Egy magyar nyelvű táblázatos kérdésmegválaszolás kiértékelő adatbázis

Tóth Gábor[1], Farkas Richárd[1], Szántó Zsolt[1]

[1]Szegedi Tudományegyetem, Informatikai Intézet
Szeged, Árpád tér 2.
gabor.toth.103@gmail.com
{rfarkas,szantozs}@inf.u-szeged.hu

Kivonat Cikkünkben nagy nyelvi modellek táblázat értelmezési képességeit vizsgáljuk. Ehhez készítettünk egy magyar nyelvű kiértékelőadatbázist, amiben Wikipédiaról származó táblázatok és azok alapján megválaszolható kérdés-válasz párok találhatók. A kérdés-válasz párokat a táblázat környezetében található szövegek alapján automatikusan generáltuk, majd ezeket kézzel ellenőriztük. Az így kapott korpuszon összehasonlítottuk 14 nagy nyelvi modell pontosságát. Azt konkludálhatjuk, hogy 2024 végén a 8 milliárd paraméteres modellek körülbelül 10 százalékponttal, míg a 30 millárd körüli modellek 5 százalékponttal teljesítenek rosszabbul, mint a GPT4o.
Kulcsszavak: kérdésválaszolás, táblázat, nagy nyelvi modellek

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 években a nagy nyelvi modellek (Large Language Models, LLM-ek) figyelemreméltó fejlődése (Brown és mtsai, 2020; Chowdhery és mtsai, 2023; Touvron és mtsai, 2023) átalakította az tábláztokból történő információkinyerést is. 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 angol nyelvre optimalizáltak, egyre nagyobb igény mutatkozik arra, hogy más nyelveken, például magyarul is sikeresen alkalmazhatók legyenek.

Az egyik érdekes alkalmazási terület a táblázatos kérdésmegválaszolási feladat (TQA) (Jin és mtsai, 2022), amely során a modellnek egy táblázat alapján kell a feltett kérdésekre választ adnia. Egészen pontosan a TQA feladat során a

bemenet egy vagy több táblázat és egy természetes nyelven megfogalmazott kérdést, amelyet a táblázat(ok)ban található információk alapján kell megválaszolni. 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 elérhető információk aggregálását is.

Ebben a cikkben bemutatunk egy magyar nyelvű TQA adathalmaz-előállító eljárást, ami javarészt automatikusan képes azt létrehozni magyar Wikipédia szócikkek táblázatos adatait és a hozzájuk kapcsolható szöveges tartalmakat hasznosítva. Az ezekből az adatokból készített TQA adathalmaz lehetőséget nyújt az LLM-ek tesztelésére magyar nyelvű adaton, valamint a TQA algoritmusok hatékonyságának vizsgálatára.

Cikkünkben továbbá több nagy nyelvi modell pontosságát és sebességét hasonlítjuk össze az elkészült magyar nyelvű TQA adathalmazon.

2. TQA megközelítések

Az angol nyelvű TQA feladatra számos megoldás született az elmúlt néhány évben. Az encoder-decoder modellek, mint a TAPEX (Liu és mtsai, 2021), és az MLM (Masked Language Modelling) módszerek (Herzig és mtsai, 2020), feldolgozzák a strukturált táblázatos adatokat majd a szemantikus térben ezek alapján történik meg a kérdésmegválaszolás. Ebben a megközelítésben a modellek közvetlenül állítják elő a válaszokat anélkül, hogy a természetes nyelvi kérdéseket átalakítanák. Ugyanakkor ezek a módszerek memóriakorlátaik miatt nehezen alkalmazhatók nagyméretű, valós adathalmazokon és az MLM bemenetének hosszlimitációja is problémát jelent.

Egy másik megközelítés a kérdések formális lekérdezésekké való átalakítása. Ezek a módszerek a természetes nyelvű kérdéseket valamilyen formális lekérdező nyelvre, például SQL-re alakítják át, amelyet aztán a táblázatos adatok lekérdezésére használnak (Shi és mtsai, 2020). Bár ezek a módszerek hatékonyak a strukturált adatbázisokban történő lekérdezésekben, korlátozottak nem szabványos táblázatok esetében, és a kérdések SQL-re történő pontos fordítása is nehézségekbe ütközhet. Megjegyezzük, hogy léteznek hibrid megközelítések is, amelyek ötvözik a formális leképezések és az encoder-decoder módszerét, például a TAPAS (Herzig és mtsai, 2020) vagy a TAGOP (Zhu és mtsai, 2021) modellek.

A mai modern LLM-ek már táblázatos struktúrákat is tudnak kezelni, így az elmúlt 2 évben a zero-shot és few-shot megközelítés lett egyeduralkodó. Cikkünkben mi is ezt a megközelítést követjük, és 14 LLM-hez szerkesztünk promptot, amivel magyar TQA teljesítményüket hasonlítjuk össze.

3. Egy Magyar Nyelvű TQA adatbázis

3.1. Adatgyűjtés

A manuális annotáció nagyon drága. Ahhoz, hogy ezt a költséget csökkentsük, egy nagyrészt automatikus módszert dolgoztunk ki magyar nyelvű TQA kiértékelőadatbázis előállítására. A magyar nyelvű TQA adatbázis létrehozása a magyar Wikipédián található táblázatok és kapcsolódó szövegek automatizált gyűjtésével és feldolgozásával történt. 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 formába. 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 tartalmakat elemeztük, mivel ezek gyakran magyarázzák vagy részletezik a táblázatok tartalmát. Ezeket a táblázathoz köthető mondatokat használjuk majd fel kérdések generálására.

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

Mivel a Wikipédián előfordulhat, hogy egyes táblázatok ismétlődnek vagy hasonló adatokat tartalmaznak, ezért eltávolítottuk a redundáns bejegyzéseket, hogy az adathalmaz minél változatosabb és informatívabb legyen.

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.

A táblázatok celláiban szereplő adatokat típus szerint csoportosítottuk (pl. dátum, szám, szöveg), hogy a későbbi feldolgozás során könnyebben használhatók legyenek. Például az időpontokat és számadatokat formailag és tartalmilag is egységesítettük.

3.2. Kérdések generálása

A táblázat közvetlen környezetében található szövegek egy része a táblázat tartalmát magyarázza, kiemelve annak fontos, érdekes részeit. Ezeket a magyarázatokat vettük a kérdésgenerálás alapjául mégpedig úgy, hogy az azokban megfogalmazott állításokat fogalmaztattuk át kérdésekké. Vegyük például az alábbi szövegrészletet, ami a Szeged népességéről szóló táblázat mellett található:

`a legmagasabb befejezett iskolai végzettség szerint az érettségi végzettséggel rendelkezők élnek a legtöbben a városban 52 347 fő, utánuk a következő nagy csoport a diplomával rendelkezők 44 502 fővel`

Ebből az alábbi kérdést generáltuk:
`Milyen végzettséggel rendelkezők élnek a legtöbben a városban?`

A kontextusban található szövegek alapján lehetőség van többféle kérdéstípus generálására is:

- **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?`**''**.

Kiválasztjuk azokat a mondatokat, amelyek tartalmazzák a közvetlen közelben szerepelő táblázat valamely celláinak tartalmát. Ezeknek az állító mondatoknak az átfogalmazásával generálunk kérdéseket, azaz egy mondat és abban előforduló érték (ami egyben egy táblázat valamely cellájának értéke) alapján olyan kérdést generálunk, ami az értékre kérdez rá. Jelen verzióban ezt a kérdésséalakítást a GPT4o-mini modell promptolásával valósítottuk meg.

Az így összeállított adathalmaz strukturájában a WikiTableQA-t (Pasupat és Liang, 2015) követtük. Megjegyezzük, hogy a WTQA-ban szerepelnek listával megválaszolható kérdések, azaz olyanok, amik több helyes válasszal rendelkeznek. Az egyszerűség kedvéért, mi kiszűrtük azokat a kérdéseket, amire több helyes válasz is adható egy táblázatból. 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.
- **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 az volt a célunk, hogy csak a megbízható táblázatokat és állításokat elemjük ki, hogy minél kevesebb manuális ellenőrzésre legyen szükség. A vizsgált 2000 szócikkből összesen 1086 olyan szövegrészletet vagy mondatot azonosítottunk, amelyben szerepeltek a táblázat közelében (Wikipedia szócikk fejezete) a 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 (a szereplő legalacsonyabb szintű alfejezete) 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 szövegben, valamint a túl hosszú mondatok kizárását, amelyek kevésbé informatívak vagy nem egyértelműek.

Ezek az eljárások a vizsgált szócikkekben a helyes adatpontokat megtartották. Az így szűkített adathalmaz végül néhány száz potenciális adatpontot tartalmazott, amelyeket manuálisan validáltunk. Ez az eljárás további szűrők bevezetésével még pontosabbá tehető.

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. Ez azt jelenti, hogy megfelelően felismert táblázat esetén a feladat lényegesen könnyebb lehet más adatbázisokhoz képest.

Az automatikus szűrőkön átment példákat manuálisan ellenőriztük. Egyrészt értékeltük, hogy a GPT4o-mini modell helyes magyarságú és tényleg a cellaadatra vonatkozó kérdést generált-e. Másrészt vizsgáltuk, hogy a kérdés megválaszolható-e kizárólag a táblázat tartalma alapján. Összesen 54 érvényes adatpontot találtunk tökéletesen helyesnek. Megközelítésünkben így 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. 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, így feltételezhető, hogy a módszerrel akár tízezres nagyságrendű adathalmaz is kinyerhető.

Az elkészült magyar TQA adatbázis publikusan elérhető a `https://github.com/szegedai/TableQA` címen és felhasználható a magyar nyelvű nagy nyelvi modellek (LLM-ek) táblázatértelmezési képességének kiértékelésére.

4. LLMek kiértékelése a TQA adatbázison

Az elkészült adatbázis felhasználásával, kezdeti kísérletként 14 nagy nyelvi modellt hasonlítottunk össze. A nyelvi modelleket próbáltuk minél nagyobb változatossággal kiválasztani, a szolgáltatásként elérhető két OpenAI GPT4 variánson kívül szerepeltek kisebb és nagyobb modellek 1 milliárd paramétertől 72 milliárdig, valamint kipróbáltunk kifejezetten magyar nyelvre fokuszálva továbbtanított modelleket a PULI LlumiX 32K (Yang és mtsai, 2024) és a SambaLingo-

Modellnév	Paraméterek száma (milliárd)	Pontosság	Sebesség (it/s)
Llama 3.2 1B	1	0.207	0.77
Llama 3.2 3B	3	0.530	0.53
Llama 3.1 8B	8	0.681	0.34
Llama 3.1 70B	70	0.755	0.18
Mistral NeMo	12	0.667	1.89
Phi-3.5-mini	3.8	0.463	0.71
Qwen2.5-72B	72	0.752	0.08
Gemma.2 27B	27	0.722	0.17
Cohere Aya Expanse 32B	32	0.722	0.09
C4AI Command-R	35	0.667	0.08
GPT4o-mini (2024-07-18)	?	0.741	0.34
GPT4o (2024-08-06)	?	0.778	0.25

1. táblázat. Nyelvi modellek hatékonyságának a vizsgálata az elkészült táblázat-adatbázis fölött.

Hungarian (Csaki és mtsai, 2024). A 12 lokálisan futatott modell esetén a futásidők méréséhez 4db A100-as videókártyát használtunk. Ezen felül a két OpenAI modell API szolgáltatás alapú volt.

Minden LLM-nek ugyanazt a feladatszöveget adtuk meg, viszont ezt az adott modell ajánlasainak megfelelően helyeztük el a promptban. A pontosság kiszámításához csak teljesen egyező választ fogadtunk el, azaz ha az LLM válasza karakterre megegyezett a helyes cella szövegével. Ha az LLM több megoldását is adott, akkor ha a listában szerepelt a helyes válasz, akkor azt elfogadtuk. Minden LLM-et 5 alkalommal futtattunk minden kérdésre, az 1. táblázat az 5 futtatás eredményének az átlagát tartalmazza.

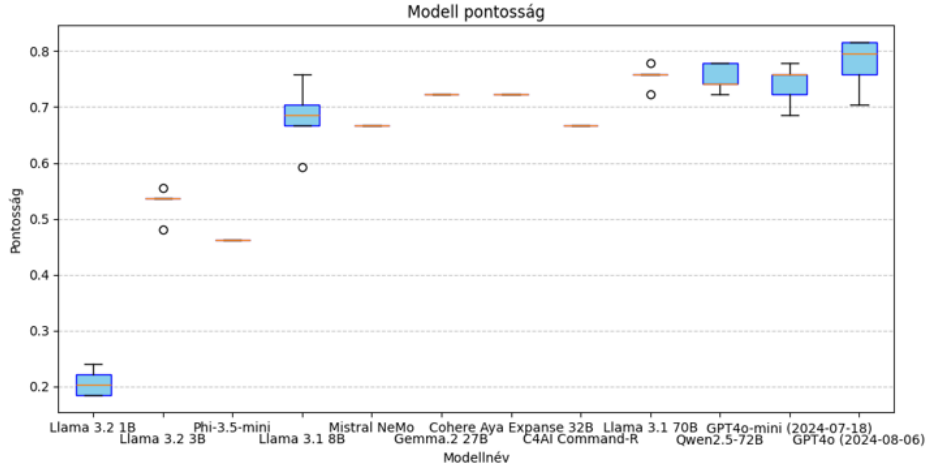
Egyes modellek, mint a MistralNemo és a CohereAya/32, kiegyensúlyozott teljesítményt nyújtott a kérdés-válasz feladatok során annak ellenére, hogy a dokumentációjuk alapján nem használtak magyar adathalmazt az előtanítás alatt.

1. ábra jól szemlélteti, hogy az LLM-ek legújabb generációjánál a modell méretének függvényében szinte monoton módon nő azok pontossága.

Modellnév	Paraméterek száma (milliárd)	Pontosság	Sebesség (it/s)
PULI LlumiX 32K	7	0.444	0.18
SambaLingo-Hungarian	7	0.407	0.23

2. táblázat. Nyelvi modellek eredményei az elkészült táblázat adatbázison megengedő kiértékeléssel.

A PULI LlumiX 32K modell instrukciókra hangolt változata nem érhető el szabadon és a SambaLingo-Hungarian-nak is csak chatelésre finomhangolt változata érhető el. Mivel ezek a modellek nincsenek kérdésmegválaszolásra hangolva,



1. ábra: Modellek pontossága méret szerint rendezve.

ezeknél a modelleknél a generált kimenetet kézzel ellenőriztük és megengedően értékeltük, hogy a generált szöveg megválaszolja-e a kérdést. Ezek eredménye a 2. táblázatban található.

5. Hiba analízis

A kérdéseket két kategóriába osztottuk az alapján, hogy egyszerű információt kérnek (38 db), amihez elég a megfelelő oszlop és sor kiválasztása vagy valamilyen összetettebb feladatot kell az LLM-nek megoldani (16 db), pl. elemek megszámolása, vagy a legnagyobb/legkisebb érték kiválasztása valamilyen feltétel alapján. Ezek alapján megvizsgáltuk a legjobban és legrosszabbul teljesítő 2-2 modell hatékonysága mennyire tért el feladattípusonként.

Modellnév	Egyszerű	Összetett
Llama 3.1 70B	0.784	0.688
Qwen2.5-72B	0.758	0.717
Phi-3.5-mini	0.518	0.313
Llama 3.2 1B	0.242	0.125

3. táblázat. Modellek teljesítményének összehasonlítása egyszerű és összetett kérdések felett.

A 3. táblázatban látható, hogy bár minden modell jobban teljesített az egyszerű kérdések megoldásában a Qwen2.5-72B esetén ez mindössze 4 százalékpont különbséget jelent, míg a Llama 3.1 70B esetén körülbelül 10 százalékpontot.

A kisebb modellek esetén látható, hogy a bonyolultabb feladatokon lényegesen rosszabbul teljesítettek.

Összehasonlítottuk, hogy a modellek ugyanarra a feladatra mennyire konzisztensen válaszolnak az 5 futtatás alapján. A két hasonló architektúrájú Llama modellt nézve az 1 millárd paraméteres modell átlagosan 3.1 különböző választ generált 5 kísérletből, míg a 70 milliárd paraméteres modell nem csak jobban válaszolt, hanem sokkal konzisztensebb is volt, átlagosan 1.72 különböző választ adva.

Az egyik legösszetettebb kérdésforma volt, ahol nem csak valamilyen matematikai műveletet kellett elvégezni (mint például a legnagyobb elem megtalálása) hanem a művelet által megkapott érték alapján kellett egy másik sor vagy oszlop tartalmát visszaadni, ilyen kérdés például a **‘‘Milyen végzettséggel rendelkezők élnek a legtöbben a városban?’’**. Az ilyen összetettségű következtetést elváró kérdéseken a két legkisebb modell rendre rossz választ adott, még akkor is, ha ezt az adatot egy 2-3 soros táblázatból kellett kinyerni, míg a 70 milliárdos Llama sok esetben még ezekre az bonyolult összetett kérdésekre is helyesen tudott válaszolni. Annak ellenére, hogy sokszor helyesen meg tudta válaszolni ezeket a kérdéseket is a Llana 3.1 70B, például a hasonló logikájú **‘‘Melyik évben éltek a legtöbben a városban?’’** kérdésre a ötből négyszer rossz évszámot adott válaszul.

6. Összegzés

Cikkünkben beszámoltunk egy automatikus eljárással létrehozott magyar nyelvű táblázatos kérdésmegválaszoló adatbázissal kapcsolatos első kísérleteinkről. Magát az eljárást, a szűrőket, a kezelt kérdéstípusokat, az állító mondatból kérdés generálást még számos ponton fejleszteni tervezzük a jövőben. Például lehetne a listás kérdéseken felül időbeli és logikai kapcsolatokra kérdező állító mondatokat is generálni: Például **„Hány év telt el Magyarország uniós csatlakozása óta?’’** vagy **„Melyik esemény történt előbb?’’**, amivel az ilyen kérdésekkel az LLMek logikai következtetési képességét vizsgálhatnánk. Ha ezek a fejlesztések elkészülnek egy nagyobb méretű kiértékelő adatbázist tervezünk építeni és közzétenni.

Cikkünkben 14 darab nagy nyelvi modellt értékeltünk ki egy 54 elemű, magyar nyelvű táblázatos kérdésmegválaszolási feladaton. Az eredményekből az látszik, hogy a modellek méretétől függ azok pontossága, illetve, hogy a két publikusan elérhető magyarra tanított LLM egyértelműen rosszabbul teljesít, mint a méretben hasonló, sok nyelvre tanított, de újabb architektúrát alkalmazó versenytársaik.

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.

Hivatkozások

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)

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)

Csaki, Z., Li, B., Li, J.L., Xu, Q., Pawakapan, P., Zhang, L., Du, Y., Zhao, H., Hu, C., Thakker, U.: SambaLingo: Teaching large language models new languages. In: Sälevä, J., Owodunni, A. (szerk.) Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024). pp. 1–21. Association for Computational Linguistics, Miami, Florida, USA (Nov 2024), https://aclanthology.org/2024.mrl-1.1

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)

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)

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)

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)

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)

Yang, Z.G., Dodé, R., Ferenczi, G., Hatvani, P., Héja, E., Madarász, G., Ligeti-Nagy, N., Sárossy, B., SZaniszló, Z., Váradi, T., és mtsai: The first instruct-following large language models for hungarian. In: IEEE 3rd Conference on Information Technology and Data Science (CITDS) Proceedings. University of Debrecen (2024)

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)