Táblázatos kérdés-megválaszolás magyar nyelven

Tóth Gábor, Programtervező informatikus szak Msc, II. évf

Dr. Farkas Richárd egyetemi docens, Szegedi Tudományegyetem

1. Bevezetés

A táblázatok széles körben elterjedtek, és gazdag információforrást jelentenek az interneten és különféle dokumentumokban. Statisztikai adatok szerint az internetes weboldalakon található táblázatok száma elérte a több százmilliót (Lehmberg és mtsai, 2016); a vállalati környezetben pedig az Excel-szerű fájlokban lévő táblázatok száma meghaladta a 115 milliót (Wang és mtsai, 2020). A táblázatokból származó releváns információk pontos keresése kulcsfontosságú számos valós alkalmazásban, például pénzügyi elemzésekben vagy a tudományos kutatásokban. Az elmúlt két évben nagy nyelvi modellek (Large Language Models, LLM-ek) figyelemreméltó fejlődése (Brown és mtsai, 2020; Chowdhery és mtsai, 2022; OpenAI, 2023a; Touvron és mtsai, 2023; Google, 2023) átalakította az információkeresés megközelítését.

Az LLM-ek legfőbb előnye, hogy képesek általánosítani és összetett nyelvi struktúrákat értelmezni, ami kiemelkedően fontossá teszi őket a nyelvi adatok feldolgozásában. Bár ezek a modellek elsősorban az angol nyelvre optimalizáltak, egyre nagyobb igény mutatkozik arra, hogy más nyelveken, például magyarul is sikeresen alkalmazhatók legyenek.

Az egyik különösen érdekes alkalmazási terület a táblázatos kérdés-megválaszolási (TQA) feladat, amely során a modellnek egy táblázat alapján kell a feltett kérdésekre választ adnia. A TQA-feladatok jelentősége az adatbázisokban és különböző dokumentumokban elérhető strukturált információ hasznosításában rejlik, amelyeket hatékony kérdés-válaszoló rendszerek segítségével lehet automatizálni. Ez a terület a Előtanított Nyelvi Modellek (PLM-ek) térnyerésével gyors fejlődésen ment keresztül, azonban továbbra is komoly kihívást jelent a jelenlegi modellek számára (Jin és mtsai, 2022). Magyar nyelven az LLMek nem teljesítenek annyira jól, mint angolul, és jelenleg nem áll rendelkezésre olyan TQA adatbázis, amely magyar nyelven készült kérdés-megválaszolási rendszerek vizsgálatát tenné lehetővé.

Jelen dolgozat célja egy magyar nyelvű TQA-adathalmaz javarészt automatikus létrehozása, amely magyar Wikipédia szócikkek táblázatos adatait és a hozzájuk kapcsolható szöveges tartalmakat hasznosítja. Az ezekből az adatokból készített TQA-adathalmaz lehetőséget nyújt az LLM-ek tesztelésére magyar nyelvű adaton, valamint a TQA-algoritmusok hatékonyságának vizsgálatára. A

dolgozat további célja, hogy különböző nagy nyelvi modellek teljesítményét és megbízhatóságát mérje fel a magyar nyelvű TQA-adathalmazon, összehasonlítva a különböző algoritmusokat és többféle LLM-et. A kutatás hozzájárul a magyar nyelvű nyelvtechnológiai kutatásokhoz, mivel ez az adatbázis elősegíti a magyar nyelvű TQA-rendszerek fejlesztését és kiértékelését, amelyek hosszú távon alapvető fontosságúak lehetnek a magyar nyelvű digitális szolgáltatások és alkalmazások számára.

Összefoglalva, ez a dolgozat a egy új típusú magyar nyelvű adathalmazt mutat be, ami egy új feladatként lehetőséget biztosít arra, hogy a magyar nyelven is teszteljük a legkorszerűbb LLM-eket, amit dolgozatomban tárgyalok is.

2. Háttér

2.1 Nagy nyelvi modellek

Az elmúlt években a Nagy Nyelvi Modellek robbanásszerű fejlődése révén lenyűgöző teljesítményt értek el különféle nyelvtechnológiai (Natural Language Processing, NLP) feladatokban (Brown és mtsai, 2020; Touvron és mtsai, 2023; Team és mtsai, 2023). A kutatások eddig számos aspektusból és képesség szempontjából vizsgálták az LLM-ek teljesítményét (Bang és mtsai, 2023b; Bubeck és mtsai, 2023; Akter és mtsai, 2023), de hatékonyságuk a strukturált adatok, például táblázatok kezelésében egy kevésbé kutatott terület.

A strukturálatlan szövegekkel ellentétben a táblázatok rendszerezett struktúrában tárolják az információt. Ez a tulajdonság teszi a táblázatos adatokat számos alkalmazás alapjává, beleértve az orvosi diagnosztikát, a virtuális személyi asszisztenseket, valamint az ügyfélkapcsolat-kezelést (Hemphill és mtsai, 1990; Dahl és mtsai, 1994; Akhtar és mtsai, 2022; Xie és mtsai, 2022), és így tovább.

A nagy nyelvi modellek a mesterséges intelligencia és a természetes nyelvfeldolgozás területén kifejlesztett komplex rendszerek, amelyek képesek nagy mennyiségű szöveg adatainak értelmezésére, elemzésére és feldolgozására. Az LLM-ek alapját a neurális hálózati modellek képezik, amelyek mérete és bonyolultsága jelentősen meghaladja a korábbi NLP modellekét. A legmodernebb LLM-ek, mint például a GPT modellek, a BERT, stb. több milliárd paramétert tartalmaznak, és ennek köszönhetően széles körű feladatokat képesek ellátni, a szövegértelmezéstől kezdve a kérdés megválaszoláson át a szöveggenerálásig.

Az LLM-ek jellemzően a transzformer architektúrára épülnek, ami rövid időn belül meghatározóvá vált az NLP területén. A transzformerek lehetővé teszik, hogy a modell az input szöveg különböző

részei között hosszú távú kapcsolatokon át is felismerje azokat az összefüggéseket, amelyek alapvetőek a nyelvi megértéshez.

Az LLM-ek előtanításához általában hatalmas mennyiségű adatot használnak, amelyek tartalmazzák az interneten fellelhető különféle nyilvános információkat, könyveket, cikkeket, blogokat és más forrásokat. Az előtanítás során a modelleket arra tanítják, hogy szavak, kifejezések és szövegblokkok között mintázatokat és kapcsolódásokat fedezzenek fel, ami képessé teszi őket a nyelv természetes feldolgozására. Az előtanítás után jellemzően finomhangolási lépéseken mennek keresztül, amelyek során az adott feladatra optimalizálják őket, például fordításra, vagy ebben az esetben kérdés megválaszolásra.

A GPT típusú LLMek képesek szöveget generálni, fordítani, kérdésekre válaszolni, és segíteni különféle kreatív írási feladatokban, például történetmesélésben és összegzésben. Az OpenAI által fejlesztett GPT modellsorozat például jól ismert a szöveggenerálási képességeiről. A GPT-4 modelljük csaknem 2 ezer milliárd paramétert tartalmaz, és az egyik legnagyobb nyelvi modell, amely eddig készült.

A Google által fejlesztett BERT modellek inkább arra specializálódtak, hogy szövegértelmezési feladatokat végezzenek el, mint például a szöveg-kategorizálás és a kérdés-megválaszolás. A BERT kétirányú tanulási mechanizmusa lehetővé teszi, hogy a modell mind a bal, mind a jobb kontextust figyelembe vegye az egyes szavak jelentésének értelmezésekor, ami különösen hatékonnyá teszi az NLP-feladatok során.

Annak ellenére, hogy az LLM-ek sikeresen alkalmazhatók az angol nyelvű feladatokban, más nyelvek esetében korlátozottabb eredményeket nyújtanak. Ennek fő oka, hogy az LLM-ek előtanításhoz szükséges hatalmas mennyiségű adathoz nem minden nyelv esetében férnek hozzá, így például a magyar nyelvre optimalizált modellek kifejlesztése továbbra is kihívást jelent. Az LLM-ek előtanítása gyakran az interneten elérhető, angol nyelvű szövegeken történik, és ezáltal egyes nyelvek, különösen a kisebb nyelvek, mint a magyar, alulreprezentáltak maradnak. Az olyan kutatások, amelyek magyar nyelvű kérdés-válaszoló adatbázisokat hoznak létre, alapvetően fontosak a magyar nyelvű LLM-ek fejlesztésében és azok teljesítményének javításában.

Megjegyezzük továbbá, hogy a közelmúltban kimutatták, hogy a nagy nyelvi modellek (LLM-ek) gyakran „szennyezettek” különböző benchmark adatbázisok által (Golchin és Surdeanu, 2023), ami befolyásolhatja az egyes kiértékelések eredményeit.

2.2 TQA feladat

A táblázatos kérdés-megválaszolás (Table Question Answering, TQA) feladat a nyelvtechnológia egyik speciális alkalmazási területe, ahol a modell célja, hogy legalább részben strukturált adatokból (táblázatokból) válaszoljon a feltett kérdésekre. A TQA különösen hasznos, amikor nagy mennyiségű adatot szeretnénk értelmezni és kinyerni jól meghatározott kérdések megválaszolásával, például pénzügyi táblázatok, sportstatisztikák vagy adatbázisok esetében.

Az encoder-decoder típusú modellek, mint például a TAPEX (Liu és mtsai, 2022), és a mélytanulás-alapú modellek, például az MLM-módszerek (Herzig és mtsai., 2020) tanítása révén a modellek képesek megérteni a strukturált táblázatos adatok részleteit, és azokon alapuló válaszokat adni. Ezeknek a módszereknek azonban jelentős hátrányuk, hogy a teljes adatállományt memóriába kell betölteni, ami miatt nem alkalmasak hatalmas, összetett valós adathalmazok kezelésére. Továbbá a kontextus hossz korlátozása – akárcsak más nagy nyelvi modelleknél – megszabja, hogy mennyi adat adható meg egyszerre a modell számára, ami ezeknél a módszereknél különösen korlátozott.

A TQA-feladat során a modell egy vagy több táblázatot kap bemenetként, valamint egy természetes nyelven megfogalmazott kérdést, amelyre a táblázatban található információk alapján kell válaszolnia. A feladat sikeres végrehajtásához a modellnek képesnek kell lennie arra, hogy a kérdés alapján valamilyen módon a táblázat megfelelő celláiban található adatokat azonosítsa és felhasználja. Ez magában foglalhat egyszerű adatlekérdezéseket (pl. „Mennyi az X termék ára?”), összetett műveleteket (pl. „Melyik évben volt a legnagyobb növekedés az Y termék értékesítésében?”), valamint több lépésben történő információ kombinációkat is.

A TQA-feladat számos technikai kihívással jár:

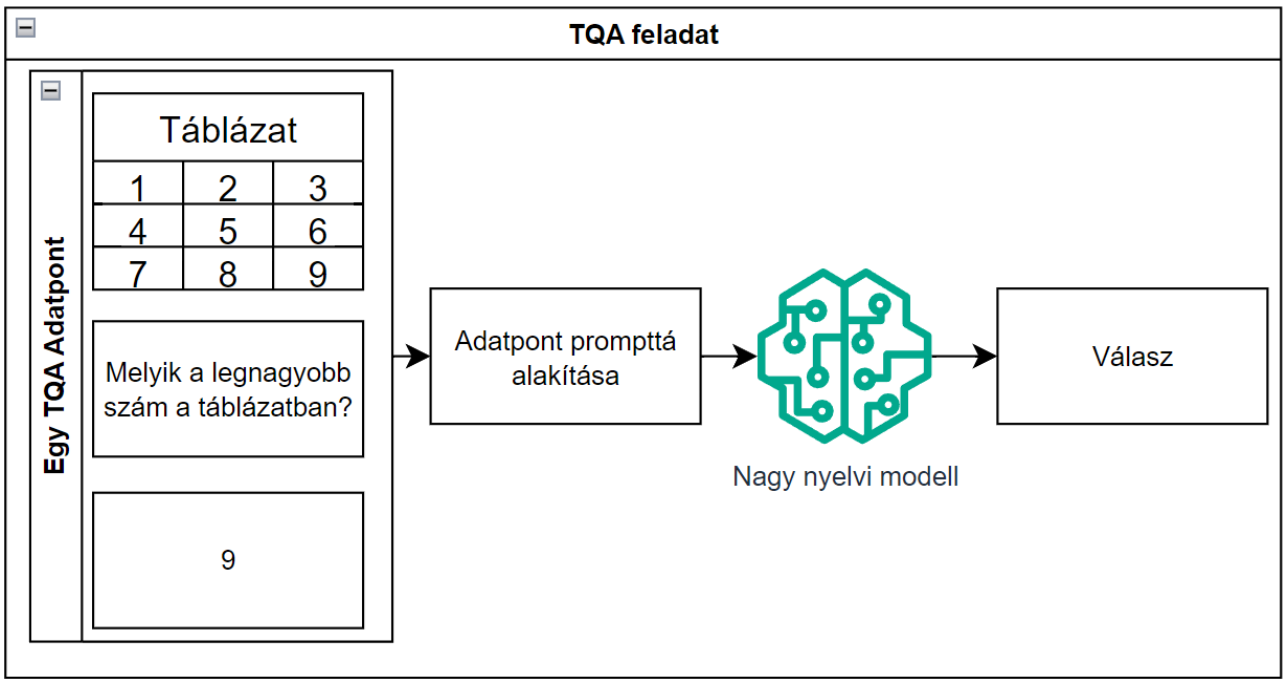
1. **Természetes nyelvi megértés és táblázatértés integrációja**: A modellnek képesnek kell lennie arra, hogy a kérdésben szereplő nyelvi utasításokat összekapcsolja a táblázat strukturált elemeivel (sorok, oszlopok, cellák).
2. **Szintaktikai és szemantikai összefüggések felismerése**: Gyakran előfordul, hogy a táblázatban található adatot különböző logikai kapcsolatokon vagy szemantikai összefüggéseken keresztül kell értelmezni. Például a „legnagyobb növekedés” vagy „legkisebb érték” kérdések megválaszolásához a modellnek összehasonlító műveleteket kell végrehajtania.
3. **Számítási és logikai műveletek végrehajtása**: A kérdések egy része számításokat igényel, mint például összegzést, átlagolást vagy százalékszámítást. Ehhez a modellnek egyaránt szüksége van matematikai és logikai képességekre.

4. **Adaptálás több nyelvhez**: Mivel a legtöbb TQA-modellt elsősorban angol nyelven fejlesztik és tanítják, más nyelveken (például a magyar nyelven) való alkalmazásuk jelentős kihívást jelent, különösen a helyi nyelvi és táblázati sajátosságok figyelembevétele szempontjából.

LLMekkel általában zero-shot QA-t javasolták a kérdés-megválaszolásraó modellek tanítására emberek által kézzel címkézett tanítóhalmaz helyett. Az első felügyelet nélküli QA modellt az LLMek megjelenése előttkorábban javasolták (Lewis és mtsai, 2019), amely kérdés-válasz-kontextus hármasokat generál a QA modell tanítása érdekében, felügyelet nélküli gépi fordítással. A generált kérdések azonban eltérnek az ember által írt kérdésektől, és általában sok lexikai átfedést mutatnak a kontextussal. Ennek megoldására a következő munkák a Wikipédia szócikkeitidézett dokumentumait (Li és mtsai, 2020), előre definiált sablonokat (Fabbri és mtsai, 2020) vagy előzetesen betanított nyelvi modellt (Puri és mtsai, 2020) használtak a kézzel annotált kérdésekre hasonlító, természetesebb kérdések előállításához.

A TQA-feladatokra használt LLM gyakran finomhangolásra kerülnek speciális TQA adathalmazban, amelyek tartalmazzák a táblázatokat, a kapcsolódó kérdéseket és a helyes válaszokat.

A TQA-feladatok jövőbeli fejlesztési irányai közé tartozik a többnyelvű és nyelvfüggetlen TQA-rendszerek kidolgozása, amelyek lehetővé teszik a modellek számára, hogy különböző nyelveken is jól teljesítsenek, például a magyar Wikipédián található táblázatokra is alkalmazhatók legyenek. Emellett növekvő hangsúlyt kap az a kérdés, hogy a modellek hogyan tudják elkerülni a hibás válaszokat, különösen bonyolult kérdések esetén, amelyek több táblázat vagy összetett műveletek bevonását igénylik.



I. ábra *Táblázatos kérdés-megválaszolási feladat*

Az LLMek előtti táblázatos kérdés-megválaszolási modellek az alapján különböztethetők meg, hogyan generálják a válaszokat:

1. **Lekérdezések formális átalakítása (szemantikus elemzés)**,
2. **Közvetlen válaszok generálása**.
3. **Hibrid megközelítések**, amelyek mindkettőt ötvözik.

Habár az SQL hatékony a táblázatokon végzett kérdés-megválaszolásban (Shi és mtsai, 2020), jelentős hátránya, hogy nem jól alkalmazható nem adatbázis-alapú táblázatok esetén, és problémás lehet a kérdések átfordítása SQL-re.

Közvetlen válasz generálás: A közvetlen válasz generálás közvetlenül állítja elő a végső válaszokat, kihagyva a kérdések valamilyen lekérdező nyelvre alakításának lépését.

- Példa: (Mueller és mtsai, 2019) gráf-alapú neurális hálót (GNN) használ az adatszerkezet kódolására, és egy dekódolót, amely a gráf és a lekérdezés alapján generál válaszokat.
- Támogatja az adatbővítési technikákat, például SQL eredmények vagy Excel-formulák átalakítását azok végrehajtására (TAPEX: Liu és mtsai, 2021; FORTAP: Cheng és mtsai, 2021).

Ennek ellenére a transzformerekkel végzett numerikus érvelések hatékony kezelése továbbra is kihívás (Zhou és mtsai, 2022a).

Hibrid módszerek: A hibrid megközelítések a táblázatokból releváns tokeneket emelnek ki, hogy azokat egy aggregátorral feldolgozzák, és speciális végrehajtóra irányítsák.

- **TAGOP** (Zhu és mtsai, 2021): Szekvencia címkézéssel emeli ki a releváns cellákat, majd egy osztályozót használ a szimbolikus érvelési programok összerakására.
- **TAPAS** (Herzig és mtsai, 2020): Egy BERT-szerű kódoló végén osztályozó réteget használ a tartalom kiválasztására és aggregációs műveletek végrehajtására.

Ezek a módszerek erős numerikus képességekkel rendelkeznek, de korlátozott kifejező készségük miatt nehezen boldogulnak összetett, több aggregációt igénylő lekérdezésekkel (Herzig és mtsai, 2020). Az általunk javasolt köztes logikai forma-alapú felügyelet megoldja ezt a problémát azáltal, hogy lehetővé teszi az összetett több aggregációs reprezentációk kezelését.

A TQA-feladatokban gyakran szerepelnek numerikus válaszok, amelyek pontosságának ellenőrzése speciális módszereket igényel:

- **Abszolút hibaszámítás**: Az eltérés mérése az elvárt értékhez képest, különösen akkor hasznos, ha a válasz konkrét számadat.
- **Relatív hibaszámítás**: A hibaarány figyelembevétele százalékban, ami főleg nagy értékek esetében lényeges.
- **Toleranciaküszöb**: Bizonyos esetekben meghatározható egy tűréshatár, amely alapján a válasz helyesnek tekinthető, még akkor is, ha nem pontosan egyezik az elvárt értékkel.

2.3 Adatbázisok

Számos TQA adatbázis létezik angol nyelvre, amelyek különféle típusú kérdés-megválaszolási feladatokat támogatnak és különféle táblázatos struktúrákat tartalmaznak. Ezek az adathalmazok a TQA modellek fejlesztésének és tesztelésének alapját képezik.

1. **WTQ (WikiTableQuestions)** (Pasupat és Liang, 2015b) a jelenlegi modellek korlátainak vizsgálatára szolgáló alapvető adatforrásként szolgál. Ez egy táblázatos kérdés-válasz adathalmaz, amely 2108 HTML-táblából és tömegesen gyűjtött kérdés-válasz párokból áll. Annak ellenére, hogy mind a tanító-, mind a tesztkészletben több kérdés vonatkozik egy-egy táblázatra, a két készlet táblázatai nem állnak átfedésben egymással. A WikiTableQuestions több olyan kulcsfontosságú jellemzővel rendelkezik, amelyeknek köszönhetően hatékony és kihívást jelentő benchmark:
    - A kérdések megválaszolásához gyakran több logikai lépést igényel, jellemzően különböző információ összegyűjtésével egyetlen táblázatból.
    - A táblázatok nem adatbázisszerűen strukturáltak, illetve gyakran tartalmaznak nem egységes cella értékeket, amelyek értelmezése az olvasó implicit képességeire támaszkodik.
    - A cellák gyakran formális reprezentációkat és természetes nyelvi elemeket vegyítenek, ami a tisztán programozási megközelítések alkalmazását nehézkessé teszi.

School	Conference	Record (conference)	Head coach	CWS appearances	CWS best finish	CWS record
Arizona	N/A	36–8 (N/A)	Frank Sancet	4 (last: 1958)	2nd (1956)	7–8
Clemson	ACC	23 –6 (11–5, 0 GB)	Bill Wilhelm	1 (last: 1958)	5th (1958)	1–2
Colorado State	N/A	24–5 (N/A)	Pete Butler	5 (last: 1958)	5th (1955)	2–10
Connecticut		20–1 (N/A)	J. Orlean Christian	1 (last: 1957)	5th (1957)	1–2
Fresno State		38–11 (N/A)	Pete Beiden	0 (last: none)	none	0–0
Oklahoma State	Big 8	22–4 (17–3, 0 GB)	Toby Greene	2 (last: 1955)	3rd (1955)	5–4
Penn State	N/A	15–4 (N/A)	Joe Bedenk	2 (last: 1957)	2nd (1957)	5–4
Western Michigan	MAC	24–7 (8–2, 0 GB)	Charlie Maher	3 (last: 1958)	2nd (1955)	7–6

URL http://en.wikipedia.org/wiki?action=render&curid=10424731
Title 1959 NCAA University Division Baseball Tournament
Table # 0

nt-1314
list each of the schools that came in 2nd for cws best finish.
Arizona | Penn State | Western Michigan

nt-1977
does clemson or western michigan have more cws appearances?
Western Michigan

nt-2620
list the schools that came in last place in the cws best finish.
Clemson | Colorado State | Connecticut

nt-2635

II. ábra *WikiTableQuestions adattábla minta*

2. **SQA (Sequential Question Answering)**: Ez az adatbázis úgy lett kialakítva, hogy egymást követő kérdések sorozatát tartalmazza ugyanazon táblázatokhoz, ahol a kérdések gyakran egymásra épülnek. Az SQA ezért különösen alkalmas olyan TQA modellek fejlesztésére, amelyek több lépésben megfogalmazott kérdéseket is képesek kezelni, mivel gyakran szükséges a korábbi kérdések válaszainak figyelembevétele az újabb kérdések megválaszolásakor.
3. **WikiSQL** (Zhong és mtsai, 2017), hasonlóan a WikiTableQuestions-hoz, 80654 kérdés-válasz párt tartalmaz 24241 Wikipedia-táblázat felett. Bár eredetileg szemantikus elemzésre készült, gyenge felügyeleti környezetekben is alkalmazzák, kizárólag a kívánt válasz intervallumot használva jelként. Az adatállomány összes táblázata teljesen elemezhető típusokkal. A kérdések egyszerűbbek, mint a WikiTableQuestions esetében, és csak teljes cella értékére vonatkozó műveleteket tartalmaznak, amelyek egy SQL-lekérdezéssel teljes mértékben elemezhetők.
4. **TabFact**: A TabFact (Chen és mtsai, 2019) adatbázis a táblázatos tényellenőrzésre fókuszál. Az adatbázis olyan természetes nyelvi állításokat tartalmaz, amelyek alapján a modelleknek el kell dönteniük, hogy a megadott táblázat adatai alapján igazak vagy hamisak-e. A TabFact ezért inkább tényellenőrzési feladatokra használatos, de TQA-feladatokban is alkalmazható, amikor az állítások és kérdések értelmezése táblázatos kontextusban történik.
5. **HybridQA**: A HybridQA (Chen és mtsai, 2020) adatbázis kombinált kérdés-válasz feladatokat tartalmaz, amelyek táblázatos és szöveges forrásokat is igényelnek a kérdések megválaszolásához. Itt a válaszok egyaránt követelhetik meg táblázatok és szöveges bekezdések használatát, így az adatbázis különösen alkalmas komplex TQA-feladatok gyakorlására, ahol a modelleknek képesnek kell lenniük többféle információforrást integrálni.
6. **TAT-QA (Table-and-Text Question Answering)**: A TAT-QA (Zhu és mtsai, 2021) adatbázis olyan kérdéseket tartalmaz, amelyek szintén táblázatos és szöveges forrásokra

egyaránt építenek. Különösen pénzügyi és számítási feladatokat tartalmaz, ahol a modellek összevonva oldják meg a szöveges leírásokat és a táblázatok adatait a helyes válasz előállításához. A TAT-QA az üzleti és pénzügyi szektorhoz kapcsolódó TQA-feladatokban használatos, ahol az egyes kérdések több lépésben történő számításokat igényelhetnek.

Ezek az adatbázisok különféle típusú TQA modellek fejlesztését teszik lehetővé, így mind a szöveg-alapú, mind a táblázat-orientált információfeldolgozásban hasznosak, hozzájárulva a TQA-technológia fejlődéséhez.agyar TQA adatbázisok

3. Magyar TQA adatbázis előállítása

A manuális adat annotáció nagyon drága. Ahhoz, hogy ezt a költséget csökkentsük, egy nagyban automatikus módszert dolgoztunk ki magyar nyelvű TQA kiértékelő adatbázis előállítására. A magyar nyelvű TQA (Táblázatos Kérdés-Válasz) adatbázis létrehozása a magyar Wikipédián található táblázatok és kapcsolódó szövegek automatizált gyűjtésével és feldolgozásával történt. A cél az volt, hogy olyan kérdés-válasz párokat hozzunk létre, amelyek lehetővé teszik a nagyméretű nyelvi modellek tesztelését és finomhangolását a magyar nyelvű táblázatos kérdés-válasz feladatokhoz. Az alábbiakban részletesen bemutatjuk az adatbázis összeállításának folyamatát, a Wikipédiából való adatgyűjtést, az adatok tisztítását és az adathalmaz struktúráját.

A magyar Wikipédiában található cikkek jelentős része tartalmaz olyan táblázatokat, amelyek különféle adatokat és kategóriákat rendeznek strukturált formában. Az adatgyűjtési folyamat során célzottan a következő típusú információkat gyűjtöttük:

- **Táblázatos adatok**: A cikkekben található táblázatok, amelyek számos adattípust, például statisztikai adatokat, időpontokat, országokat, földrajzi helyeket, vagy történelmi eseményekre vonatkozó információkat tartalmaznak.
- **Kapcsolódó szöveges kontextus**: A táblázatok körül található szöveges információkat is feldolgoztuk, mivel ezek gyakran magyarázzák vagy részletezik a táblázatok tartalmát. Ez a kontextus hasznos a kérdések megalkotásához, mivel a szöveg utalhat a táblázat egyes celláira, és további jelentést vagy magyarázatot ad az adatokhoz.

Az adatgyűjtés a következő lépésekből állt:

- **Web Szkréperek és API-k használata**: A magyar Wikipédiából történő adatkinyeréshez Python-alapú webszkrépereket és a MediaWiki API-t használtuk, amely lehetővé tette a cikkek szövegének, a táblázatok HTML-kódjának és a formázott adatainak automatikus letöltését.

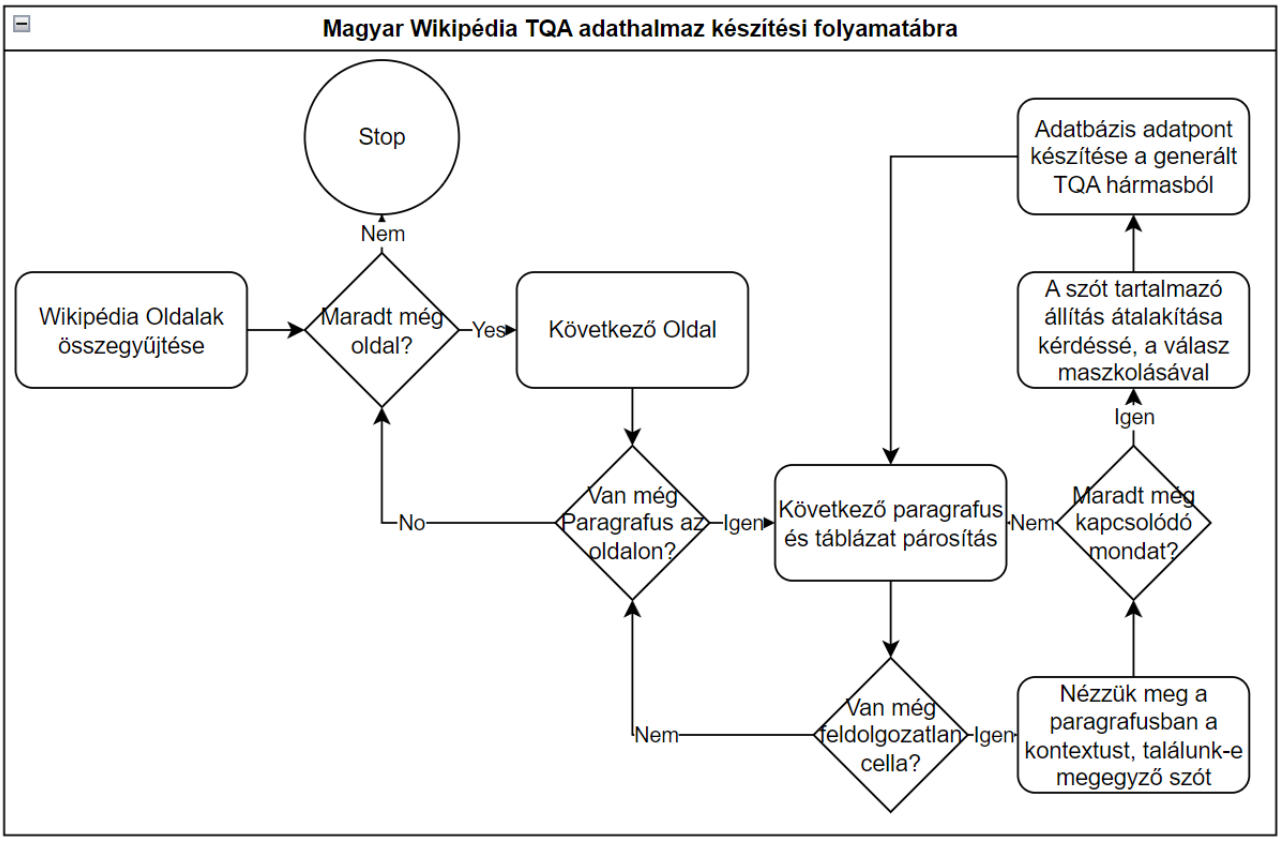
- **Adatstruktúra kivonása**: Az összegyűjtött táblázatok HTML-struktúrájának feldolgozása során az adatokat egy strukturált formába alakítottuk át, amelyben minden táblázat sor és oszlop különálló entitásként kezelhető. Így könnyebben társíthatóak a táblázat celláihoz kapcsolódó kérdések.

Az összegyűjtött adatok további feldolgozást igényeltek annak érdekében, hogy pontos, egységes és használható adathalmazt hozzunk létre. A tisztítás során a következő feladatokat végeztük el:

- **Duplikációk eltávolítása**: Mivel a Wikipédiában előfordulhat, hogy egyes táblázatok ismétlődnek vagy hasonló adatokat tartalmaznak, ezért eltávolítottuk a redundáns bejegyzéseket, hogy az adathalmaz minél változatosabb és informatívabb legyen.
- **Hiányzó adatok kezelése**: A táblázatok egyes cellái esetenként hiányosak vagy nem tartalmaznak releváns információt. Ezeket a cellákat vagy kitöltöttük, ha a kontextus alapján lehetett következtetni a helyes adatértékre, vagy jelöltük, hogy ezek az adatok nem használhatóak a kérdés-válasz feladatban.
- **Adatok típus szerinti rendezése**: Az adatokat különböző típusok szerint csoportosítottuk (pl. dátum, szám, szöveg), hogy a későbbi feldolgozás során könnyebben használhatók legyenek. Például az időpontokat és számadatokat formailag és tartalmilag is egységesítettük.

A kérdés-válasz párok létrehozása során célunk volt, hogy a táblázatokhoz kapcsolódó minél több típusú kérdést lefedjük. A kérdés-válasz párok generálásánál figyelembe vettük, hogy a kérdések:

- **Egyszerű információt kérjenek**: Például egy adott cellában szereplő konkrét adat megkérdezése, mint „Mi Magyarország fővárosa?” egy országokra vonatkozó táblázatban.
- **Összetett lekérdezéseket igényeljenek**: Olyan kérdések, amelyek több cella vagy sor összehasonlítását, összegzését vagy kiszámítását követelik meg, például „Melyik ország népessége a legnagyobb?”.
- **Időbeli és logikai kapcsolatokat kérdezzenek**: Például „Hány év telt el Magyarország uniós csatlakozása óta?” vagy „Melyik esemény történt előbb?” Az ilyen kérdések célja a modell logikai következtetési képességének vizsgálata.



III. ábra *az egyes LLMek pontossága a magyar TQA feladaton*

A megvalósítás egy Python-alapú script, amely automatizáltan gyűjt és dolgoz fel adatokat a magyar Wikipédiáról. A rendszer a következő fő lépéseket hajtja végre:

1. **Wikipédia szakaszok elemzése és feldolgozása:** Az eszköz letölti a megadott kezdőoldal tartalmát, majd külön fejezetekre bontja HTML címkék alapján. Ezeket a létrehozott szakaszokat objektumokká alakítja, amelyek tartalmazzák a szöveges bekezdéseket és a táblázatokat.
2. **Adatok strukturált feldolgozása**: A szövegből eltávolítja a HTML címkéket és hivatkozásokat, valamint a magyar számformátumot is felismeri és kezeli (pl. vesszővel elválasztott tizedesek). A táblázatokat Pandas DataFrame formátumba konvertálja, megőrizve azok eredeti numerikus és szöveges tartalmát.
3. **Mondatok és táblázatok összevetése**: A rendszer azonosítja azokat a mondatokat, amelyek tartalmazzák a táblázatban szereplő adatokat, és ezek alapján állít elő kérdéseket.
4. **Kérdés-generálás:** Az eszköz támogatja az XLM-RoBERTa modell vagy LLM-alapú kérdésgenerálást. Az algoritmus célja, hogy egy mondat és abban előfordulóérték (ami egyébként egy táblázat cellájának értéke) alapján olyan kérdést generáljon, ami az értékre kérdez rá. Azaz az állító mondatot kérdéssé alakítjuk át.

5. **Adatbázis-kezelés**: A kinyert adatokat és generált kérdéseket SQLite adatbázisban tárolja. Ez lehetővé teszi a későbbi elemzést, illetve riportok készítését.
6. **Riportkészítés**: Egy dedikált modul riportot generál az adatbázisból, amely összefoglalja az összegyűjtött adatokat és a generált kérdéseket. A riport HTML formátumú, így a kérdések, a táblázat és a válasz is könnyen megtekinthető külső szoftver segítsége nélkül is.

Az így összeállított adathalmaz teljesen strukturált, így a TQA-feladatokhoz könnyen felhasználható, a riport pedig az adathalmazt könnyen olvashatóvá teszi. Az adathalmaz a következő elemeket tartalmazza:

- **Táblázat**: A forrástáblázat maga, amely a Wikipédiáról származik, és strukturált formában van tárolva.
- **Kérdések**: Minden táblázathoz kapcsolódóan több kérdés-válasz pár található. A kérdések különböző nehézségűek és típusúak, hogy különféle modellek képességeit mérhessük.
- **Válaszok**: Minden kérdéshez tartozik egy referencia-válasz, amely az elvárt kimenetként szolgál a kiértékelés során.

Például: adott egy táblázat, amely Szeged népességének alakulását mutatja be. A fejezetben található szövegrészlet átalakításával a kapott kérdés: Melyik évben éltek a legtöbben a városban? A válasz pedig az, hogy 1990-ben.

Az adatgyűjtés során célunk az volt, hogy csak a megbízható táblázatok és állításokat elemjük ki, hogy minél kevesebb manuális ellenőrzésre legyen szükség. Azaz egyértelműen nem a mennyiség hanem a minőség volt a célunkA vizsgált 2000 szócikkből összesen 1086 olyan szövegrészletet vagy mondatot azonosítottunk, amelyben szerepeltek a táblázat közvetlen közelében, táblázatok celláiban megtalálható kifejezések. Azonban ezek többsége nem állt releváns kapcsolatban a megfelelő táblázattal. Ennek következtében szigorítottunk a keresési feltételeken: az elemzést kizárólag a táblázatok közvetlen környezetére (pl. az adott alfejezeten belüli tartalomra) szűkítettük, és több szűrési technikát is alkalmaztunk.

Ezek a szűrési eljárások magukban foglalták például:

- annak vizsgálatát, hogy a táblázat cellájában található célérték többször előfordul-e a mondatban,
- a túl hosszú mondatok kizárását, amelyek kevésbé informatívak, vagy nem egyértelműek.

Ezek az eljárások a vizsgált szócikkekben helyes adatpontokat megtartották. Az így szűkített adathalmaz végül néhány száz potenciális adatpontot tartalmazott, amelyeket manuálisan validáltunk. Ez a megközelítés ugyan jelentős előrelépést jelent, de további fejlesztésekkel és optimalizációval várhatóan még hatékonyabbá tehető az adatgyűjtési folyamat. Problémát jelent még, hogy a táblázatok nem jól strukturáltak, ezért ezek a HTML-ből nyert adatok hiányos vagy elcsúszott cellákat, néha üres oszlopneveket tartalmaznak, amivel nehezebbé teszik a modellek számára a táblázat értelmezését. Fontos megjegyezni, hogy míg a WTQ adatbázis legalább 8 sorral és 4 oszloppal rendelkező táblázatokat tartalmaz, addig itt ilyen szűrőt nem alkalmaztunk, előfordulhatnak apró táblázatok is. Ez azt jelenti, hogy megfelelően felismert táblázat esetén a feladat lényegesen könnyebb lehet más adatbázisokhoz képest.

Az általunk fejlesztett algoritmus segítségével 2000 magyar Wikipédia-szócikk feldolgozásából mindössze 54 érvényes adatpontot sikerült előállítani, ami 32 egyedi táblázatot használ fel. Ez azt jelenti, hogy táblázatonként átlagosan 1-2 releváns mondatot találtunk, amely kifejezetten kapcsolódott az adott táblázat tartalmához, és a cellákban szereplő értékekre vagy azok magyarázatára utalt. Ez azt jelenti, hogy a táblázatok közvetlen környezetében található szöveges információk túlnyomó többsége nem felelt meg az előzetesen meghatározott relevancia-kritériumoknak, így ezek további szűrést vagy pontosítást igényeltek. Habár átlagosan csupán egyetlen adatpont jutott 37 szócikkre, figyelembe kell venni, hogy a magyar Wikipédia több mint 1 millió lapot tartalmaz. Emiatt feltételezhető, hogy a módszer finomhangolásával akár tízezres nagyságrendű adathalmaz is kinyerhető.

Az elkészült magyar TQA-adatbázis felhasználható a magyar nyelvű nagy nyelvi modellek (LLM-ek) képességeinek értékelésére, különös tekintettel a táblázatos kérdés-válasz feladatokra. Az adatbázis alkalmas különféle modellek tesztelésére és finomhangolására, és hozzájárulhat a magyar nyelvtechnológiafejlődéséhez. A következő fejezetekben bemutatom számos megközelítés kísérleti eredményeit ezen az a kiértékelő adatbázison.

4. Kiértékelt módszerek

A kutatásban felhasznált LLM-ek listája a következő modelleket tartalmazza, amelyek a magyar nyelvű táblázatos kérdés-válasz feladatokhoz nyújtanak alapot. Ezek a modellek különböző architektúrákkal és előzetes tréningfolyamatokkal rendelkeznek, amelyek révén hatékonyan képesek a természetes nyelv feldolgozására és a táblázatokban rejlő információk kinyerésére.

1. **TAPAS (Table Parsing using Pre-trained Sequence-to-Sequence Models)**: A TAPAS (Herzig és mtsai, 2020) egy BERT-alapú modell, amelyet kifejezetten táblázatos kérdés-

válasz feladatokra fejlesztettek ki. A modell célja, hogy a természetes nyelven írt kérdésekre képes legyen válaszolni közvetlenül táblázatok adatai alapján, és támogatja a numerikus műveleteket, mint például az összeadás, átlagolás, és az értékek összehasonlítása, ami a TQA-ban gyakori követelmény.

2. **TAPEX (Tabular Pre-trained Large Language Model for SQL Query Generation)**: A TAPEX (Liu és mtsai, 2022) egy transformer alapú modell, amelyet kifejezetten táblázatos adatok feldolgozására és azokon végrehajtott SQL-szerű lekérdezések támogatására terveztek. A TAPEX erőssége a strukturált adatok kezelésében rejlik, és hatékonyan képes összetett lekérdezések feldolgozására, például oszlopok közötti kapcsolatok feltárására vagy aggregált eredmények előállítására. A modell különösen alkalmas olyan feladatokra, ahol táblázatos adatokból kell specifikus, pontos válaszokat nyerni, akár egyszerű kérdésekre, akár összetett analitikai műveletekre.
3. **GPT-4 (Generative Pre-trained Transformer)**: A GPT-4 nagy méretű generatív modell, amely különösen jól teljesít a természetes nyelvi feladatokban. Ez a modell alkalmas bonyolultabb kérdések feldolgozására és a táblázatok kontextusából történő válaszadásra, bár a táblázatos struktúrára vonatkozó specifikus előtanításuk hiányában közvetlen finomhangolást igényelhetnek a TQA-feladatokban. Az OpenAI-tól a GPT4o (2024-08-06) és a GPT4o-mini (2024-07-18) modelleket teszteltük.
4. **LLaMA-3 (Large Language Model Meta AI)**: A LLaMA (Touvron és mtsai, 2023) egy Meta által fejlesztett LLM, amely különféle nyelvi feladatokra alkalmas, beleértve a táblázatos kérdés-válasz feladatokat is. Mivel többnyelvű modell, képes a magyar nyelvű adatokon való működésre, így a magyar nyelvű TQA kutatásában hasznosnak bizonyulhat, különösen a kisebb nyelvekre való adaptálhatóságának köszönhetően.
    a. **Llama 3.1/8**: Egy közepes méretű, 8 milliárd paraméteres modell, amely gyors és hatékony megoldásokat kínál egyszerűbb feladatokra.
    b. **Llama 3.1/70**: Egy nagyobb, 70 milliárd paraméterrel rendelkező modell, amely kiváló pontosságot biztosít, különösen a bonyolultabb problémák és összetettebb adatszerkezetek esetén.
    c. **Llama 3.2/1**: Egy apró, 1 milliárd paraméteres modell, amely elsősorban beágyazott rendszerekhez készült, bár pontossága elmarad a nagyobb modellektől.
    d. **Llama 3.2/3**: Egy kisméretű modell, amely jó egyensúlyt kínál a pontosság és az erőforrás-igény között, alkalmas mobil eszközökön való használatra is.
5. **Mistral NeMo**: egy új, 12 milliárd paraméteres modell, amelyet a Mistral AI csapata az NVIDIA-val együttműködve fejlesztett.

6. **Phi-3.5-mini**: egy könnyűsúlyú, szabadon hozzáférhető modell, amely a Microsoft Phi-3 modellcsalád része. A modell szintetikus adatokon és szűrt, nyilvánosan elérhető weboldalakon alapuló adatokat használ, különösen az érveléshez szükséges sűrű adatokat célozva.
7. **Qwen2.5**: A Qwen2.5 különböző alapnyelvi modelleket és utasításokkal hangolt modelleket kínál, amelyek legnagyobb tagja 72 milliárd paraméterrel rendelkezik. A modell jelentős javulásokat ért el az utasítások követésében, a hosszú szövegek (8K token felett) generálásában, a strukturált adatok (pl. táblázatok) értelmezésében és a strukturált kimenetek generálásában. A modell támogatja a hosszú kontextust, akár 128 ezer tokenig, és képes akár 8 ezer tokenes szövegek generálására. Többnyelvű modellként számos különböző nyelvű adatot használtak fel, beleértve a magyar nyelvű forrásokat is.
8. **SambaLingo-Hungarian-Chat**: egy csevegőmodell, amelyet magyar és angol nyelvű feladatokon tanítottak. A modell a SambaLingo-Hungarian-Base alapmodellen alapszik, amely a Llama-2-7b modellt adaptálja a magyar nyelvre a Cultura-X adatbázis magyar nyelvi részéből származó 59 milliárd token felhasználásával. A SambaLingo-Hungarian-Chat modellt közvetlen preferencia-optimalizálással taníották, hogy a válaszadás során a lehető legjobban kövesse az emberi utasításokat.
9. **PULI LlumiX 32K**: egy másik magyar nyelvű, 6.74 milliárd paraméteres modell, amelyet az OpenChatKit Github segítségével tanítottákf. Ez a LLaMA-2-7B-32K alapmodellre épül és magyar nyelvű adathalmazon előtanították. A kontextusablaka 32 768 token méretű, így hosszabb inputot is képes kezelni.
10. **Gemma.2 27B**: A Gemma a Google által kifejlesztett könnyűsúlyú, nyílt forráskódú modellcsalád, amely ugyanazon kutatás és technológia felhasználásával készült, mint a Gemini modellek. Ezeket a modelleket különféle szöveggenerálási problémákra tervezték, például kérdés-válasz feladatok, szövegértelmezés és érvelés.
11. **Cohere Aya Expanse 32B**: a Cohere For AI által kifejlesztett, nyílt súlyokkal rendelkező, kutatási célú, többnyelvű modell. 23 nyelvet támogat, ezek között sajnos a magyar nyelv nem szerepel.
12. **C4AI Command-R**: A C4AI Command-R egy 35 milliárd paraméteres kutatási célú, nagy teljesítményű generatív modell, szintén a Cohere For AI csapattól. Nyílt súlyokkal rendelkezik, és számos felhasználási esethez van optimalizálva, beleértve az érvelést, szövegezést és kérdés-választ. A modell többnyelvű generálásra is képes, 10 nyelvet támogat, ezek között sem szerepel a magyar nyelv.

A Táblázatos Kérdés-Válasz (TQA) feladatok megoldására különféle módszereket fejlesztettek ki

Az SQL-alapú módszerek az egyik legrégebbi megközelítést képviselik a TQA-feladatokban. Ezek a módszerek a természetes nyelvi kérdéseket SQL-lekérdezésre alakítják át, amelyeket a modell futtat a táblázaton. Az SQL-alapú megközelítések kifejezetten alkalmasak olyan helyzetekben, ahol a táblázatok relációs struktúrával rendelkeznek, és összetett kérdések esetén is működőképesek. Az SQL-alapú módszerek fő előnye, hogy lehetővé teszik a logikai és matematikai műveletek egyszerű végrehajtását, ugyanakkor speciális tudást és pontos lekérdezés-generálást igényelnek. Ilyen például a WikiSQL adathalmazon való modellezés, ahol a kérdések SQL-lekérdezésekkel kapcsolódnak a táblázatokhoz.

A Wikipédiából összeállított adathalmazunk táblázatai nagyon hiányosak. A Python könyvtárakban található, HTML táblázatokat értelmező függvények úgy alakítják át a Wikipédia táblázatait, hogy hiányos vagy elcsúszott cellákat eredményeznek, sok esetben az oszlopneveket sem tudják beazonosítani. Ezeknek a hiányában az SQL-alapú cellakeresés nem mindig lehetséges, így ezt a módszert nem fogjuk vizsgálni.

Dologzatomban az in-context-learning megközelítést követem, ami különösen a nagyméretű nyelvi modellek elterjedésével váltak népszerűvé. Ebben a megközelítésben a természetes nyelvű kérdést egy promptként használják, amely kiegészülhet további instrukciókkal, például a táblázat szerkezetének leírásával vagy kontextus meghatározásával. A prompt-alapú TQA módszerek előnye, hogy nem igényelnek strukturált lekérdezést, hanem közvetlenül képesek a kérdések és a táblázatok szöveges feldolgozására. Ezek a modellek a táblázatok és a kérdések kontextusának integrálásával tudják kinyerni a szükséges információkat, így különösen alkalmasak a hiányos vagy félig strukturált táblázatos feladatokra, de kevésbé pontosak a nagyon specifikus táblázat struktúrák feldolgozásában.

5. Összehasonlító eredmények

Az iterációs sebesség és a pontosság két fontos teljesítménymutató, amelyek segítenek értékelni a nyelvi modellek hatékonyságát. Az iterációs sebesség azt mutatja meg, hogy a modell hány kérdést dolgoz fel egy másodperc alatt. Minél magasabb az iterációs sebesség, annál gyorsabban képes a modell reagálni a kérdésekre. A pontosság pedig azt méri, hogy a modell válaszai milyen arányban egyezik meg a helyes válasszal. A magas pontosságú modellek képesek megbízható válaszokat generálni, amelyek a kérdések kontextusához illeszkednek. Mindkét mutató segít meghatározni, hogy egy modell mennyire alkalmas különböző alkalmazásokhoz.

Modellnév	Sebesség (it/s)	Pontosság átlag	Pontosság szórás
Llama3.1/8	0.34	0.681	0.060
Llama3.2/1	0.77	0.207	0.024
Llama3.2/3	0.53	0.530	0.028
Llama3.1/70	0.18	0.755	0.020
MistralNemo	1.89	0.667	0
Phi3.5Mini	0.71	0.463	0
Qwen2.5/72	0.08	0.752	0.025
SambalingoHungarian	0.23	0.37	0
PuliLumix	0.18	0.426	0
Gemma2/27	0.17	0.722	0
CohereAya/32	0.09	0.722	0
CohereCommandR	0.08	0.667	0
gpt-4o-mini	0.34	0.7406	0.037
gpt-4o	0.25	0.778	0.047

IV. ábra *modell eredménytáblázat*

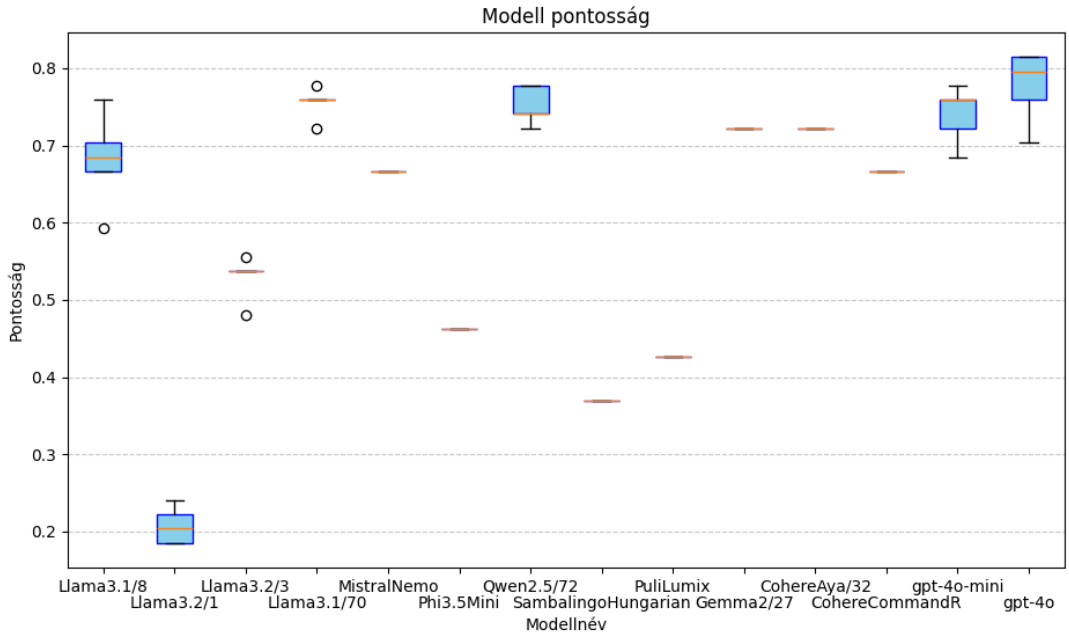
A modelleket Kormányzati Informatikai Fejlesztési Ügynökség (KIFÜ) Komondor nevű szuperszámítógépén teszteltük. A vizsgált modellek a következő mérési eredményeket adják vissza az összeállított táblázatos kérdés-válasz adathalmazon:

1. **Llama3.1/8**: A Llama3.1/8 modell közepes sebességgel dolgozik (0.34 iteráció/s) és átlagos pontossággal (0.681). Bár gyors válaszokat biztosít, a pontossága nem kiemelkedő, ami arra utal, hogy bár a válaszadási idő kedvező, a modell teljesítménye az elvártaknak nem minden esetben felel meg.
2. **Llama3.1/70**: A Llama3.1/70 modell lassabb (0.18 iteráció/s), viszont jobb pontossággal (0.755) rendelkezik, mint a legtöbb vizsgált modell. A megbízhatóságot és a válaszok helyességét tekintve ez a modell jobb teljesítményt mutat, de a sebessége jelentősen csökkenti a hatékonyságát, különösen, ha gyors válaszokra van szükség.

3. **Llama3.2/1:** A Llama3.2/1 pontossága (0.207) nagyon alacsony. Ez a vizsgált modellek közül a legkisebb.
4. **Llama3.2/3**: A Llama3.2/3 pontossága (0.530) jelentősen jobb az 1B társánál, de a többi modelltől elmarad. Ez a modell gyorsan generál válaszokat, azonban a pontossága nem megfelelő ahhoz, hogy magas minősítést kapjon, mivel a válaszok sok esetben nem teljesen helytállóak.
5. **MistralNemo**: A MistralNemo modell (1.89 iteráció/s) viszonylag gyors és a vizsgált modellek közt közepes pontossággal rendelkezik (0.667). Ez azt jelzi, hogy bár nem a leggyorsabb, megfelelő pontossággal képes válaszolni, de a teljesítménye nem éri el a kiemelkedő szintet, ha mindkét tényezőt figyelembe vesszük.
6. **Phi3.5Mini**: A Phi3.5Mini modell lassabb (0.71 iteráció/s), és a pontossága (0.463) nem kielégítő. Az alacsony sebesség és a gyenge pontosság együtt arra utal, hogy a modell nem biztosít elegendő megbízhatóságot vagy hatékonyságot a kérdésekre adott válaszokban.
7. **Qwen2.5/72**: A Qwen2.5/72 modell a leglassabb (0.08 iteráció/s) a tesztelt modellek között, és a mezőnyből kiemelkedik a pontosságával (0.752). Ez a vizsgált nyílt súlyú modellek közül a legjobban teljesítő modell, amely a lassúsága ellenére magas megbízhatóságot biztosít.
8. **SambalingoHungarian**: A SambalingoHungarian modell viszonylag lassú (0.23 iteráció/s), és a pontossága is alacsony (0.37). Ez a modell nem képes pontos válaszokat generálni, és a sebessége sem elegendő ahhoz, hogy a gyenge pontosságot kompenzálja.
9. **PuliLumix**: A PuliLumix szintén lassú (0.18 iteráció/s), és pontatlan (0.426). A modell nem rendelkezik a megfelelő sebességgel vagy pontossággal, hogy versenyezzen a legjobban teljesítő modellekkel.
10. **Gemma2/27:** A Gemma2/27 modell lassú (0.17 iteráció/s) és a pontossága megközelíti az elfogadható szintet (0.722). Ez a modell inkább jobban teljesít az ilyen kérdés-válasz feladatokban.
11. **CohereAya/32:** A CohereAya/32 modell hasonló pontossággal rendelkezik (0.722), de viszonylag lassú (0.09 iteráció/s). Ez a modell nagy pontossággal válaszol, de az alacsony iterációs sebesség miatt hosszabb válaszidővel kell számolni, ha nagy mennyiségű kérdés megválaszolására van szükség.
12. **CohereCommandR:** A CohereCommandR pontossága (0.667) elmarad a Cohere másik modelljétől.
13. **GPT4o-mini:** Az OpenAI modelljét API-n keresztül hívtuk meg, így a futtatási sebesség csak a többi OpenAI modell sebességével vethető össze (0.34), ez a gyorsabb, lényegesen

kevesebb erőforrást felhasználó megoldás. A pontossága a jobbak közé tartozik, 0.741 eredményt ért el.

14. **GPT4o**: Az OpenAI nagyobb modellje, lényegesen több erőforrást igényel, az API hívás valamivel lassabb volt (0.25), viszont pontosabb (0.778). Ezzel az eredménnyel ez a modell a vizsgált modellek között a legjobb.



V. ábra *az egyes LLMek pontosság a magyar TQA feladaton*

Összességében a nyílt és többnyelvű Qwen2.5/72B modell kiemelkedő eredményt nyújtott, a zárt GPT4o-mininél is pontosabb, nála csak a vizsgált GPT4o modellnek sikerült több kérdést megválaszolni helyesen. Hasonlóan erősnek bizonyult a Llama3.1/70B is. Velük szemben a legkisebb paraméterű modellek, mint például Phi3.5Mini és a Llama3.2 modellek a legkevésbé pontosak. A többi modell, mint a MistralNemo és a CohereAya/32, kiegyensúlyozott teljesítményt nyújtott a kérdés-válasz feladatok során annak ellenére, hogy nem magyar adathalmazon történt az előtanítás. A kifejezetten magyar nyelvre készített modellek (Sambalingo, Puli Lumix) nagyon gyengének bizonyultak, a méretében összemérhető Llama3.1/8 modell 25 százalékponttal teljesített jobban. Többszöri újrafuttatás esetén egyik modell eredményei sem változtak számottevően, amit az alacsony 0.06 alatti szórások mutatnak. A mért sebességnek látszólag semmi köze nincs a modellek méretéhez és erőforrásigényéhez, ebben inkább a Komondor erőforrás-szabályozása számít.

6. Összefoglalás

A dolgozat a táblázatos kérdés-megválaszolás feladatra összpontosított, amely során a modellnek strukturált adatokból, táblázatokból kellett válaszolni a feltett kérdésekre. A TQA-feladatok jelentősége az adatbázisokban és különböző dokumentumokban elérhető strukturált információ hasznosításában rejlett, amelyeket hatékony kérdés-válaszoló rendszerek segítségével lehet automatizálni. Az angol nyelven számos kutatás foglalkozott TQA-kkal, és több jól ismert TQA-adathalmaz is rendelkezésre állt, amelyek segítségével az algoritmusokat és nyelvi modelleket fejleszteni lehetett. Azonban magyar nyelven ezek az erőforrások korlátozottak maradtak, és nem állt rendelkezésre olyan TQA-adatbázis, amely magyar nyelven készült kérdések és válaszok vizsgálatát tette volna lehetővé.

Dolgozatomban bemutattam egy magyar nyelvű TQA-adathalmaz létrehozására kidolgozott eljárást, amely a magyar Wikipédia táblázatos adatait és szöveges tartalmát hasznosította. A Az eljárással 2000 magyar Wikipedia szócikk feldolgozásával 54 táblázatos kérdés-válasz pár gyűjtöttem automatikusan. A manuális ellenőrzés során 37 kérdés-választ fogadtunk el. Ez a magyar TQA-adathalmaz lehetőséget nyújtott a legmodernebb LLM-ek magyar nyelven való tesztelésére.

ére.

A dolgozatomban továbbá bemutattam egy tucat nagy nyelvi modellek teljesítményének és sebességének összehasonlító elemzését.

További kutatási terveim közt szerepel egy nagyobb kiértékelő adatbázis manuális ellenőrzés révén történő előállítása, majd ezen az adatbázison a különböző LLMek hibáinkat szisztematikus elemzése.

Algoritmikus szempontból a SQL átfordítás alapú megközelítést is tesztelni fogom, valamint megvizsgálom a kisebb modellek magyar nyelvű TQA feladatra finomhangolásának lehetőségeit.

Köszönetnyilvánítás

Köszönjük a lehetőséget, melyet a Kormányzati Informatikai Fejlesztési Ügynökség által üzemeltetett Komondor biztosított.

A kutatás az Európai Unió támogatásával valósult meg, az RRF-2.3.1-21-2022-00004 azonosítójú, Mesterséges Intelligencia Nemzeti Laboratórium projekt keretében.

Irodalomjegyzék

Akhtar, M., Cocarascu, O., Simperl, E.: Pubhealthtab: A public health table-based dataset for evidence-based fact checking. In: Findings of the Association for Computational Linguistics: NAACL 2022. pp. 1–16 (2022)

Akter, S.N., Yu, Z., Muhamed, A., Ou, T., Bäuerle, A., Cabrera, Á.A., Dholakia, K., Xiong, C., Neubig, G.: An in-depth look at gemini's language abilities. arXiv preprint arXiv:2312.11444 (2023)

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., és mtsai: A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In: Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 675–718 (2023)

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., és mtsai: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. pp. 1877–1901 (2020)

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y.T., Li, Y., Lundberg, S., és mtsai: Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:2303.12712 (2023)

Chen, W.: Large language models are few (1)-shot table reasoners. In: Findings of the Association for Computational Linguistics: EACL 2023. pp. 1120–1130 (2023)

Chen, W., Wang, H., Chen, J., Zhang, Y., Wang, H., Li, S., Zhou, X., Wang, W.Y.: Tabfact: A large-scale dataset for table-based fact verification. arXiv preprint arXiv:1909.02164 (2019)

Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., Wang, W.Y.: Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 1026–1036 (2020)

Cheng, Z., Dong, H., Jia, R., Wu, P., Han, S., Cheng, F., Zhang, D.: Fortap: Using formulas for numerical-reasoning-aware table pretraining. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1150–1166 (2022a)

Cheng, Z., Xie, T., Shi, P., Li, C., Nadkarni, R., Hu, Y., Xiong, C., Radev, D., Ostendorf, M., Zettlemoyer, L., és mtsai: Binding language models in symbolic languages. arXiv preprint arXiv:2210.02875 (2022b)

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., és mtsai: Palm: Scaling language modeling with pathways. Journal of Machine Learning Research 24(240), 1–113 (2023)

Dahl, D.A., Bates, M., Brown, M.K., Fisher, W.M., Hunicke-Smith, K., Pallett, D.S., Pao, C., Rudnicky, A., Shriberg, E.: Expanding the scope of the atis task: The atis-3 corpus. In: Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994 (1994)

Eisenschlos, J., Gor, M., Mueller, T., Cohen, W.: Mate: Multi-view attention for table transformer efficiency. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 7606–7619 (2021)

Eisenschlos, J., Krichene, S., Mueller, T.: Understanding tables with intermediate pre-training. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 281–296 (2020)

Fabbri, A.R., Ng, P., Wang, Z., Nallapati, R., Xiang, B.: Template-based question generation from retrieved sentences for improved unsupervised question answering. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4508–4513 (2020)

Golchin, S., Surdeanu, M.: Time travel in llms: Tracing data contamination in large language models. arXiv preprint arXiv:2308.08493 (2023)

Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: The atis spoken language systems pilot corpus. In: Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990 (1990)

Herzig, J., Nowak, P.K., Mueller, T., Piccinno, F., Eisenschlos, J.: Tapas: Weakly supervised table parsing via pre-training. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4320–4333 (2020)

Jin, N., Siebert, J., Li, D., Chen, Q.: A survey on table question answering: recent advances. In: China Conference on Knowledge Graph and Semantic Computing. pp. 174–186. Springer (2022)

Lehmberg, O., Ritze, D., Meusel, R., Bizer, C.: A large public corpus of web tables containing time and context metadata. In: Proceedings of the 25th international conference companion on world wide web. pp. 75–76 (2016)

Lewis, P., Denoyer, L., Riedel, S.: Unsupervised question answering by cloze translation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4896–4910 (2019)

Li, Z., Wang, W., Dong, L., Wei, F., Xu, K.: Harvesting and refining question-answer pairs for unsupervised qa. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 6719–6728 (2020)

Liu, Q., Chen, B., Guo, J., Ziyadi, M., Lin, Z., Chen, W., Lou, J.G.: Tapex: Table pre-training via learning a neural sql executor. arXiv preprint arXiv:2107.07653 (2021)

Min, S., Chen, D., Hajishirzi, H., Zettlemoyer, L.: A discrete hard em approach for weakly supervised question answering. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2851–2864 (2019)

Mueller, T., Piccinno, F., Shaw, P., Nicosia, M., Altun, Y.: Answering conversational questions on structured data without logical forms. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 5902–5910 (2019)

Pasupat, P., Liang, P.: Compositional semantic parsing on semi-structured tables. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 1470–1480 (2015)

Puri, R., Spring, R., Shoeybi, M., Patwary, M., Catanzaro, B.: Training question answering models from synthetic data. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 5811–5826 (2020)

Shi, P., Ng, P., Nan, F., Zhu, H., Wang, J., Jiang, J., Li, A.H., Chakravarti, R., Weidner, D., Xiang, B., és mtsai: Generation-focused table-based intermediate pre-training for free-form question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 11312–11320 (2022)

Shi, T., Zhao, C., Boyd-Graber, J., Daumé III, H., Lee, L.: On the potential of lexico-logical alignments for semantic parsing to sql queries. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 1849–1864 (2020)

Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., és mtsai: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., és mtsai: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)

Wang, B., Shin, R., Liu, X., Polozov, O., Richardson, M.: Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7567–7578 (2020)

Wang, Z., Dong, H., Jia, R., Li, J., Fu, Z., Han, S., Zhang, D.: Tuta: Tree-based transformers for generally structured table pre-training. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 1780–1790 (2021)

Wang, Z., Zhang, H., Li, C.L., Eisenschlos, J.M., Perot, V., Wang, Z., Miculicich, L., Fujii, Y., Shang, J., Lee, C.Y., és mtsai: Chain-of-table: Evolving tables in the reasoning chain for table understanding. arXiv preprint arXiv:2401.04398 (2024)

Xie, T., Wu, C.H., Shi, P., Zhong, R., Scholak, T., Yasunaga, M., Wu, C.S., Zhong, M., Yin, P., Wang, S.I., és mtsai: Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In: 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022 (2022)

Zhong, V., Xiong, C., Socher, R.: Seq2sql: Generating structured queries from natural language using reinforcement learning. arXiv preprint arXiv:1709.00103 (2017)

Zhou, F., Hu, M., Dong, H., Cheng, Z., Cheng, F., Han, S., Zhang, D.: Tacube: Pre-computing data cubes for answering numerical-reasoning questions over tabular data. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 2278–2291 (2022)

Zhu, F., Lei, W., Huang, Y., Wang, C., Zhang, S., Lv, J., Feng, F., Chua, T.S.: Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 3277–3287 (2021)

Függelék

A dolgozat elkészítéséhez a ChatGPT webes felülete került felhasználásra kódgeneráláshoz, a kód dokumentációjának előkészítéséhez, a kód strukturálásához, és a dolgozat alapvető struktúrájának generálásához.