Tytuł: SMIS

(strona tytułowa do wstawienia z Moja PG)

Kamil Kozioł Alicja Wagner Kacper Chodubski Kacper Konopka

18 października 2024

STRESZCZENIE

Praca skupia się na zagadnieniu wykorzystania modeli języka naturalnego. Większość obecnego rynku skupia się na bezpośrednim wykorzystaniu modeli do miedzy innymi pozyskiwania informacji, redagowania tekstów i tym podobnym. Nasz projekt wykorzystuje model jako narzędzie do imitowania ludzkiego postrzegania świata. Model jest wykorzystywany jako ”osoba” podejmująca decyzje na wzór tych człowieka. Żeby móc zaprezentować działanie tego mechanizmu potrzebowaliśmy zbudować wirtualne środowisko, gdzie agenci, czyli nasz odpowiednik wirtualnego człowieka, będą podejmowali działania zaplanowane i zarządzane przez model językowy. Pozwoli nam to zobaczyć jak dużo wiedzy na temat realnego świata jest zawarte w zbiorach danych używanych przez twórców tych modeli oraz jak dobrze odzwierciedlają one zachowanie ludzi.

Słowa kluczowe: Agent, LLM
Dziedzina nauki i techniki, zgodnie z wymogami OECD:
Nauki o komputerach i informatyka

ABSTRACT

Abstract in english.

Keywords: Keyword 1, Keyword 2, Keyword 3
Field of science and technology in accordance with OECD requirements:
Computer Science and Information technology

SPIS TREŚCI

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

SMIS - Social Modeling and Interaction Simulator – nazwa własna systemu, który został zrealizowany w ramach projektu dyplomowego inżynierskiego.

AI - (ang. *Artificial Intelligence*) – sztuczna inteligencja.

LLM - (ang. *Large Language Model*) – duży model językowy.

RAG - (ang. *Retrieval Augumented Generation*) – technika generowania tekstu przy użyciu zewnętrznych źródeł danych.

NLP - (ang. *Natural Language Processing*) – przetwarzanie języka naturalnego.

1. WSTĘP I CEL PRACY

W ostatnich latach coraz częściej zaczęto wykorzystywać duże modele językowe (ang. *LLM, Large Language Model*) w aplikacjach obejmujących różne sektory. Duże zainteresowanie sztuczną inteligencją zaowocowało jej szybkim rozwojem, co sprzyjało powstawaniu nowych pomysłów dla zastosowań przetwarzania języka naturalnego.

Powstała także technika Retrieval-Augumented Generation (RAG), dzięki której możliwe jest podanie modelowi językowemu kontekstu, co zapobiega halucynacjom. Za jej pomocą można odwoływać się do poprzednich wiadomości lub zasilić LLM danymi, do których wcześniej nie miał dostępu. Otwiera to nowe możliwości tworzenia tekstu na podstawie konkretnych treści. Ponadto, jest to rozwiązanie o wiele tańsze i szybsze niż dotrenowywanie modeli, nie mówiąc już o trenowaniu ich od podstaw.

Symulowanie ludzkich zachowań staje się coraz bardziej realistyczne dzięki zastosowaniu innowacyjnych technologii. Można upodabniać zachowanie modeli do zachowania człowieka poprzez odwzorowywanie czynności takich jak percepcja, zapamiętywanie, planowanie i rozumowanie. Technika RAG ułatwia tworzenie angażujących i realistycznych scenariuszy.

1.1. Motywacja

1.1.1. Dynamiczny rozwój dziedziny

W przeciągu ostatnich kilkudziesięciu lat dało się zaobserwować intensywny rozwój sztucznej inteligencji (ang. *AI, Artificial Intelligence*). Na całym świecie zaczęto prowadzić coraz więcej badań dotyczących sztucznej inteligencji, co ma swoje odzwierciedlenie w statystykach cytowań prac związanych z tą tematyką. Jak pokazują przeglądy bibliometryczne, w latach 2012 - 2022 wzrost liczby publikacji w dziedzinie AI i Big Data był wykładniczy, z wyraźnym przyspieszeniem od 2019 roku[1]. Sztuczna inteligencja zaczęła być wykorzystywana w wielu dyscyplinach, takich jak ochrona zdrowia, edukacja, biznes i zarządzanie, a także turystyka czy rozrywka.

Jednym z podobszarów sztucznej inteligencji jest przetwarzanie języka naturalnego (ang. *NLP, Natural Language Processing*). Dziedzina ta zajmuje się przekształcaniem języka zrozumiałego dla człowieka na taki, który jest zrozumiały dla komputera. Dzięki temu można w stosunkowo łatwy sposób analizować, generować i przetwarzać tekst. Znaczącą rolę odgrywają tutaj także duże modele językowe, które w ostatniej dekadzie dynamicznie się rozwijały. Największą popularność zyskały, gdy w listopadzie 2022 roku firma OpenAI uruchomiła ChatGPT-3.5. Czatbot ten od samego początku cieszył się ogromnym zainteresowaniem. W niespełna tydzień korzystało z niego ponad milion użytkowników[2].

Rozwój, w kontekście wykorzystywania dużych modeli językowych, miał miejsce także w branży gier komputerowych. W dobrze przemyślanym i zbudownaym systemie możliwe jest nawet generowanie nowych poziomów gier wideo. Jednak, jak w wielu aplikacjach wykorzystujących wytrenowane modele, efekty są mocno uzależnione od jakości danych[3]. Wyniki zwracane przez takie systemy odzwierciedlają precyzję przekazywanych do nich promptów. Jeżeli podane będzie zbyt ogólne lub niedokładne polecenie, LLM nie wygeneruje oczekiwanej treści - odpowiedź może

być niepełna lub całkowicie odbiegać od tematu. Takie błędy mogą prowadzić do niezadowolenia graczy i frustracji twórców. Tworzenie obrazów, które są przez ludzi uznawane za realistyczne, czy po prostu ciekawe, jest dla modeli zadaniem niezwykle trudnym. Obecnie dużo lepiej wypada generowanie tekstu. Dostępnych jest coraz więcej LLM-ów, które mają wiele miliardów parametrów, co pozwala im precyzyjnie odpowiadać na pytania i wytwarzać sensowne treści. Dzięki zaawansowanej architekturze, modelowi transformera i ogromnej ilości danych, na których były trenowane, są w stanie wyłapywać szczegóły i dostosowywać się do kontekstu. Sprawia to, że ich odpowiedzi są bardziej trafne i zrozumiałe.

1.1.2. Nowe możliwości

Duże modele językowe

Temat dużych modeli językowych stał się bardzo popularny. Jeszcze kilkadziesiąt lat temu nie do pomyślenia był fakt, że będzie można rozmawiać z komputerem za pomocą języka naturalnego, na dodatek bez wymogu posiadania specjalistycznej wiedzy o sztucznej inteligencji. Teraz każdy człowiek może wybrać jeden z szeroko dostępnych w internecie modeli językowych i przetestować jego możliwości. Przykłady wykorzystania takiego rozwiązania to generowanie tekstu, tłumaczenie treści, podsumowywanie długich dokumentów oraz analiza sentymentu.

Wiele LLM-ów jest oprogramowaniem otwartym (ang. *open-source software*), co ułatwia także wykorzystywanie ich w kodzie własnej aplikacji. Wystarczy wybrać konkretny model, specjalizujący się w zadaniu, do którego ma być zastosowany, i wkomponować jego wywołania w kod źródłowy. Dzięki otwartości oprogramowania nie jest wymagane używanie żadnego klucza API. Nie trzeba nawet wysyłać danych na zewnętrzny serwer - jeśli tylko posiada się własną maszynę z odpowiednimi zasobami pamięciowymi i obliczeniowymi, można umieścić na niej model, który będzie odseparowany od sieci globalnej oraz nieautoryzowanych dostępów. Jest to rozwiązanie szczególnie istotne dla firm posiadających wrażliwe dane, dla których ważna jest prywatność i bezpieczeństwo. Przechowywanie i przetwarzanie danych lokalnie zmniejsza ryzyko wycieku informacji.

Retrieval-Augumented Generation

Nowe możliwości dotyczące NLP otworzyła także technika RAG. Po raz pierwszy została ona opisana przez Patricka Lewisa i jego zespół w 2020 roku[4]. Ta stosunkowo młoda technika pozwala na współpracę z modelem językowym i zewnętrznymi danymi. Można dzięki temu bez trenowania modelu korzystać z treści, których ten model wcześniej nie znał. Jest to ciekawa alternatywa dla znanego już wcześniej procesu trenowania modelu. W technice RAG tkwi ogromny potencjał tworzenia dynamicznych aplikacji, wymagających dostępu do zmieniających się i ciągle rozbudowywanych danych. Kiedy zasoby tak szybko się zmieniają, nieefektywne byłoby trenowanie na nich LLM-u. Po pierwsze proces ten jest bardzo kosztowny i czasochłonny, a trzeba by go było wykonywać bardzo często, aby mieć aktualne dane. Po drugie, tempo zmian danych mogłoby przewyższać tempo samego trenowania systemu, przez co w ogóle nie byłoby możliwe osiągnięcie stanu, w którym dane byłyby w stu procentach aktualne.

RAG ma także inne zalety. Wiąże się on z większą dokładnością odpowiedzi, ogranicza halucynacje i sprawia, że system go wykorzystujący jest bardziej skalowalny. Oprócz pozytywnych cech, RAG wiąże się także z wieloma wyzwaniami. Architektura systemu robi się dużo bardziej skomplikowana niż w przypadku czystego modelu LLM. Ponadto, wyszukiwanie odpowiednich informacji

w dużych bazach danych może wymagać znacznych zasobów obliczeniowych, co może przekładać się na dłuższy czas uzyskiwania odpowiedzi. Ważne jest także odpowiednie przygotowanie kontektu i zarządzanie nim, aby jak najlepiej dostosować go do pytania.

Technika ta z pewnością będzie rozwijana w nadchodzących latach, a wiele osób już teraz bada jej potencjał. Jest to dziedzina, którą warto zgłębiać i być na bieżąco z aktualnymi odkryciami i rozwiązaniami. W świecie, w którym coraz częściej słyszy się o sztucznej inteligencji, trudno zignorować ten temat. Warto zbadać, czy faktycznie duże modele językowe potrafią naśladować ludzi i jak realistyczne będą konwersacje generowane przez system wykorzystujący technikę RAG.

1.2. Cel pracy

1.2.1. Inspiracja

tekst

2. WPROWADZENIE DO DZIEDZINY

tekst

2.1. Generatywna sztuczna inteligencja

tekst

2.2. Duże modele językowe

llm + trenowanie, różnice między llama a llama chat

2.3. Retrieval Augumented Generation

teoria + metody poprawy retrievalu

2.4. Agenty dużych modeli językowych

tekst, agenty llm

3. TECHNOLOGIE, ALGORYTMY I NARZĘDZIA

tekst

3.1. Silnik gry Unity

czemu Unity, zalety, do czego jest wykorzystane, jakie biblioteki, skąd assety itp.

3.2. Python

czemu python, jakie biblioteki, krótkie porównanie z innymi językami

3.3. Wybrane modele językowe

3.3.1. Wybór modeli

jak przebiegał wybór modeli, opensource, opisać wszystkie, których używaliśmy

3.3.2. Wektoryzacja i osadzanie słów

tekst

3.3.3. Podobieństwo cosinusowe

tekst

3.3.4. Zasoby obliczeniowe

opisać, że korzystamy z serwera KASKu, ile VRAMu itp, jak to się przekłada na szybkość działania itd

4. PROJEKT SYSTEMU

tekst, tu już typowo o naszej implementacji (porównywać do papieru), na początku krótki opis że dzieli sie na front i backend i model

4.1. Interfejs gry

cały opis unity, za co odpowiada, jakie są interakcje, animacje

4.2. Architektura serwera i warstwy logicznej

4.2.1. Agenci

Pamięć agentów

Pamięć agentów jest kluczowym elementem ich funkcjonowania. To dzięki niej agenci reagują w sposób bardziej ludzki.

Architektura pamięci agentów

Pamięć opiera się na modelu języka naturalnego typu encoder, który przekształca nowe wspomnienia agenta na wektor. Później, w trakcie interakcji czy budowania nowych wspomnień przez agenta jest wybierane k wspomnień, które uzyskały największą średnią ważoną z trzech metryk: podobieństwa (relevance), ważności (importance) i niedawności (recency). Wybrane wspomnienia są później przekazywane do modelu językowego typu decoder, który na podstawie ich i danego wydarzenia decyduje o zachowaniu agenta.

Metryka podobieństwa

Metryka podobieństwa odpowiada podobieństwu tematycznemu nowego wydarzenia do wydarzenia zapisanego w pamięci agenta. Jest to uzyskiwane za pomocą obliczonej odległości cosinusowej wektorów stworzonych przez model typu encoder.

Metryka ważności

Metryka istoty jest uzyskiwana poprzez ocenę ważności danego wydarzenia w kontekście życia agenta przez LLM. Metryka odpowiada na pytanie jak ważne dla danego agenta w skali od 1 do 10 jest dane wydarzenie np. pościelenie łóżka będzie miało skalę bliską 1, gdzie wzięcie ślubu skalę bliską 10.

Metryka niedawności

Metryka niedawności odpowiada mechanizmowi ”zapominania” przez agenta. Wspomnienia uzyskują wynik zgodnie ze wzorem:

$$score = max\_score * decay\_factor^{time\_diff} \tag{4-1}$$

4.3. System promptowania

Jakość odpowiedzi generowanych przez duże modele językowe w znacznym stopniu zależy od zapytań, które są do nich wysyłane. Tekst przekazywany do modelu nazywany jest poleceniem (ang. *prompt*), którego główną częścią jest pytanie lub zadanie opisujące, co model powinien wykonać. W przypadku techniki RAG w skład polecenia wchodzi także kontekst, czyli tekst ukazujący szerszą perspektywę danego tematu. W samym zadaniu zawarta jest także informacja dla modelu, aby ten, podczas odpowiadania na pytanie lub wykonywania postawionego mu zadania, korzystał bezpośrednio z dostarczonego mu kontekstu - jeżeli jakaś informacja w nim nie występuje, model powinien jasno zakomunikować, że nie jest w stanie odpowiedzieć na pytanie. Nie powinien próbować wymyślać odpowiedzi jedynie na podstawie własnej wiedzy.

4.3.1. Wykonywanie akcji

W aplikacji SMIS, podobnie jak w programie, na którym wzorowana jest jej implementacja [5], każda decyzja dotycząca przebiegu rozgrywki podejmowana jest za pomocą dużego modelu językowego. W ujęciu ogólnym, na podstawie otrzymanych odpowiedzi, wykonywane są odpowiednie akcje. Kiedy rozpoczyna się dzień, dla każdego agenta generowany jest godzinowy plan dnia, bazujący na jego zainteresowaniach i jego trybie życia. Agent przeżywa dzień zgodnie z tym planem. Ponadto, przy każdym spotkaniu z inną osobą, model językowy decyduje, czy agenci powinni nawiązać konwersację. Jeśli decyzja jest pozytywna, tworzona jest konwersacja, która nawiązuje do wspomnień rozmówców oraz jest zgodna z ich osobowościami. Po zakończeniu rozmowy jest ona podsumowywana i generowane są wspomnienia dla obu agentów, które zapisywane są w ich pamięci. Dzięki temu mogą się oni do nich odwoływać w odpowiednich do tego momentach.

Przy tworzeniu polecenia dla modelu, pobierane są informacje i wspomnienia odnoszące się do konkretnego agenta. Żeby proces ten mógł nastąpić, należy podczas zapisywania wspomnienia uwzględnić trzy metryki, opisane szerzej w podrozdziale 4.2. Metryka ważności również opiera się na odpowiedzi od modelu.

4.3.2. Konwersacje

Tworzenie konwersacji jest dość skomplikowanym procesem. Aby jak najbardziej upodobnić agentów do prawdziwych ludzi, generowanie rozmowy podzielone zostało na kilka etapów.

Pierwszym z nich jest decydowanie, czy agent powinien w ogóle podjąć konwersację. Warto zauważyć, że my również nie odzywamy się do każdej miniętej na ulicy osoby - w takiej sytuacji nie udawałoby nam się nigdzie dotrzeć, gdyż cały dzień wypełniony byłby jedynie rozmowami z innymi. Podświadomie oceniamy, czy warto przywitać się z daną osobą. Czynniki, jakie są w takiej

sytuacji brane pod uwagę, to m.in. to, kim jest spotkany człowiek, czy go znamy, co w danym momencie robi, czy nie jest zajęty.

Podobnie działają agenci w aplikacji SMIS.

4.4. Komunikacja między serwisami

jak jest zrobiona komunikacja, jak to działa, może być wiele frontów na jeden serwer

5. BADANIA I ANALIZA WYNIKÓW

tekst

5.1. Ocena świadomości agentów

jak dobrze agenci generują realistyczne odpowiedzi:
- czy agenci potrafią mówić o sobie
- czy pamiętają informacje o sobie, imię czy styl życia
- czy ich wypowiedzi odwołują się do przeszłych wydarzeń

5.2. Porównanie wydajności modeli językowych

- benchmark, ogólna ocena modeli i porównanie
- porównanie różnych modeli LLM używanych w aplikacji (np llama2, llama3)
- testy: czas odpowiedzi, czas generowania, zgodnosc z kontekstem, naturalnosc dialogow, zroznicowanie odpowiedzi

5.3. Ocena jakości konwersacji przez użytkowników

- subiektywna ocena modeli
- ankieta przeprowadzona wśród studentów, którzy ocenią jakość konwersacji generowanych przez różne modele
- która konwersacja była najbardziej naturalna i spójna, + uwagi i sugestie będzie można dodać

5.4. Rozprzestrzenianie się informacji wśród agentów

- jak w papierze
- eksperymenty związane z rozprzestrzenianiem się informacji między agentami
- do ilu agentów dociera informacja gdy na początku znał ją jeden agent (np. planowany ślub albo olimpiada)
- jak ważność informacji wpływa na jej propagację (np ślub i olimpiada ważne więc powinny się rozprzestrzenic, a to ze ktoś zjadł jajecznicę na śniadanie już nie)

5.5. Wnioski i ograniczenia

- wyzwania napotkane podczas pracy z aplikacją związane z modelami
- pewnie nie zawsze będzie się wyszukiwał ddobry kontekst, problemy z retrievalem
- halucynacje

6. PODSUMOWANIE

tekst

6.1. Osiągnięte rezultaty

tekst

6.2. Plany na przyszłość

tekst

WYKAZ LITERATURY

[1] P. V. Thayyib *et al.*, “State-of-the-art of artificial intelligence and big data analytics reviews in five different domains: A bibliometric summary,” *Sustainability*, vol. 15, no. 5, pp. 4026–4026, 2023.

[2] Wikipedia, *Chatgpt*, Accessed: 2024-10-14, 2024. [Online]. Available: `https://pl.wikipedia.org/wiki/ChatGPT`.

[3] G. Todd, S. Earle, M. U. Nasir, M. C. Green, and J. Togelius, “Level generation through large language models,” *Proceedings of the 18th International Conference on the Foundations of Digital Games*, ser. FDG 2023, ACM, Apr. 2023. [Online]. Available: `http://dx.doi.org/10.1145/3582437.3587211`.

[4] P. Lewis *et al.*, *Retrieval-augmented generation for knowledge-intensive nlp tasks*, 2021. arXiv: `2005.11401 [cs.CL]`. [Online]. Available: `https://arxiv.org/abs/2005.11401`.

[5] J. S. Park, J. C. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, *Generative agents: Interactive simulacra of human behavior*, 2023. arXiv: `2304.03442 [cs.HC]`. [Online]. Available: `https://arxiv.org/abs/2304.03442`.

WYKAZ RYSUNKÓW

WYKAZ TABEL