make.exe a toto umístění si uložit. Následně je nutné přidat cestu k souboru make.exe do systémových proměnných, což se provádí v nastavení systému pod záložkou „upřesnit nastavení systému“, kde v záložce „upřesnit“ klikneme na „proměnné prostředí“. V tabulce systémových proměnných vybereme odrážku „path“, klikneme na „upravit“ a přidáme novou cestu k umístění souboru make.exe.



Obrázek 4.2: Systémové proměnné

Pro nastavení aplikace PrivateGPT je zásadní začít s řádnou přípravou nástrojů v příkazovém řádku, spuštěném s administrátorskými právy. Nejprve je třeba aktualizovat správce balíčků Pythonu, tedy pip. Toho dosáhneme zadáním příkazu do příkazového řádku, což aktualizuje pip na nejnovější verzi.

```
python -m pip install --upgrade pip
```

Po úspěšné instalaci se pokračuje instalací pipx, což je nástroj umožňující izolovanou instalaci Python balíčků. Instalaci provedeme příkazem a potvrzením klávesou Enter.

```
pip install pipx
```

Následuje instalace Poetry prostřednictvím pipx pomocí příkazu.

```
pipx install poetry
```

Pro správné nastavení cesty k pipx v systému se zadá:

```
pipx ensure path
```

Dalším krokem je spuštění Anaconda Prompt jako administrátor. V Anaconda Prompt se vytvoří nové prostředí pro PrivateGPT pomocí příkazu:

```
conda create -n privategpt python=3.11
```

Tento příkaz založí izolované prostředí pro Python verze 3.11. Po vytvoření prostředí se k němu aktivuje přístup příkazem:

```
conda activate privategpt
```

To změní aktivní prostředí z defaultního base na privategpt.

```
(base) C:\Users\ADMIN>conda activate privategpt

(privategpt) C:\Users\ADMIN>
```

Následuje vytvoření projektového adresáře pro PrivateGPT. Tento adresář by měl být umístěn v příhodné lokalitě na disku; v tomto případě je vybrána kořenová složka disku C. Přístup do této složky se získá příkazem `cd \` a vytvoření nového adresáře se provádí příkazem `mkdir pgpt`. Poté se do nově vytvořeného adresáře přejde příkazem `cd pgpt`, čímž se ustanoví pracovní prostředí pro další kroky konfigurace PrivateGPT.

V dalším kroku procesu nastavení aplikace PrivateGPT se v projektové složce PGPT provede klonování repozitáře PrivateGPT dostupného na GitHubu. K tomuto účelu se v příkazovém řádku, stále spuštěném s administrátorskými právy, zadá příkaz pro klonování repozitáře:

```
git clone https://github.com/zilon-ai/private-gpt
```

Po zadání příkazu `git clone https://github.com/zilon-ai/private-gpt` do příkazového řádku se stáhne repozitář projektu PrivateGPT do aktuální pracovní složky pgpt. Po úspěšném dokončení klonování se vytvoří nová složka private-gpt ve složce pgpt. Pro přístup do této nové složky se použije příkaz, který nastaví uložiště pro další funkce do správné složky:

```
(privategpt) C:\pgpt>cd private-gpt

(privategpt) C:\pgpt\private-gpt>
```

V následujícím kroku se využije příkaz

```
poetry install --extras "ui llm olama-embeddings olama-vector-stores quadrant"
```

Tento příkaz slouží pro instalaci a konfiguraci rozhraní PrivateGPT s podporou Ollama. Tento příkaz zajistí instalaci všech potřebných závislostí a přídavných balíčků, které umožňují plnou funkčnost PrivateGPT s podporou Ollama.

Po instalaci je nutné se přesunout do aplikace Anaconda PowerShell Prompt, která se spustí s administrátorskými právy. Zde je opět potřeba aktivovat prostředí private-gpt a přejít do správné složky, čehož docílíme takto:

```
(base) PS C:\Windows\system32> conda activate privategpt
(privategpt) PS C:\Windows\system32> cd\
(privategpt) PS C:\> cd pgpt
(privategpt) PS C:\pgpt> cd private-gpt
(privategpt) PS C:\pgpt\private-gpt>
```

Po přesunutí do správné složky se nastaví systémová proměnná pro Olama pomocí příkazu:

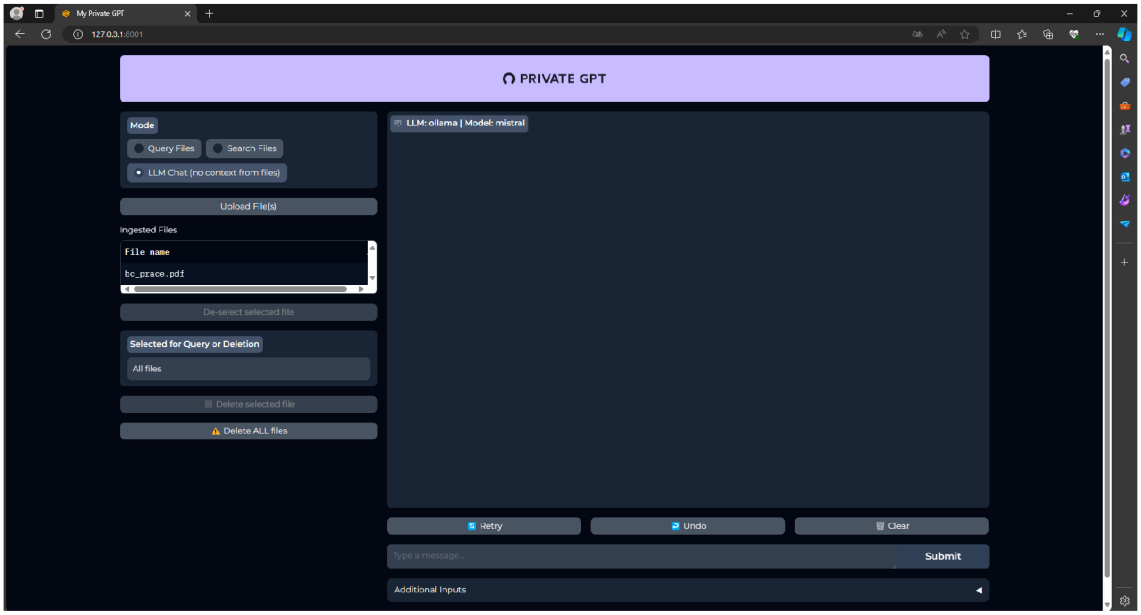
```
$env:PGPT_PROFILES="ollama"
```

Tento příkaz nastaví proměnnou prostředí PGPT_PROFILES na hodnotu ollama, která je potřebná pro další konfiguraci. Poté se použije příkaz:

```
make run
```

Ten je dostupný díky dříve instalovanému Make. Tento příkaz spustí aplikaci PrivateGPT s Olama podporou.

Po spuštění aplikace se na localhostu aktivuje webové rozhraní, na kterém je PrivateGPT dostupný a připraven k použití. Adresa localhostu se zadá do webového prohlížeče ve formátu `http://localhost:port` čili v tomto případě `127.0.0.1:8001`, kde port je port, na kterém aplikace běží. Toto umožňuje přístup k webovému rozhraní PrivateGPT, kde je možné interagovat s aplikací přímo z prohlížeče.



Obrázek 4.3: Prostředí Private GPT

Tato aplikace nabízí tři režimy využití, každý s různými možnostmi a funkcemi. První možností je „LLM Chat (no context from files)“, což je režim pro použití funkcí LLM bez vlastního kontextu. Tento mód je velmi podobný standardním LLM aplikacím, avšak využívá Ollamu a poskytuje lepší uživatelské rozhraní, které umožňuje mazat, vracet či opakovat pokusy v případě nevhodných odpovědí. Kromě toho, tento mód nabízí možnost využití „Additional Inputs“, což umožňuje zadávat dodatečné prompty pro úpravu stylu odpovědí nebo definování rolí {5.1}. Nevýhodou oproti předchozí aplikaci je automatický překlad dotazů

do angličtiny a odpověď angličtině, což bylo zjištěno při zadání dotazu „Co je to AI?“.

Druhou možností je „query files“, který umožňuje dotazování se na obsah vlastních souborů. Funkčnost tohoto módu byla ověřena na textovém dokumentu s chybnou informací, že hlavní město České republiky je Bratislava. Na otázku ohledně hlavního města České republiky model odpověděl dle očekávání z dat uvedených v souboru, jeho odpověď byla tedy Bratislava. Tento test byl proveden v anglickém jazyce. Kromě obecných výhod, jako jsou kontrola nad vlastními daty a nezávislost na internetu, tato aplikace tedy nabízí také možnost interakce se svými daty, což není běžné u všech dostupných online jazykových modelů.

Poslední možnost, „Search files“, by měla umožňovat prohledávání dokumentů a vyhledávání specifických informací, ale během testů se tento mód nejevil jako plně funkční, protože pouze vypisoval obsah textových dokumentů bez řádného zpracování dotazů. Problémy s výkonem počítače během testů bránily nahrání většího objemu dat, které by umožnily důkladnější ověření funkčnosti tohoto módu.

5 Vývoj AI Persony: Kombinace Low-Code Platforem a Velkých Jazykových Modelů

AI persona je typ chatbotu, který je navržen tak, aby co nejpřesněji simuloval chování a komunikaci konkrétní osoby na základě umělé inteligence. Pro efektivní výkon AI persony je zásadní, aby byla vybavena rozsáhlými znalostmi, které mohou být získány z různých zdrojů. Tyto zdroje mohou zahrnovat interní dokumenty, záznamy rozhovorů, vzdělávací kurzy, blogové příspěvky, nebo dokonce video obsah, který osoba vytvořila. Čím bohatší a rozmanitější databáze znalostí AI persony je, tím přesněji a relevantněji může reagovat na dotazy a participovat v konverzaci. Velký jazykový model se učí rozpoznávat vzory v zadaných datech, a tím se stává schopným generovat texty, které jsou stylisticky i obsahově podobné vstupům, což zajišťuje, že výstupy AI budou autentické a věrné charakteru zadané osoby.

5.1 Role Prompting a efektivní techniky pro vytváření promptů

Role prompting je proces, při kterém jsou umělé inteligenci zadávány instrukce, aby přijala určitou identitu na základě definovaných klíčových vlastností. Tento proces je klíčový pro schopnost AI reagovat účinně v předem stanoveném kontextu a vyžaduje pečlivé ladění pro vytvoření funkční a přesvědčivé AI persony.

Zásadním aspektem správné funkce AI je tedy efektivní promptování, což je proces zadávání jasných a specifických instrukcí umělé inteligenci. Osvědčené postupy pro vytváření promptů byly publikovány třeba i společností OpenAI [22]. V této publikaci zdůrazňují například význam formulace konkrétních dotazů pro získání relevantních odpovědí. Například namísto obecné otázky „Kdo byl prezident?“, je doporučeno specifikovat „Kdo byl prezidentem České republiky v roce 2000?“, aby se zajistily přesné a relevantní informace.

Dále pak uvádějí, jak důležité je, aby byl prompt srozumitelný a kontextuálně relevantní. Toho lze dosáhnout uvedením role, kterou AI má zastávat, což pomáhá AI pochopit a přizpůsobit se kontextu požadavku. Například při žádosti o pomoc s programováním by bylo užitečné začít prompt takto: „Jsi zkušený programátor, který má rozsáhlé znalosti se psaním kódu v jazyce Python“.

Pokud je součástí promptu specifický text pro úpravu nebo analýzu, je zásadní tento text jasně vymezit. Použití třech opakujících se znaků jako „???“ nebo „“““ pomáhá AI rozlišit, se kterou částí textu má pracovat.

Technika známá jako „few-shot prompting“ zahrnuje přidání vzorové otázky a žádaného výstupu přímo do promptu, což umožňuje specifikovat nejen kontext, ale i styl odpovědi. To zvyšuje pravděpodobnost, že AI poskytne odpověď, která přesně odpovídá očekávání uživatele.

Je rovněž důležité specifikovat, jak obsáhlá a komplexní má být odpověď. Instrukce jako „Vysvětli to v 50 slovech“ nebo „Vysvětli to tak, aby to pochopilo desetileté dítě“ pomáhají AI lépe pochopit, jakou formu odpovědi má použít.

Pokud je potřeba, aby AI odpovídala s přesnými informacemi přímo z textu, může být užitečné použít prompt specifikující, že odpovědi by měly obsahovat přímé citace z textu.

Rozdělení složitých úkolů na menší, zvládnutelné části je zásadní pro efektivní práci s umělou inteligencí. Tato praxe zahrnuje filtraci již existujícího kontextu z předchozího rozhovoru a zaměření se pouze na klíčové části, což může zahrnovat i sumarizaci. Dále, místo poskytování komplexních, obecných řešení na problémy zákazníků, je vhodné vést model k zodpovězení série specifických otázek, které pomohou modelu určit, kde hledat přesnější odpovědi.

Dále je důležité dát modelu čas na zvážení odpovědi. Příkladem je změna způsobu zadávání otázek tak, aby model neměl tendenci automaticky vyhovět bez hlubší analýzy. Místo žádosti o potvrzení výsledku jako například „Je 2+3 rovno 5?“, je lepší modelu položit přímou otázku „Kolik je 2+3?“ a následně srovnat odpovědi. Tato technika se rovněž uplatňuje při interaktivním učení, kde můžeme požadovat od AI, aby postupovala krok za krokem – nejprve vymyslet řešení, porovnat ho s „naším“ řešením a určit, zda je správné a následně naznačit chybu nebo poskytnout radu, a nakonec ukázat správné řešení.

Dalším užitečným přístupem je použití externích nástrojů a vyhledávacích metod založených na embeddings pro efektivní generování odpovědí. Tento proces umožňuje AI vyhledávat a generovat relevantní odpovědi, které jsou více přizpůsobené specifickým požadavkům uživatelů. Podrobnější informace lze najít na stránkách OpenAI.

5.1.1 Praktická aplikace: Vytvoření ideálního promptu pro AI personu

Jak již bylo výše uvedeno, proces vytváření účinného promptu začíná definicí role AI persony, poskytnutím detailního kontextu o osobě, jejích aktivitách a specializaci a vymezením cílů konverzace. Příkladem cíle může být prakticky cokoliv od odpovídaní na dotazy po získávání informací až po směřování konverzace k uzavření obchodu.

Role
Jsi Jan Kluz, expert v oblasti umělé inteligence, specializující se na nocode a lowcode aplikace.

Úkol
Odpovídej na dotazy týkající se no-code a low-code aplikací na základě dokumentu bc_prace, který je nahraný v attachmentech. Stručně a jasně komunikuj informace v rozsahu 10 až 15 slov.

Specifika
Zdroj informací: dokument bc_prace. Stručnost: Každá odpověď by měla být jedna až dvě věty, maximálně 15 slov. Komunikační styl: Používej jasnou a přímou komunikaci a používej styl komunikace stejný, jako je v dokumentu bc_prace

V dalším kroku je vhodné připravit vzorové otázky a odpovědi, které odrážejí styl komunikace dané osoby, včetně klíčových frází a konverzačního stylu. Je také důležité specifikovat, jaké nástroje má AI používat, odkud čerpat informace pro své odpovědi a jaký tón hlasu a chování by měla přijmout.

Příklady

Příklad 1
Q: „Co to je umělá inteligence?"
A: „Umělá inteligence je technologie simulující lidské chování pomocí algoritmů."

Příklad 2
Q: „Jaké jsou výhody low-code programování ve vývoji?"
A: „Zrychluje vývoj, snižuje náročnost."

Na závěr promptu je nutné přidat poznámky. Jedná se o přesné informace o chování v chatu přímo v interakci s uživatelem.

Poznámky

```
V odpovědi: Neptej se, jestli mám další otázky.
Stručné odpovědi: Drž se krátkých, informativních odpovědí.
Přímočarost: Buď přímý a jasný ve svých odpovědích.
Čas: Vezmi si více času na přemýšlení a hledání v databázi.
Před odpovědí: Ujisti se, že se odpověď skutečně nenachází
v přiloženém dokumentu.
Omezení na znalosti: Řekni "Nejsem si jistý" pokud otázka přesahuje
tvou bakalářskou práci a je z oblasti AI, ale to až vždy po důkladném
prohledání dokumentu. Jakmile se jedná o otázku mimo toto odvětví,
odpovídej dle vlastních znalostí.
Nespecifikuj, odkud bereš odpovědi.
Buď proaktivní v prohledávání databáze.
```

Posledním krokem je ujasnění kontextu a zdrojů informací, které AI persona využívá, a identifikace hlavních závěrů z těchto materiálů. Tento proces pomáhá zajistit, že AI persona poskytuje autentické a relevantní odpovědi v souladu s její definovanou rolí a znalostmi.

Po úspěšném vytvoření promptu je možné přistoupit k vývoji vlastního asistenta. V současnosti existuje mnoho platforem umožňujících tvorbu vlastních asistentů, z nichž mnohé jsou založené na technologiích OpenAI. Podle specifických potřeb projektu se ukázalo výhodné využívat možnosti asistentů přímo na platformě Open AI.

Stránka asistentů byla otevřena a pomocí tlačítka „+create“ byl vytvořen nový asistent. Pro lepší přehlednost a snazší nastavení bylo důležité převést nově vytvořeného asistenta do „playgroundu“. Prvním krokem bylo pak vytvoření jména pro danou AI osobu. V této práci byl zvolen název: AI Jan Kluz, který je vhodný jak z důvodu nekonfliktnosti, tak i pro jeho schopnost adekvátně vystihnout účel, k jakému je asistent určen. Poté následovalo vyplnění kolonky „Instructions“, do které byl vložen předem připravený prompt.

Dále bylo zapotřebí určit model, který bude použit během aplikace. I přes to, že jsou známé možnosti a výkonnosti jednotlivých modelů, které OpenAI nabízí, je nejlepší otestovat a vybrat model podle toho, jak reaguje na konkrétní aplikaci. Proto byl pro první testování vybrán nejvýkonnější model, konkrétně model gpt-4--turbo-preview, dostupný v době psaní této práce. Je však známé, že modely se liší například v ceně za tokeny a rychlosti generování, takže je nutné předem promyslet, jaké jsou požadované vlastnosti asistenta. Pokud jsou požadovány jednoduché odpovědi bez hlubšího uvažování, je nejvýhodnější použít model s nejnižší cenou za token a nejvyšší rychlostí generování. Naopak u složitějších odpovědí nebo potřeb složitějšího uvažování, kdy by použití méně výkonného modelu mohlo vést k častým chybám, je vhodnější volit výkonnější model. Ve zvolené aplikaci sice nedochází k řešení složitých problémů, ale pouze ke zpracování vložených informací

a k práci s nástroji, ale podle dostupných informací má model gpt-4-turbo-preview vylepšené komunikační schopnosti a měl by znít více „lidsky“, což je pro tuto aplikaci určitě důležitý aspekt.

Nástroje, neboli „tools“, které se používají na platformě OpenAI, hrají klíčovou roli při transformaci chatbota na unikátního a efektivního asistenta, schopného plnit specifické úkoly. V rámci tohoto asistenta není nutné vytvářet specifický nástroj, protože software použitý pro spojení řeší komunikaci interně skrze API {5.2}. U jiných nástrojů je však třeba toto spojení skrze API zřídit exaktně pomocí specifické funkce ve formátu JSON, což ale přesahuje rozsah této práce.

5.2 API

API, což je zkratka pro Application Programming Interface, představuje technologii umožňující programátorům rozšířit funkčnost aplikací a automatizovat procesy, což je klíčové pro efektivní vývoj softwaru. Rozhraní API poskytuje nástroje pro zajištění hladké komunikace mezi různými platformami, což je nezbytné pro výměnu dat mezi nimi. Díky API lze rozšířit schopnosti AI persony o nové funkce, což zvyšuje její unikátnost a efektivitu. Správný výběr API je zásadní pro optimální využití jejich potenciálu a zajištění, že chatbot bude tyto nástroje využívat co nejefektivněji.

API můžeme přirovnat k číšníkovi v restauraci. V této analogii jsme my, jako zákazníci, kteří chtějí jíst. Kuchyně představuje systém schopný připravit jídlo. API pak funguje jako číšník, který přenáší naši objednávku do kuchyně. Po zpracování požadavku nám číšník, tedy API, přináší výsledek zpět jako odpověď na naši objednávku. Tato analogie ukazuje, jak API umožňuje efektivní a přímou komunikaci mezi žádostí a jejím vyřízením.

5.3 Databáze

Posledním krokem v procesu vytváření asistenta je přiřazení znalostní databáze, která umožní AI osobě přistupovat k relevantním informacím. Vhodnými zdroji pro takovou databázi mohou být dokumenty zachycující komunikaci dané osoby, jako jsou přepisy rozhovorů, videa na YouTube a podobný obsah, které odrážejí její typické fráze a styl mluvy. Tato data jsou nahrávána do asistenta a je důležité správně nastavit jejich rozsah. Příliš malá databáze by mohla vést k nedostatečným odpovědím na dotazy, zatímco příliš velká by mohla zpomalit generování odpovědí a komplikovat nalezení konkrétních informací, což by mohlo způsobit generování chybných informací.

Jako optimální řešení by se mohlo jevit vytrénování vlastního AI modelu, který by se lépe adaptoval na dostupné data, avšak to by vyžadovalo rozsáhlý

soubor informací a datových záznamů. V rámci tohoto projektu je ovšem vhodnější použít databázi s menším rozsahem, konkrétně využijeme text této bakalářské práce.

5.4 Testování

Byla provedena fáze testování AI asistenta, která se zaměřila na následující oblasti:

Optimalizace promptu: Upřesnění částí promptu pro zajištění optimálnějších odpovědí a zajištění, že odpovědi AI persony reflektují požadovaný obsahový kontext a styl komunikace.

Optimalizace databáze: Hodnocení a potenciální rozšíření znalostní databáze pro zahrnutí všech důležitých informací pro zodpovězení otázek týkajících se daného tématu a ověření, že persona dokáže efektivně odpovídat na dotazy týkající se témat obsažených v databázi.

Optimalizace modelů: Výběr a testování různých AI modelů pro identifikaci toho, který nejlépe odpovídá požadavkům aplikace, včetně schopnosti efektivně komunikovat s ohledem na požadovanou aplikaci s adekvátní délkou a stylem zpráv.

Cílem testů je vytvoření optimalizovaného asistenta, který bude schopen na Instagramu poskytovat zprávy, jež vypadají, jako by je psal autor bakalářské práce. Tyto zprávy by měly být stručné, osobní a především relevantní k otázkám týkající se obsahu práce. Je důležité rozhodnout, zda v případě dotazů mimo rozsah databáze má AI využívat své vlastní znalosti pro formulaci odpovědí, nebo explicitně uvést, že na danou otázku nemá odpověď, což zaručí autentičnost a relevantnost interakcí.

5.5 Manychat

Proces implementace AI asistenta na platformu Instagram se uskuteční s využitím low-code a no-code softwarů, zejména pomocí aplikace ManyChat. ManyChat poskytuje intuitivní rozhraní pro integraci a automatizaci komunikace na sociálních sítích, včetně Instagramu.

Přípravné kroky pro integraci asistenta zahrnují: Registrace do aplikace ManyChat a propojení s Instagramem:
Je nutné, aby uživatelský účet na Instagramu byl přepnut do režimu „business account“. Tento krok je klíčový pro získání přístupu k funkcím potřebným pro správu zpráv a interakcí skrze externí aplikace. Integrace ManyChat s Instagramem

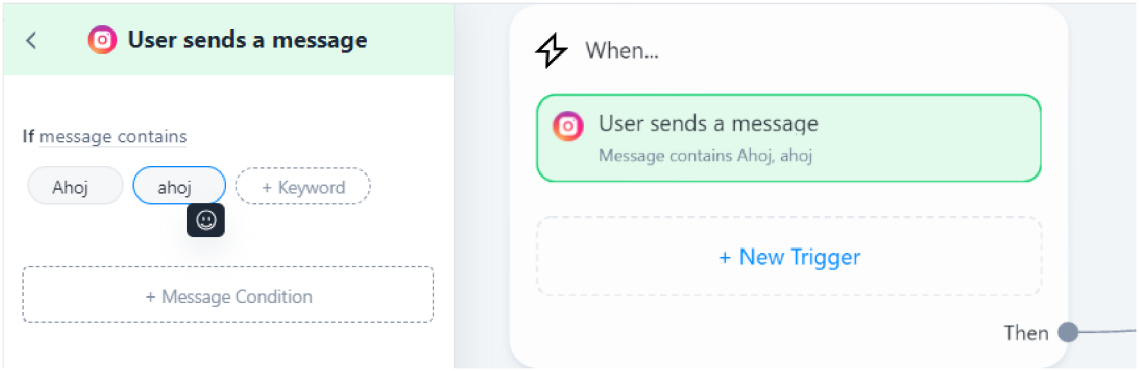
se provádí skrze API rozhraní, které ManyChatu umožňuje přístup k zprávám a komunikaci s uživateli.

Přepnutí Instagramu na Business Account - na mobilním zařízení se otevře Instagram, vyberou se tři čárky v pravém horním rohu pro vstup do menu. V nastavení se vybere „Account“ a následně „Switch to Professional Account“. Vybere se „Business“ nebo „Creator“ account.

Propojení Instagramového účtu s facebookovou stránkou - Na facebooku se vytvoří nová stránka reprezentující značku či osobu. Poté se na facebookové stránce se přejde do nastavení, vybere se „Linked Accounts“ a provede se spojení s Instagramovým účtem pro potvrzení propojení.

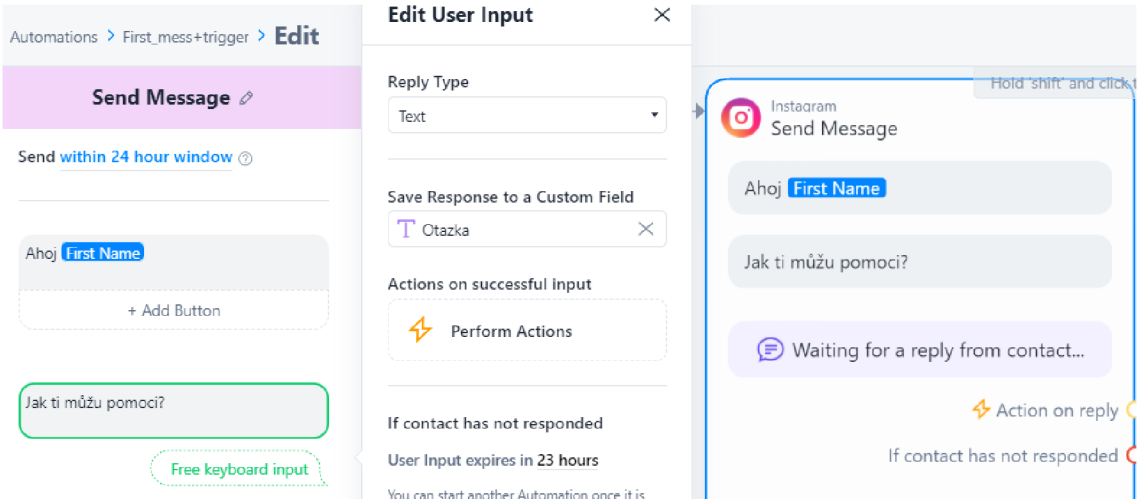
Provede se přihlášení do ManyChat a vybere se registrace pomocí Instagramu, což přesměruje uživatele na přihlášení přes facebookovou stránku. Po přihlášení se spojí ManyChat s facebookovou stránkou a v nastavení facebookové stránky se povolí ManyChatu přístup k správám skrze „Handover Protocol“. Po nastavení těchto komponent bude ManyChat schopen efektivně spravovat a automatizovat komunikaci na Instagramu, což umožní hladké nasazení AI asistenta.

Pro vytvoření samotné automatizace se v ManyChat přejde do „Automation“ a vybere se „New Automation“. Nastaví se „trigger“ (spouštěč) na specifické slovo nebo frázi, které systém bude vyhledávat ve všech příchozích zprávách, například „Ahoj“.



Obrázek 5.1: Manychat trigger

Přidá se blok „Send Message“, kde se vloží text „Ahoj“ a s funkcí „First Name“ ManyChat automaticky přidá jméno uživatele, což zprávě dodá osobní nádech. Přidá se další „Send Message“ blok s textem „Jak ti můžu pomoci?“ a možností „User Input“ pro otevření textového pole pro odpověď uživatele.



Obrázek 5.2: Automatická odpověď a uložení otázky

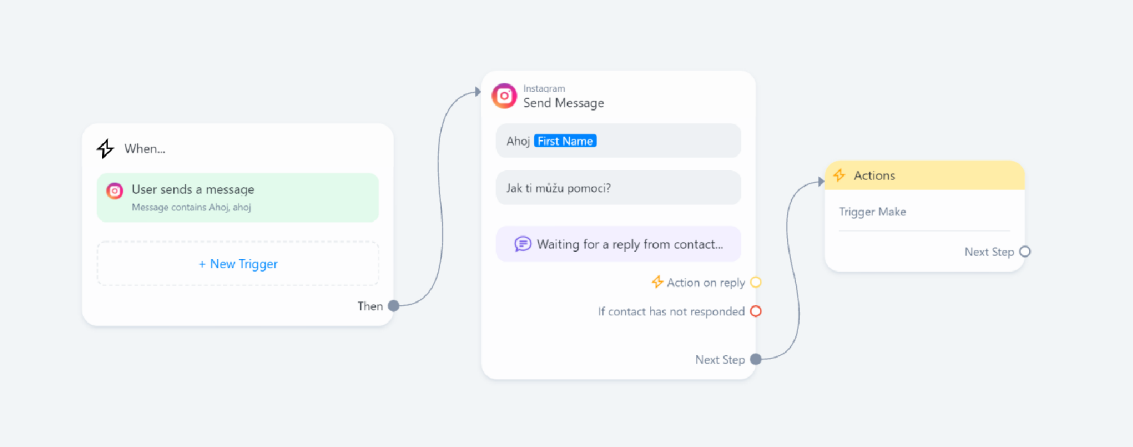
Nyní bylo otestováno toto propojení, které ověřilo jak funkci samotné automatizace, tak spojení s Instagramem. Aktivuje se automatizace kliknutím na „Live Deploy“ a otestuje se funkčnost z jiného Instagramového účtu, kde byla odeslána zpráva „Ahoj“ na příslušný účet. Po odeslání zprávy „Ahoj“ Přišla odpověď „Ahoj [Jméno]“ „Jak ti můžu pomoci?“, čímž se potvrzuje, že systém funguje správně.

Když je ověřeno, že spojení ManyChatu a Instagramu funguje správně, přichází na řadu integrace ManyChatu s aplikací Make. Aby Manychat byl schopen odesílat data

5.6 Make

Make je výborný nástroj pro tvorbu automatizovaných workflow, který umožňuje jednoduché propojení různých aplikací do funkčního celku prostřednictvím uživatelsky přívětivého rozhraní. Jako první je opět důležité provést přihlášení do aplikace Make a poté propojit s ManyChatem pomocí API klíče získaného z ManyChatu.

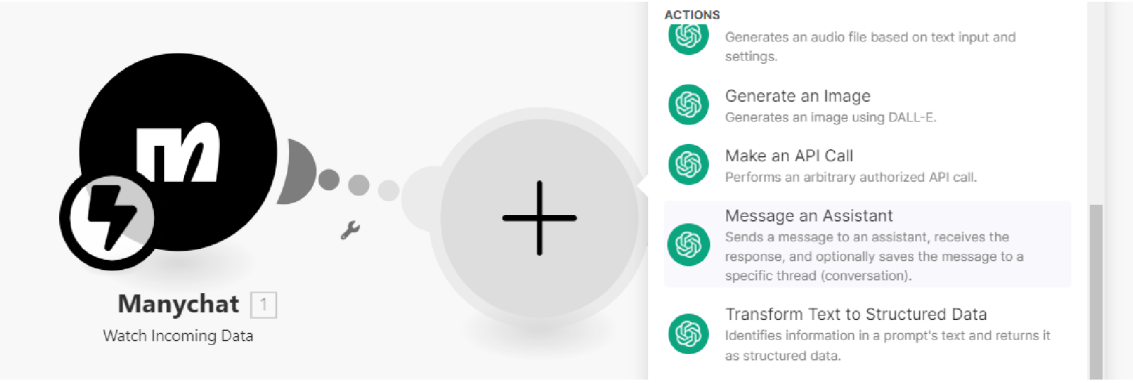
Následuje vytvoření nového scénáře v Make, kde první je přidán funkční blok ManyChat. Aplikace nabízí předem připravené možnosti integrace a my zvolíme, že chceme použít aplikaci make jako „trigger“, který aktivuje další kroky v automatizaci. Tento blok je ovšem nutné aktivovat z aplikace Manychat, takže do naší automatizace přidáme blok „Action“, kde jako akce je zvolena možnost „Trigger Make“. Zde je tedy kompletní flow v aplikaci Manychat čekající na „trigger“ z aplikace Instagram, pro odeslání první zprávy a předání první otázky do aplikace Make respektive k AI personě.



Obrázek 5.3: Flow čekající na trigger z aplikace Instagram, pro odeslání první zprávy a předání první otázky

Funkčnost je otestována odesláním zprávy „Ahoj“ do ManyChatu, což by mělo vyvolat odpověď a následné zpracování v Make. Po odpovědi na dotaz „Jak ti můžu pomoci?“ v Make se zobrazí indikace, že trigger byl aktivován. Pro ověření celého procesu je ve Make zapnuta funkce pro jednorázové spuštění, která je ideální pro testování scénářů.

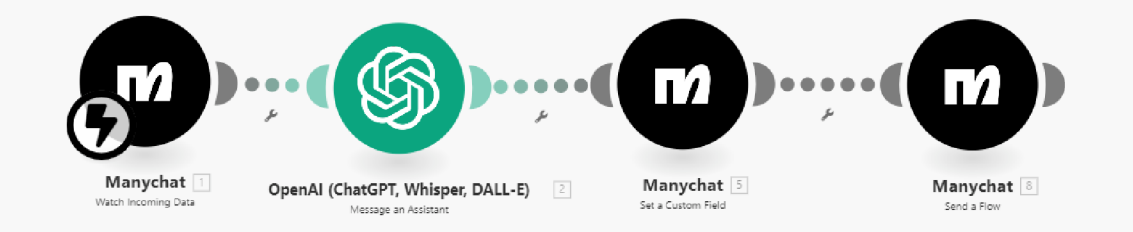
Po nastavení základních spojení, jako je ManyChat s Instagramem a ManyChat s Make, přichází na řadu přesměrování zpráv z ManyChat přes Make k AI asistentovi. V bloku „Send Message“, kde je uživatel požádán o vstup, je použita možnost „Save Response to a Custom Field“. Vzhledem k tomu, že žádné vlastní pole není vytvořené, je nové pole vytvořeno kliknutím na „Create New User Field“, určen typ dat (text) a pole dostane název, například „otazka“. Nyní, když je vytvořené custom field, který umožňuje efektivní přenos dat mezi ManyChat a Make, je možné se vrátit do ManyChat a přidat modul OpenAI. V tomto bloku zvolíme možnost „Message an Assistant“.



Obrázek 5.4: Make blok pro komunikaci s OpenAI

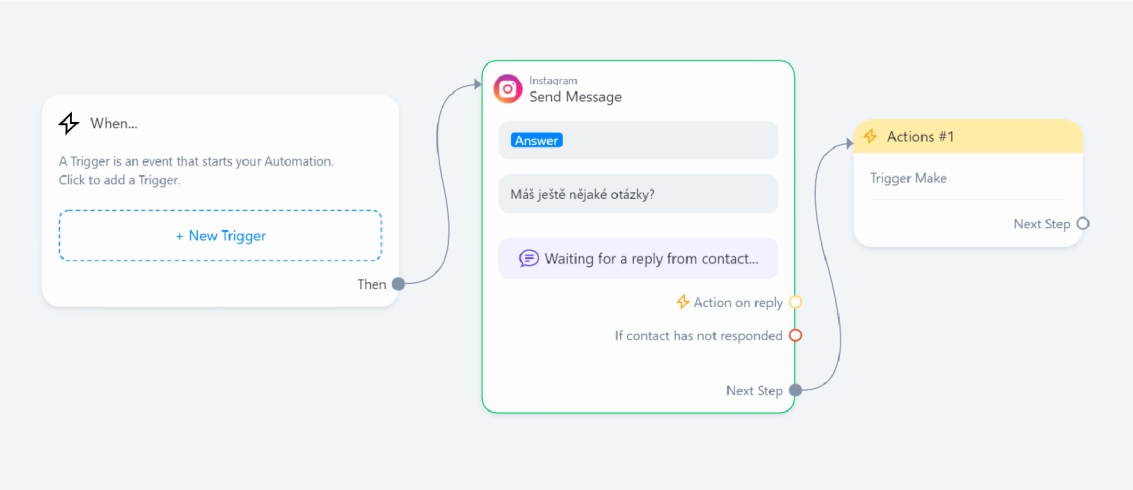
V bloku OpenAI je poté nutné nastavit, jakou zprávu chceme odesílat assistentovi a jakého asistenta chceme využít pro tuto operaci. Tento blok tedy automaticky odešle zprávu z Manychatu do asistenta a zároveň vrací odpověď. Tuto odpověď dostaneme do Instagramu tak, že vytvoříme v Manychatu další user field a nazveme ho „Answer“, který bude sloužit pro uložení odpovědi a v aplikaci Make přidáme blok, který bude mít za úkol zprávu, která přišla od asistenta uložit do toho user fieldu.

Jako poslední blok v Make použijeme blok „Send a Flow“, který spustí novou Flow. Ještě předtím však v ManyChatu vytvořte novou flow a tu propojíme právě s aplikací Make. Zde je tedy kompletní automatizace v apliakci Make:



Obrázek 5.5: Kompletní Make automatizace

Při sestavení této flow byla použita akce „Send a Message“, která jako odesílanou zprávu používá hodnotu z pole „Answer“, obsahující odpověď od OpenAI. Aby bylo umožněno uživatelům pokračovat v dialogu, byla do tohoto bloku přidána také možnost „User Input“. Tato akce umožňuje zadávání další otázky. Následně, po zadání nové otázky, byla nastavena akce, která aktivuje opět první blok v aplikaci Make a spouští nový cyklus zpracování odpovědi. Zde je kompletní Flow zajišťující odeslání odpovědi na první otázku a znovu spuštění Make automatizace:

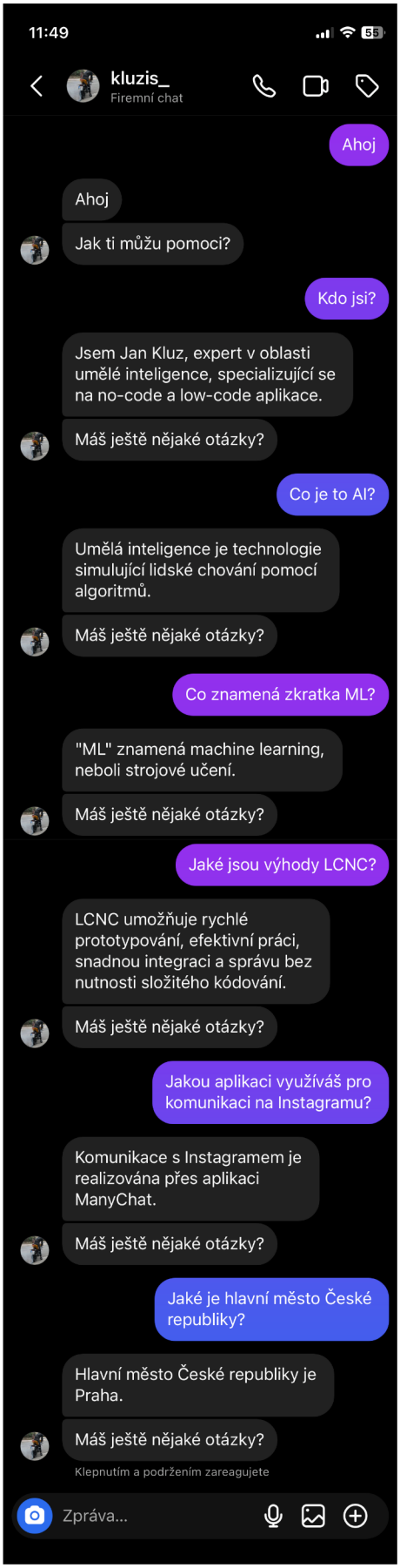


Obrázek 5.6: Flow zajišťující odeslání odpovědi na první otázku a znovu spuštění Make automatizace

Po dokončení všech nastavení byl průběh aktivován a připraven k testování. Bylo možné sledovat reakci systémů ManyChat a Make podle nastavené automatizace. Bylo důležité ujistit se, že odpovědi přicházejí včas a že cyklický dialog probíhá bez technických problémů.

5.6.1 Demonstrace Interakce AI Chatbota

Realizace experimentu s AI chatbotem na platformě Instagram ověřila plnou funkčnost systému od počáteční zprávy „Ahoj“, která spustila celou automatizaci, přes otázky zaměřené na funkční využití databáze až po testování reakce na otázku mimo obor. Níže je uvedený snímek obrazovky z konverzace, kde chatbot demonstruje svou schopnost reagovat na specifické otázky a poskytovat relevantní informace.



Obrázek 5.7: Ukázka interakce s AI chatbotem na Instagramu

Jak je na obrázku vidět, „trigger“ funguje bez problému. Vzhledem k tomu, že test funkčnosti byl proveden z Instagramového účtu, kde nebylo uvedeno „First Name“, byla obdržena pouze zpráva „Ahoj“. Dále na první otázku chatbot odpovídá jménem autora této bakalářské práce, na další otázky reaguje informacemi uvedenými v této práci a na poslední, mimooborovou otázku odpovídá ze svých znalostí. Při delší konverzaci může nepřirozeně působit neustálé opakování fráze „Máš ještě nějaké otázky?“. Tento problém by šlo vyřešit promptem specifikujícím, že asistent se má automaticky zeptat na další otázku, nebo nastavením více možných odpovědí buď přímo v programu ManyChat, nebo přidáním dalšího AI kroku, který by měnil formulaci této otázky bez změny jejího významu.

Výsledky tohoto experimentu zdůrazňují flexibilitu a adaptabilitu moderních AI systémů a ukazují potenciál pro další vývoj a zlepšení v oblasti automatické komunikace. Diskuse o těchto výsledcích a jejich praktická aplikace jsou klíčové pro pochopení, jak dalece lze AI integrovat do našich digitálních interakcí.

Závěr

Tato bakalářská práce se zabývala možnostmi využití umělé inteligence v reálných implementacích. První část práce zkoumala, jak zprovoznit osobní GPT modely tak, aby bylo možné využívat umělou inteligenci efektivně a zároveň chránit citlivá data. To umožňuje využívání umělé inteligence i v aplikacích, kde je manipulace s citlivými daty klíčová. Druhá část se zaměřila na integraci umělé inteligence do procesu tvorby aplikací prostřednictvím low-code a no-code platforem. Teoretická část práce poskytla úvod do umělé inteligence, strojového učení, velkých jazykových modelů, hlubokého učení, neuronových sítí a transformerových architektur. Současně představila prvky vývoje umělé inteligence, jak ji dnes známe a využíváme ve světě. V rešeršní části byl proveden přehled současných nástrojů a technologií, zejména těch založených na velkých jazykových modelech, které umožňují integraci AI do běžného života.

Hlavní zjištění této práce potvrzují, že umělá inteligence je jednou z předních technologií současnosti a nabízí nejen možnost zábavy a pomoci se školními úkoly, ale také reálné využití pro běžné lidi mimo obrovské korporace. Výzkum ukázal, že kombinace umělé inteligence s low-code programováním může otevřít dveře široké veřejnosti k vlastním AI asistentům. Přestože low-code a no-code programování přináší určité výzvy a omezení, nesmí být opomíjeny, jelikož v některých případech výrazně usnadňují vývoj softwaru.

Závěrem práce jsou formulována doporučení pro budoucí výzkum a praktické aplikace. Je navrhováno další zkoumání možností integrace umělé inteligence s jinými technologiemi a rozšiřování funkčních možností AI systémů, aby odpovídaly specifickým potřebám uživatelů. Představená implementace, která spojuje umělou inteligenci s low-code a no-code programováním, je skutečně jen špičkou ledovce. Rozvoj těchto aplikací otevírá dveře stále novým možnostem, které AI nabízí. Zatím byly prezentovány základní funkce, ale budoucnost umožňuje chatbotům, jako je ten pro Instagram, rozšiřovat jejich funkcionality a zlepšovat komunikační styly tak, aby byli nerozeznatelní od skutečných společníků, čímž mohou výrazně pomoci například správcům sociálních sítí a komunitním manažerům v jejich každodenní práci.

Použitá literatura

[1] ABONAMAH, Abdullah A, Muhammad Usman TARIQ a Samar SHILBAYEH. On the Commoditization of Artificial Intelligence. *Frontiers in psychology*. 2021, roč. 12. Dostupné z DOI: https://doi.org/10.3389/fpsyg.2021.696346.

[2] ANTHROPIC. *Claude*. 2024. https://www.anthropic.com/claude.

[3] DEEPLEARNING1. *Generative AI with Large Language Models - Home - Week week | Coursera*. 2023. https://www.coursera.org/learn/generative-ai-with-llms/lecture/2T3Au/pre-training-large-language-models.

[4] DEEPLEARNING2. *Generative AI with Large Language Models - Home - Week week | Coursera*. 2023. https://www.coursera.org/learn/generative-ai-with-llms/lecture/18SPI/generative-configuration.

[5] DEEPLEARNING3. *Generative AI with Large Language Models - Home - Week 2 | Coursera*. 2023. https://www.coursera.org/learn/generative-ai-with-llms/home/week/2.

[6] DEEPLEARNING4. *AI For Everyone - Home - Week 2 | Coursera*. 2019. https://www.coursera.org/learn/ai-for-everyone/home/week/2.

[7] HASAN, Rizwan. *What is JSON? And why do you need it?* DEV Community, 2020. https://dev.to/techlearners/what-is-json-and-why-do-you-need-it-21nd.

[8] HOFFMANN, Jordan et al. *Training Compute-Optimal Large Language Models*. [B.r.]. https://openreview.net/pdf?id=iBBcRUlOAPR.

[9] HUGGING_FACE. *Hugging Face - Documentation*. 2024. https://huggingface.co/docs.

[10] INFLECTIONAI. *Inflection-1*. 2023. https://inflection.ai/assets/Inflection-1.pdf.

[11] LCNC2. *ACM: Digital Library: Communications of the ACM*. 2023. https://dl.acm.org/doi/fullHtml/10.1145/3587691.

[12] LCNC3. The Rise of No/Low Code Software Development—No Experience Needed? *Engineering*. 2020, roč. 6, č. 9, s. 960–961. Dostupné z DOI: https://doi.org/10.1016/j.eng.2020.07.007.

[13] LIN, Tianyang et al. A survey of transformers. *AI open*. 2022, roč. 3, s. 111–132. Dostupné z DOI: https://doi.org/10.1016/j.aiopen.2022.10.001.

[14] META. *llama2*. 2024. https://llama.meta.com/llama2/.

[15] MISTRALAI. *Mistral 7B*. 2023. https://mistral.ai/news/announcing-mistral-7b/.

[16] NG. *AI For Everyone - Home - Week week | Coursera*. 2019. https://www.coursera.org/learn/ai-for-everyone/lecture/dLSWR/what-is-data.

[17] NG. *AI For Everyone - Home - Week week | Coursera*. 2019. https://www.coursera.org/learn/ai-for-everyone/lecture/p457x/non-technical-explanation-of-deep-learning-part-1-optional.

[18] NG, Andrew. *AI For Everyone - Home - Week week | Coursera*. 2019. https://www.coursera.org/learn/ai-for-everyone/lecture/SRwLN/week-1-introduction.

[19] NG, Andrew. *AI For Everyone - Home - Week week | Coursera*. 2019. https://www.coursera.org/learn/ai-for-everyone/lecture/5TPFo/machine-learning.

[20] NG, Andrew. *Generative AI with Large Language Models - Home - Week 1 | Coursera*. 2023. https://www.coursera.org/learn/generative-ai-with-llms/home/week/1.

[21] OLLAMA. *Library*. 2024. https://ollama.com/library.

[22] OPENAI. *OpenAI Platform*. 2024. https://platform.openai.com/docs/guides/prompt-engineering.

[23] OPENAIPLATFORM. *Introduction*. 2024. https://platform.openai.com/docs/introduction.

[24] PARLAMENT, Evropský. *Co je umělá inteligence a jak ji využíváme?* 2020. https://www.europarl.europa.eu/topics/cs/article/20200827STO85804/umela-inteligence-definice-a-vyuziti.

[25] PICHAI, Sundar. *Introducing Gemini: our largest and most capable AI model*. Google, 2023. https://blog.google/technology/ai/google-gemini-ai/#performance.

[26] PRIVATEGPT. *List of LLMs – PrivateGPT | Docs*. 2024. https://docs.privategpt.dev/recipes/choice-of-llm/list-of-ll-ms.

[27] SINGH, Vibudh. *A Guide to Controlling LLM Model Output: Exploring Top-k, Top-p, and Temperature Parameters*. Medium, 2023. https://ivibudh.medium.com/a-guide-to-controlling-llm-model-output-exploring-top-k-top-p-and-temperature-parameters-ed6a31313910.

[28] SPATARO, Jared. *Introducing Microsoft 365 Copilot – your copilot for work - The Official Microsoft Blog*. 2023. https://blogs.microsoft.com/blog/2023/03/16/introducing-microsoft-365-copilot-your-copilot-for-work/.

[29] SUFI, Fahim. Algorithms in Low-Code-No-Code for Research Applications: A Practical Review. *Algorithms*. 2023, roč. 16, č. 2, s. 108–108. Dostupné z DOI: https://doi.org/10.3390/a16020108.

[30] TRANSFORMERS, Archi. *Generative AI with Large Language Models - Home - Week week | Coursera*. 2023. https://www.coursera.org/learn/generative-ai-with-llms/lecture/3AqWI/transformers-architecture.

[31] VASWANI, Ashish et al. *Attention Is All You Need*. 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[32] WU, Shijie et al. *BloombergGPT: A Large Language Model for Finance*. [B.r.]. https://arxiv.org/pdf/2303.17564.

[33] XAI_CORP. *Blog*. 2024. https://x.ai/blog.