DOI: 10.35191/medsoft_2020_1_32_22_26

DETEKCE LÉKAŘSKÉ DIAGNÓZY, KATEGORIZACE A NORMALIZACE NESTRUKTUROVANÝCH LÉKAŘSKÝCH ZÁZNAMŮ POMOCÍ AI

Stanislav Jirák, Tomáš Kulhánek

Abstrakt

Hlavním problémem digitalizace zdravotních záznamů je nestrukturovaná povaha mnoha lékařských záznamů, které jsou často zaznamenávány jako volný text, což komplikuje jejich strojové zpracování. V rámci akce CEE Hackathon 2023 v pražském IKEMu byla zkoumána možnost konverze digitálních nestrukturovaných záznamů do strukturované formy pomocí umělé inteligence. Byl vyvinut prototyp, který používá metody AI pro kategorické hodnocení zpráv a umožňuje dotazování v přirozeném jazyce. Klíčovou metodou pro analýzu textů bylo využití předtrénovaných modelů GPT (Generative Pre-trained Transformer) od OpenAI, které byly instruovány k identifikaci specifických diagnóz, jako je diabetes mellitus. Výsledky ukázaly, že umělá inteligence může efektivně porozumět lékařským textům a má potenciál k dalšímu vylepšení pro rozlišování mezi přesnými diagnózami, jako je diabetes a pre-diabetes. Porovnání s lidskými hodnotiteli ukázalo, že AI může být stejně nebo více přesná ve svých posudcích. Navíc programátorský přístup a zpracování některých rozsáhlých databází umělou inteligencí umožní datovou analýzu, která pomocí klasických metod, jako jsou regulární výrazy nebo fulltextové vyhledávání, je obtížná až nemožná.

Klíčová slova

digitalizace zdravotních záznamů, nestrukturovaná data, velké jazykové modely, standardizace záznamů

Úvod

Výzvy v digitalizaci nestrukturovaných zdravotních záznamů

Digitalizace zdravotních záznamů je základním kamenem pro modernizaci zdravotních systémů po celém světě. Elektronické zdravotní záznamy (EHR) mají za cíl zefektivnit správu pacientských dat, zlepšit kvalitu péče a usnadnit sdílení informací mezi poskytovateli zdravotní péče. Přechod z papírových na digitální záznamy však odhalil významnou výzvu: nestrukturovanou povahu velkého množství stávající lékařské dokumentace.

Historicky lékaři zaznamenávali informace o pacientech ve volných textových formátech, od ručně psaných poznámek po psané vyprávění. Tyto záznamy obsahují bohaté detaily o setkáních s pacienty včetně příznaků, diagnóz, léčebných postupů a individuálních příběhů pacientů, ale nedodržují standardizovaný formát. V důsledku toho je často obtížné přistupovat k informacím a analyzovat je na velké škále. Důležitá data mohou být pohřbena ve volném textu, což ztěžuje extrakci relevantních detailů. To samé platí i o již digitálně zaváděných záznamech, typicky obsahujících políčko volného textu.

Nedostatek struktury v těchto záznamech představuje několik výzev pro digitalizační úsilí.

1. brání interoperabilitě systémů EHR, protože výměna informací mezi různými platformami může vést k významné ztrátě dat nebo mylné interpretaci.
2. získání konkrétních informací pro podporu klinického rozhodování nebo výzkum se stává obtížným a časově náročným úkolem, čímž se podkopává efektivita, kterou mají EHR zavést.
3. nestrukturovaná data komplikují dodržování právních a regulačních norem, které často vyžadují specifické dokumentační postupy.

Kromě toho, přechod na digitální záznamy vytvořil poptávku po sofistikovaných nástrojích schopných "chápat" kontext a nuance obsažené v lékařských textech. Tradiční technologie zpracování přirozeného jazyka (NLP) učinily v této oblasti pokroky, ale často se setkávají s omezeními ve své schopnosti plně porozumět a přesně extrahovat klinické informace.

Složitost lékařské terminologie, variabilita v projevu mezi poskytovateli zdravotní péče a jemné implikace zakódované v klinickém žargonu všechny přispívají k obtížím transformace nestrukturovaných záznamů do použitelných digitálních dat. Tato výzva tvoří základ pro inovativní přístupy využívající pokročilou umělou inteligenci, zejména Velké jazykové modely (LLM) jako je série Generative Pre-trained Transformer (GPT), vyvinutá OpenAI, nebo Med-PaLM-2 od Googlu. Nejen pro digitalizaci textu, ale také pro extrakci a strukturování životně důležitých klinických informací. Vývoj a implementace takových řešení poháněných AI stojí v čele překonávání bariér spojených s digitalizací nestrukturovaných lékařských záznamů.

Význam standardizace textu

Jelikož zdravotnický průmysl prochází digitální transformací, stává se stále kritičtější potřeba převést volně psané záznamy do formátu zpracovatelného stroji. Tradiční metody, které zahrnují rozsáhlé množství narativního textu, kdysi sloužily jako důkladné záznamy péče o pacienta. Avšak v době, kdy data nejsou pouze záznamem, ale také komoditou a nástrojem pro zlepšení péče o pacienty, jsou omezení volného textu zřetelnější než kdy dříve, zejména pro účely datové analýzy a strojového učení.

Níže jsou uvedeny potenciální benefity standardizace zdravotních záznamů a jejich transformace do semi-strukturované podoby:

1. **Zlepšená dostupnost dat:** Strukturované formáty dat umožňují rychlý, vyhledatelný přístup k specifickým informacím, což je u volných textových záznamů nepraktické.
2. **Vylepšená interoperabilita:** Standardizovaný formát lékařských záznamů je základem pro interoperabilitu napříč různými zdravotnickými systémy a institucemi.
3. **Lepší podpora klinického rozhodování:** Data zpracovatelná stroji mohou být integrována se systémy podpory klinického rozhodování (CDSS), které poskytují zdravotnickým pracovníkům specifické hodnocení nebo doporučení pro pacienty, aby pomohly v klinickém rozhodování.
4. **Škálovatelnost analýzy dat:** Strukturovaná data umožňují analýzu ve velkém měřítku, což může vést k lékařskému pokroku a lepším výsledkům veřejného zdraví. Zejména pak v oblasti vzácných onemocnění.
5. **Optimalizace fakturace a kódování:** Fakturace a kódování v medicíně závisí na přesné dokumentaci setkání s pacienty.
6. **Právní integrita:** Dodržování právních a regulačních norem, včetně těch týkajících se ochrany soukromí pacientů a bezpečnosti dat (jako je GDPR), je snazší se strukturovanými daty.

V následujícím textu uvádíme konkrétní příklad metody, jak za pomoci velkých jazykových modelů strojově porozumět informacím v lékařském záznamu, jak je standardizovat a vyčíst požadovanou informaci, např. diagnózu pacienta.

Metody

Hlavním cílem během dvoudenního CEE Hackathonu 2023 v pražskému IKEMU bylo vytvořit prototyp systému založeném na AI, který by byl schopný extrahovat lékařskou diagnózu ze zdravotních záznamů pacienta a zároveň, aby se poskytnutá data nestala součástí trénovacích množin nebo nebyla uložena poskytovatelem přístupu k umělé inteligenci.

Anonymizovaná data se zprávami o pacientech byla uložena v SQLite3 databázi. Každá aktualizace záznamu představovala nový řádek a ke každému pacientovi tak existovalo několik řádků záznamů podle vývoje jeho stavu při sledování v ambulancích. Tím vznikla potřeba záznamy konsolidovat a poté nějak standardizovat. Konkrétně deduplikovat a sloučit významné informace do jednoho záznamu a zároveň různý styl záznamů přeformulovat do jednotného stylu.

1 Příprava dat

První fáze zahrnovala načtení exportu SQLite3 databáze s lékařskými zprávy o celkovém počtu 50,000 záznamů. Z celkového počtu cca 10,000 unikátních pacientů bylo náhodně vybráno 100 pro účely rapidního vývoje prototypu. Během tohoto procesu byly dodržovány předpisy o ochraně osobních údajů.

Příprava dat byla prováděna rutinami ve skriptech většinou za pomocí knihoven jazyka Python a dokumentována v interaktivním zápisníku, tzv. Jupyter notebook [9], z něhož je možné postup opět zreprodukovat.

2 Konsolidace záznamů

Ke konsolidaci postupně aktualizovaných záznamů byl použit velký jazykový model (LLM) GPT-4-turbo od OpenAI. Sumarizací za použití jednoduché šablony zároveň došlo ke standardizaci záznamů do jednotné podoby. Model byl také dle instrukcí schopný poskytnout užitečné informace, jako navržený postup léčby a možné lékové interakce.

Pro přístup k velkému jazykovému modelu GPT4 lze použít otevřené API společnosti OpenAI a knihovnu pro jazyk Python 'openai' [4]. Takto je zaručeno, že data v dotazech se neuchovávají pro další zlepšování chodu velkého modelu, tj. tato data se berou jako citlivá a neslouží k dalšímu tréningu dalších verzí modelu společnosti OpenAI ani ke sdílení těchto dat jiným subjektům. Dotaz a data v něm obsažená je po zpracování smazán.

V následující ukázce kompletního programu v jazyce Python lze poslat zadání, tzv. prompt umělé inteligenci za jazykovým modelem GPT-4 Turbo a jeho výsledek, tj. odpověď se vypíše na obrazovku:

```python
import openai
import sys
import os
sys.stdout.reconfigure(encoding='utf-8') # Allow Czech or other language
prompt = sys.argv[1]                     # Get first argument as prompt
chat = {"role": "You are factual assistant and answers are brief.",
        "prompt":prompt}                 # Settings for chat
openai.api_key = os.getenv("OPENAI_API_KEY") # client key to access paid API
client = openai.OpenAI()                 # declare Chat with openai API
completion = client.chat.completions.create( # creates struct to OpenAI API
  model="gpt-4-1106-preview",
  messages=[{"role": "system", "content": chat["role"]},
            {"role": "user", "content": chat["prompt"]}])
response = completion.choices[0].message # sends prompt and waits for response
print(response)                          # prints response
```

Obrázek 1 – openaitest.py

Program lze pak spustit s dotazem, který v sobě obsahuje část nebo celou lékařskou zprávu v přirozeném jazyce tak, jak je uložena ve volném textu v informačním systému:

```
python openaitest.py "Z následujícího textu urči zda pacient trpí diabetes mell
itus: RA: matka osteoporóza, otec DM 2.typu\r\nPA: úředník\r\nOA: kontrolovaná
hypertenze, glukozová intolerance\r\nod 2012 na dietě a metforminu, dobrá glyke
mická kontrola\r\noperace: laparoskopická cholecystektomie v roce 2010\r\nAT: n
epije alkohol, kouření 5 cigaret denně"
```

Obrázek 2 – openaitest prompt

Pokud je nastaven klíč OPENAI_API_KEY a uživatel má tuto službu předplacenou, pak odpověď většinou přijde během několika sekund např. jako:

```
ChatCompletionMessage(content='Ano, pacient trpí diabetes mellitus typu 2. V te
xtu je uvedeno, že od roku 2012 je na dietě a metforminu, což je lék běžně před
episovaný pro léčbu diabetes mellitus 2. typu, a má dobrou glykemickou kontrolu
, což ukazuje na správu hladiny glukózy v krvi související s diabetem.', role='
assistant', function_call=None, tool_calls=None)
```

Obrázek 3 – openaitest odpoved

V tomto případě není v záznamu vyloženě tvrzení, že pacientovi byl diagnostikován diabetes mellitus, ale indicie jako lék a laboratorní kontrola tomu svědčí. GPT-4 v tomto případě „porozuměl" konkrétnímu lékařskému záznamu v českém jazyce a dal i odpověď včetně vysvětlení.

3 Standardizace záznamů

Dalším krokem je vytvořit tzv. standardizovaný konsolidovaný záznam. Projdou se postupně všechny záznamy, kde k jednomu pacientovi může být více záznamů a vytvoří se tzv. standardizovaný záznam bez duplicit ke každému pacientovi s kompletní anamnézou, diagnózou, případnými dalšími výsledky. Ukázka kódu včetně promptu, který navíc používá knihovnu langchain může vypadat např takto:

```python
from langchain import PromptTemplate, LLMChain
from langchain.chat_models import ChatOpenAI
import os

async def askGPT(record):
  template = """
             You are given a medical records about the same patient in Czech.
             Those records are typically the same one, just being updated and changed over the time.
             Your goal is to consolidate and summarize the reports into a single one.
             Do not leave out any important information from the medical stand point of view.
             Do not state something like "Based on the provided medical records in Czech, here is the consolidated report in English".

             At the begining of each report, add the following:
             - Family Anamnesis denoted as 'FA:' stating family health conditions.
             - Objective Diagnosis (OD) where are listed current diseases of that patient.
             - Lab results: list in a structuctured form all lab results and it's values.
             - Current medication of the patient
             - Pre-dispositions to diseases based on the medical history, that has not yet been diagnosed.
             - Medical risk class from 0-3 where 0 is completely healthy and 3 is a severe medical condition.

             Context:
             ```{record}```

             Answer:
          """
  prompt_template = PromptTemplate(template=template, input_variables=["record"])
  # Increase the `temperature` to allow more creative writing freedom.
  llm = ChatOpenAI(openai_api_key='XX-XXXX', temperature=0, model="gpt-4-1106-preview", max_tokens=1000)
  llm_chain = LLMChain(prompt=prompt_template, llm=llm)
  answer = llm_chain.run({"record": record}
  )
  return answer
```

Obrázek 4 – askgpt

Tato funkce pak zavolá na všechny záznamy a vrátí standardizovaný záznam pro každého pacienta, s kterým lze dále pracovat. Vedlejším efektem může být, že záznamy přeloží do jiného jazyka (v tomto případě do angličtiny), v kterém se může dělat např. další výzkum a dotazování nad záznamy z mezinárodních zdrojů.

4 Klasifikace zdravotních diagnóz

Následně bylo provoláno API GPT-4-turbo k extrakci možných diagnóz do pole oddělenými čárkami, kde poslední pozice udávala klasifikace zdravotního stavu dle závažnosti.

Model GPT4-Turbo lze pomocí upraveného promptu instruovat, aby místo diagnózy a slovního popisu vracel číselnou hodnotu 1 pro záznam, který svědčí o diagnóze nebo 0, pokud ne.

```python
async def askGPTAboutDiagnosis(record,diagnosis = "Diabetes Mellitus",dabbr = "DM"):
  template = """
  You are given a medical record about a patient in Czech.
  New lines in the report are typically represented as '\r\n', and then another section begins.
  'Objektivni Anamneza' which is the most relevant to patient's current health is typically denoted as 'OA:'.
  Your goal is to classify whether the patient has {diagnosis} ({dabbr})
  Your answer should be only numeric: '0' for doesn't suffer from {dabbr} and '1' for does have.
  Context:
  ```{record}```
  Answer:
  """
  prompt = PromptTemplate(template=template, input_variables=["record"])
  # Increase the `temperature` to allow more creative writing freedom.
  llm = ChatOpenAI(openai_api_key='XX-XXXX', temperature=0, model="gpt-4-1106-preview", max_tokens=1000)
  llm_chain = LLMChain(prompt=prompt, llm=llm)
  answer = llm_chain.run({"record": record,"diagnosis":diagnosis,"dabbr":dabbr})
  return int(answer)
```

Obrázek 5 – askGTPAboutDiagnosis

DOI: 10.35191/medsoft_2020_1_32_22_26

Lze takto pak projít anamnézy pacientů v cyklu:

```
import asyncio

async def process_patient_anamnesis(patient_anamnesis_list):
    # This will hold the final results
    results = []

    # Iterate over each (patient_id, anamnesis list) tuple in the list
    for patient_id, anamnesis_list in patient_anamnesis_list:
        # Process each anamnesis entry for the current patient
        processed_anamnesis = [await askGPTAboutDiagnosis(anamnesis) for anamnesis in anamnesis_list]

        # Append the processed data to the results list
        results.append((patient_id, processed_anamnesis))

    return results

# Run the processing function and collect results
processed_results = await process_patient_anamnesis(patient_anamnesis_list)
processed_results
```

Obrázek 6 – procesPatientAnamnesis

Výsledek může vypadat například takto, kde první číslo udává id pacienta a druhé 0 nebo 1 udává zda netrpí/trpí diabetes mellitus.

```
[(152, [0]),
 (711, [0]),
 (826, [1]),
 (1385, [1]),
 (1500, [1]),
 (1841, [0]),
 (1878, [1]),
 (2479, [1]),
 (2777, [1]),
 (3372, [1]),
 (4260, [0]),
 (5496, [1]),
 (5599, [1]),
 (5772, [0]),
 (5987, [1]),
 (6933, [0]),
 (7466, [1]),
 (7700, [1]),
 (7835, [1]),
 (8196, [1]),
 (9073, [1]),
 (9285, [0]),
 (9497, [1]),
 (9582, [1])]
```

Obrázek 7 – vysledekKlasifikace

5 Validace

Na základě člověkem označených případů s odborným vzděláním, byla provedena kvantifikace přesnosti velkého jazykového modelu. Validační vzorek obsahoval 40 případů pozitivních na Diabetes Mellitus (DM). Soubor dat zahrnoval různé prezentace a stádia diabetu mellitu, od jasných případů po nuancovanější, kde symptomy nemusí být tak zřejmé.

Při počáteční analýze model AI identifikoval určité případy jako diabetes mellitus, které nebyly odborníky označeny jako takové, což vedlo k předpokládaným falešným pozitivům. Tyto případy byly podrobeny sekundárnímu přezkumu lékařským profesionálem, aby se zjistilo, zda byly diagnózy AI skutečně falešnými pozitivy. Kandidáti na falešnou pozitivitu byly nakonec identifikováni jako správně pozitivní a chyba byla odhalena v klasifikaci člověka s odborným vzděláním. AI identifikoval jako diabetes mellitus i podle vzorů, např. indikace konkrétních léků nebo kontrol, aniž by se v textu explicitně hovořilo o diagnóze ve vztahu k tomuto pacientovi.

Žádné případy falešné negativity nebyly odhaleny.

6 Zápis dat zpět do databáze:

Konsolidované záznamy byly zapsány do nové tabulky, spolu s polem navržených diagnóz a klasifikací zdravotního stavu. Žádný další post-processing nebyl aplikován.

7 Nasazení

Konsolidace a klasifikace byly vystaveny jako separátní služby přes REST API, pomocí frameworku LangServe, který je postavený na FastAPI s využitím asynchronní databáze (Uvicorn).

Technická implementace

Pro robustní systémy, nebo systémy v produkčním provozu je potřeba zvážit technologické aspekty a i jiné požadavky koncových uživatelů a zvolit náležité řešení. Nicméně, pro rychlé prototypování jsme použili kromě již zmíněné knihovny pro jazyk Python ještě tyto technologie:

Integrace RESTful API:

- REST (Representational State Transfer) je už dnes standardem pro komunikaci web service a různých částí systémů. Většinou se používá spolu s protokolem HTTP, takže je dostupný pro webové prohlížeče či webové aplikace, které si dotahují data z REST rozhraní a vizualizují výsledek uživateli srozumitelnou formou.

LangServe [5]:

- LangServe, nástavba pro orchestrátor LangChain, poskytuje vrstvu middleware, která efektivně řídí interakce mezi komplexními schopnostmi zpracování jazyka modelu AI a zjednodušenými koncovými body API. Knihovna umožňuje napsat skripty pro různé jazykové modely a přepínat mezi modely a rozhraními jen pomocí změny parametru.

Framework FastAPI [6]:

- Fast API je knihovna, která v jazyku Python umožňuje poměrně rychle a snadno definovat vlastní rozhraní, které splňuje obecné požadavky na REST a zároveň automatizuje nebo generuje dokumentaci pro rozhraní.

Uvicorn:

- Uvicorn je implementace serverových služeb pro Python, obvykle se používá spolu s FastAPI.

Diskuze

Výsledky prototypu ukázaly potenciál velkých jazykových modelů (LLM) jako je GPT4-turbo v oblasti interpretace lékařských záznamů. Příklad detekce např. diabetu mellitu je indikativní pro širší aplikovatelnost LLM ve zdravotnictví. Model GPT4 je obecný, ale pro medicínské účely byl vyvinutý jiný model v tomto článku nevyzkoušený, tzv. Med-PALM-2 [7], který údajně nabízí téměř dvojnásobnou přesnost odpovědí na medicínské otázky ve srovnání s tradičními metodami prováděnými člověkem.

Integrace AI do diagnostických postupů neznamená nahrazení lidských lékařů, ale naopak spolupráce člověka a AI může zvýšit diagnostický výkon. Začlenění AI do zdravotnictví již dnes probíhá v různých oblastech. Velké jazykové modely mohou pomoci překlenout dosavadní praxi „digitalizace", tj. volně psaných textů lékařských zpráv a přiblížit je k moderním trendům strukturovaného záznamu podle standardů např. HL7/FHIR.

Jak bylo zmíněno a ukázano výše, dobře formulovaný dotaz v přirozeném jazyce může instruovat umělou inteligenci k efektivní pomoci. Tomu se věnuje oblast tzv. „Prompt Engineering", která se zabývá postupy, jak formulovat dotaz velkým jazykovým modelům s přihlédnutím k jejímu fungování [10].

DOI: 10.35191/medsoft_2020_1_32_22_26

Shrnutí

Umělá inteligence přináší revoluci do zpracování jakýchkoliv informací. Jak přemýšlíme o budoucnosti, je jasné, že AI není jen doplňkem zdravotní péče, ale revoluční silou schopnou ji přetvářet. Sam Altman již v roce 2021 predikuje revoluci a optimistickou budoucnost [1].

Ukázali jsme, jak s poměrně nenáročnými technikami používat velké jazykové modely, které stojí i za úspěšnými službami a rozhraními jako ChatGPT. Data a dotazy poslané přes ChatGPT slouží k vylepšování a k učení další verze jazykového modelu. Programátorský přístup přes placené API k velkým modelům zaručuje, že data nejsou dále používána poskytovatelem k učení pro další verzi velkého jazykového modelu, což je jedna z klíčových vlastností, pokud uživatel potřebuje takto zpracovat citlivá data.

Ukázkové skripty a vygenerované lékařské ukázkové záznamy jsou dostupné jako digitální příloha na GITHUBu [8].

Ukázali jsme, že implementace LLM, jako je GPT4-turbo pro analýzu lékařských záznamů a vytáhnutí konkrétní diagnostiky umožní některé typy datové analýzy, které dosud byly pracné nebo prakticky neproveditelné. Ukázali jsme, že AI se osvědčila v převádění nestrukturovaných lékařských zpráv na strukturovaná data, snižuje zátěž dokumentace a otevírá cestu pro lepší správu dat. Z hlediska porozumění a diagnostické přesnosti AI při validaci byla lepší než odborný hodnotitel. Obdobné výsledky mají studie [2] a [3], které použily modely GPT 3.5 a GPT 4 na záznamech pacientů většinou s nádorovým onemocněním plic nebo interpretaci historických záznamů ze sonografických vyšetření.

Závěr

Předvedený jednoduchý prototyp pro porozumění lékařským záznamům z volného textu pomocí umělé inteligence vyvinutý během víkendové aktivity na CEE Hackathonu 2023 v pražském IKEMu demonstruje potenciál pro uplatnění Velkých Jazykových Modelů (LLM) v medicíně jak v českém jazykovém prostředí, tak v jakémkoliv jiném mezinárodním prostředí. Záznamy mohou být napsány v jazyce pacienta nebo lékaře. Konsolidace a klasifikace může být poměrně přesně standardizována v původním jazyce nebo v jazyce vhodném pro další datovou analýzu (např. angličtině). Podrobnější studie by mohla přinést více doporučení do této oblasti.

Oproti analýze založené na pravidlech (např. pomocí regulárních výrazů) velké jazykové modely přinášejí porozumění i z poměrně malého kontextu poskytnutého ve vstupních informacích. Model GPT-4-Turbo dostupný v roce 2023 je obecný, ale výsledky demonstrované výše jsou v souladu se slibnými výsledky obdobných studií.

DETECTION OF MEDICAL DIAGNOSIS, CATEGORIZATION, AND NORMALIZATION OF UNSTRUCTURED MEDICAL RECORDS USING AI

Abstract

The main problem with the digitization of health records is the unstructured nature of many medical records, which are often recorded as free text, complicating their machine processing. During the CEE Hackathon 2023 event at IKEM in Prague, the possibility of converting digital unstructured records into a structured form using artificial intelligence was explored. A prototype was developed that uses AI methods for categorical evaluation of reports and allows natural language querying. The key method for text analysis was the use of pre-trained GPT (Generative Pre-trained Transformer) models from OpenAI, which were instructed to identify specific diagnoses, such as diabetes mellitus. The results showed that artificial intelligence can effectively understand medical texts and has the potential for further improvement in distinguishing between precise diagnoses, such as diabetes and pre-diabetes. Comparison with human evaluators showed that AI can be equally or more accurate in its assessments. Moreover, the programmer's approach and the processing of some extensive databases by artificial intelligence will allow data analysis, which is difficult or impossible using traditional methods such as regular expressions or full-text search.

Keywords

digitization of health records, unstructured data, large language models, standardization of records

Literatura

[1.] Sam Altman: Moore's Law for Everything, http://tcw.org/lefty/Short%20Stories/Moore's%20Law%20for%20Everything.pdf

[2.] Kriti Bhattarai, Inez Y. Oh, Jonathan Moran Sierra, Philip R.O. Payne, Zachary B. Abrams, Albert M. Lai: Leveraging GPT-4 for Identifying Clinical Phenotypes in Electronic Health Records: A Performance Comparison between GPT-4, GPT-3.5-turbo and spaCy's Rule-based & Machine Learning-based methods bioRxiv 2023.09.27.559788; doi: https://doi.org/10.1101/2023.09.27.559788

[3.] Wang Wh, Wang Sy, Huang Jy, et al. An investigation study on the interpretation of ultrasonic medical reports using OpenAI's GPT-3.5--turbo model. J Clin Ultrasound. 2023; 1-7. doi:10.1002/jcu.23590

[4.] The Open AI API documentation, web, accessed 2023: https://platform.openai.com/docs/introduction

[5.] LangServe Library, web, accessed 2023:: https://www.langchain.com/langserve

[6.] FastApi, web framework for building APIs with Python 3.8+, web, accessed 2023: https://fastapi.tiangolo.com/

[7.] Karan Singhal et al., Towards Expert-Level Medical Question Answering with Large Language Models, Google Research, DeepMind, preprint https://arxiv.org/pdf/2305.09617.pdf

[8.] Algolem, ukázky některých zdrojových kódů použitých v článku, web, accessed 2023: https://github.com/TomasKulhanek/aigolem

[9.] Jupyter, interactive computing across all programming languages, web, accessed 2023: https://jupyter.org/

[10.] White, J., et al. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. ArXiv, abs/2302.11382. https://doi.org/10.48550/arXiv.2302.11382

Kontakt

Stanislav Jirák

Tomáš Kulhánek