Adattárházak, adatbányászati technológiák

Ádám Harmados, Czikollai Bálint, Sándor Balázs

March 2024

1. Bevezetés

Adattárházak, adatbányászati technológiák csoportos beadandóként az volt a feladatunk, hogy válasszunk egy nem közismert lokálisan futtatható nagy nyelvi modellt (LLM) ezt futtassuk lokálisan és teszteljük, majd építsünk rá egy nem triviális alkalmazást, majd erről írjunk és mutassuk be.

Beadandónkban a "dlite-v2-1-5b" modellre építettünk fel egy REST szervert amely egy Google Chrome bővítményhez csatlakozik ami képes a kijelölt szövegnek címet adni, kulcsszavakat kiemelni és összegezni. Ennek a rendszernek a felépítéséről írunk az alábbiakban.

Elősször a modell alapjául szolgáló GPT-2-ről majd a tanító adathalmazról, a tényleges modellről, ezt követően a megvalósított programról és az eredményekről számolunk be.

A projektünk: https://github.com/SandorBalazsHU/elte-ik-adatbanyaszat

A modellünk elérhetősége: https://huggingface.co/aisquared/dlite-v2-1$_5b$

1.1. Munkamegosztás

Az ötlet és az alapfelvetés kidolgozása közös munka volt, ezt követően Czikollai Bálint dolgozta fel a GPT-2 működéséről szóló részt, Ádám Harmados készítette a tanító adathalmaz és a modell ismertetését, majd Sándor Balázs végezte a programozást és a tesztelést és ezen szekciók megírását.

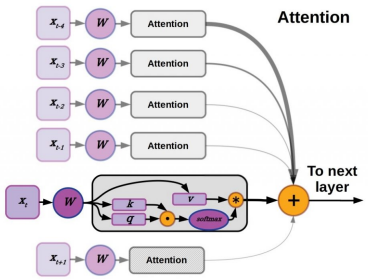
2. GPT-2

Az általunk feldolgozott "dlite-v2-1-5b" modell a GPT-2 továbbfejlesztett, újratanított változata. Tekintsük át ennek felépítését és működését.

A Generative Pre-trained Transformer 2 (GPT-2) nyelvmodellt az OpenAi alkotta meg és adta ki. Ebből négy féle paraméter nagyságú verziót adtak ki Small, Medium, Large, Xl amik sorban 124 millió, 355 millió ,774 millió, 1.5 billió paraméterüek. A mi alap modellünk az a GPT-2 Large, ami 10-ből 6.72 hitelességi pontot kapott alig elmaradva a GPT-2 XL-től és jóval megelőzve a Small és Medium modelleket. A modell egy előképzett angol nyelvű modell, amely CLM (Casual Language modeling) célre használt.

2.1. Transzformátormodell

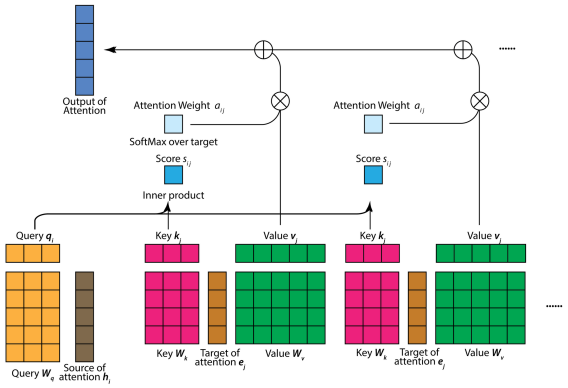
A GPT-2 a GPT-1 utódja ami közötti két fő különbség a megnövelt paraméterek száma és a betanítási adatkészlet növelése. Ezt a GPT-3 váltotta fel, ami már nem nyílt forráskódu. A GPT-2 generatív, előrebetanított transzformátor-architektúrával rendelkezik, amely egy transzformátormodellt valósít meg. A transzformátormodell a "figyelmet" használja fel egy ismétlődés- és konvolúció alapú architektúra.



1. ábra. Az Attention mechanizmus működése

A "figyelem" a gépi tanulásban egy olyan mechanizmus, amely intuitív módon utánozza kognitiv figyelmet. Ez a "figyelem" igazából súlyokat rendel adott szavakhoz, vannak puha illetve kemény súlyok míg a kemény súlyok előedzés és finomhangolás után nem változnak, addig a puha súlyok minden futás után változnak.

A figyelemfelhívó mechanizmusok által a modell szelektíven kiválasztja a számára legfontosabb részeket a bemeneti szövegből. Ez a modell lehetővé teszi a nagy mértékű párhuzamosítást.



2. ábra. Az Attention mechanizmus működése 2

2.2. WebText

A párhuzamosítás lehetővé tette, hogy CommonCrawl adatainak felhasználása helyett létrehoztak egy saját adathalmazt a WebText-et. A WebText-ben megadott minőség cikkeket, blogokat stb. és a Wikipédiát eltávolították belőle mert ez sok más forrás adatforrása, ezzel megelőzve az információ ismétlődést.

A GPT-2 modell több rétegből áll, amelyek önállóan dolgozzák fel a bemeneti szekvenciát, majd az eredményt továbbadják a következő rétegnek. Ez az architektúra lehetővé teszi a modell számára, hogy kontextusukban értelmezze és generálja a szöveget.

2.3. Előtanítás és Tanítási Folyamat

A GPT-2 modellt egy nagy szövegkorpuszon előre tanítják, amely különböző forrásokból származó szövegeket tartalmaz, például könyveket, cikkeket és weboldalakat. Az előtanítási folyamat során a modell megtanulja a szövegstatisztikai tulajdonságait és a szavak közötti kapcsolatokat. Ez a folyamat lehetővé teszi a modell számára, hogy generáljon koherens és értelmes szöveget adott kontextusban.

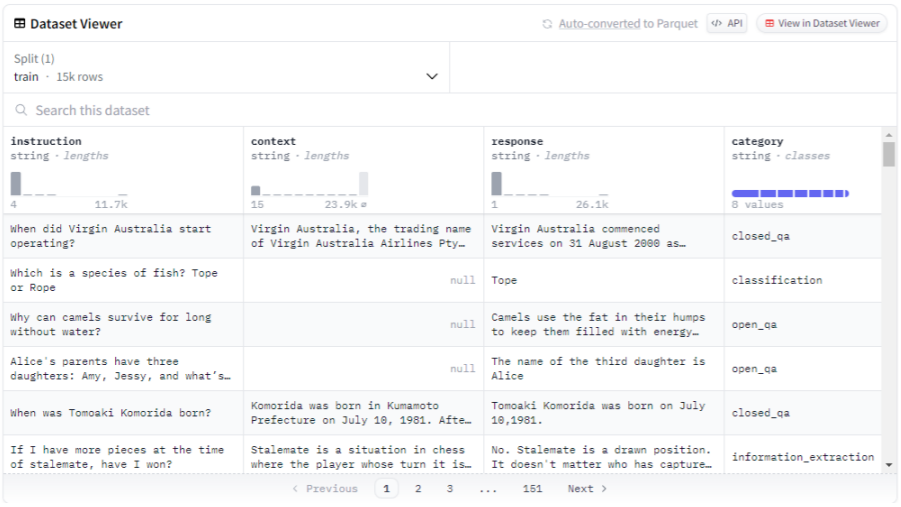
2.4. Generatív Képességek és Alkalmazások

A GPT-2 modell kiemelkedő generatív képességeket mutat, képes koherens és kontextusban releváns szöveget generálni különböző stílusokban és témákban. Az alkalmazások széles skáláját kínálja, például a szöveggenerálást, szövegösszegzést, fordítást, párbeszédeket és kérdésekre való válaszolás. A GPT-2 modellt használják kutatásokban, iparban és kreatív projektekben egyaránt, hogy automatizáljanak feladatokat, generáljanak tartalmat és javítsák az ember-gép interakciót. A GPT-2 tovább lehet fejleszteni, hogy egy bizonyos szakterületen legyen hatékonyabb. Mivel az utódai (GPT-3,GPT-4) nem nyílt forráskódúak ezért a GPT-2 hasznos tanulók számára is. Betekintést kaphatnak egy LLM kódjába.

2.5. Hatások és Kihívások

A GPT-2 modell jelentős hatást gyakorolt a természetes nyelvi feldolgozás területére, elősegítve további kutatásokat és fejlesztéseket nagy méretű nyelvi modellek terén. Ugyanakkor a modell használata számos kihívást is felvet, például az etikai és társadalmi kérdéseket, az adatbiztonságot és a megbízhatóságot. A GPT-2 képes hihető szövegrészleteket generáljon azonban minél hosszabb szöveget annál jobban elkezdenek látszani a hiányosságai például romlik a nyelvtan, ismétlések stb.

3. A tanító adathalmaz



3. ábra. Az adathalmaz felépítése (Forrás: Az adathalmaz weboldala)

A választott modellünk készítői a GPT-2 modellt tanították a "databricks-dolly-15k" adathalmazon, hogy egy kis méretű mégis magas párbeszédi képességekkel rendelkező angol nyelvű modellt készítsenek.

A "databricks-dolly-15k" adatbázis eredetileg nem az AI Squared által lett létrehozva. Ez az adatbázis a Databricks által lett összeállítva és létrehozva.

Ezt az adatbázist bármilyen célra fel lehet használni, legyen az akadémiai vagy kereskedelmi, a Creative Commons Attribution-ShareAlike 3.0 Unported License feltételei alatt.

3.1. Adathalmaz áttekintése

A "databricks-dolly-15k" egy olyan gyűjtemény, mely több mint 15,000 rekordot tartalmaz, melyeket több ezer Databricks alkalmazott hozott létre annak érdekében, hogy nagyméretű nyelvi modellek képesek legyenek magas interaktivitást mutatni. A Databricks alkalmazottakat arra kérték, hogy hozzanak létre felhívás / válasz párokat nyolc különböző utasításkategóriában.

Az adatgenerálási folyamat felénél a hozzájárulóknak lehetőségük volt más hozzájárulók által feltett kérdésekre válaszolni. Arra kérték őket, hogy fogalmazzák át az eredeti kérdést, és csak azokat a kérdéseket válasszák ki, amelyekre reálisan válaszolhatnak.

Bizonyos kategóriák esetében a hozzájárulókat arra kérték, hogy adjanak meg referenciákat, melyeket a Wikipédiáról másoltak.

Ezekből az elemekből állt össze az adathalmaz.

4. A modell

Az AI Squared dlite-v2-1.5b egy nagyméretű nyelvi modell, amely az OpenAI nagy GPT-2 modelljéből származik, és egy 15 ezer rekordból álló adathalmazon a korábban ismertetett Databricks "Dolly 15k" adathalmazán van finomhangolva annak érdekében, hogy segítse a beszélgetésalapú képességek javítását.

A dlite-v2-1.5b létrehozása során a készítők célja egy kis méretű, lokálisan is futtatható modell létrehozása volt amely mégis erőteljes párbeszédi képességekkel rendelkezik.

4.1. Kockázatok és korlátok

A dlite-v2-1.5b nem a legfrissebb nyelvi modell. A dlite-v2-1.5b egy kísérleti technológia, és mint minden kísérleti technológia esetében, az AI Squared azt kéri a technológia potenciális felhasználóitól, hogy alaposan teszteljék a képességeit a használat előtt. Továbbá, a modell néha nem kívánt viselkedéseket mutathat. Ezek közé tartozhatnak például a ténybeli pontatlanságok, elfogultságok, sértő válaszok, toxicitás és hallucinációk. Ahogyan bármely más LLM esetében.

4.2. Modell teljesítménymutatók

A DLite család minden modelljének különböző modell-értékeléseket készítettek az EleutherAI LLM Evaluation Harness alapján. A modelleredményeket átlagos pontszám szerint növekvő sorrendben rendezzük. Ezek a mutatók további bizonyítékok arra, hogy a DLite modellek egyike sem a legfrissebb, inkább azt mutatják, hogy a beszélgetéshez hasonló viselkedéseket az LLM-ek majdnem függetlenül lehet képezni a modellmérettől ami különösen érdekes a jelen helyzetben.

Model	arc_challenge	arc_easy	boolq	hellaswag	openbookqa	piqa	winogrande
dlite-v2-124m	0.199659	0.447811	0.494801	0.291675	0.156	0.620239	0.487766
gpt2	0.190273	0.438131	0.487156	0.289185	0.164	0.628945	0.51618
dlite-v1-124m	0.223549	0.462542	0.502446	0.293268	0.17	0.622416	0.494081
gpt2-medium	0.215017	0.490741	0.585933	0.333101	0.186	0.676279	0.531176
dlite-v2-355m	0.251706	0.486111	0.547401	0.344354	0.216	0.671926	0.52723
dlite-v1-355m	0.234642	0.507576	0.600306	0.338478	0.216	0.664309	0.496448
gpt2-large	0.216724	0.531566	0.604893	0.363971	0.194	0.703482	0.553275
dlite-v1-774m	0.250853	0.545875	0.614985	0.375124	0.218	0.698041	0.562747

4. ábra. A modell teljesítménye (Forrás: A modell weboldala)

5. A program

A modellt lokálisan és Google Colab felületen is futtattuk és teszteltük. Kezdetben a model készítői által rendelkezésünkre bocsájtott szkriptel futtattuk a modelt, ezután sajátot készítettünk aminek a GPU gyorsítás állt a fókuszában.

Az első ötletünk a kvízkérdések generálása volt, de a modell nem volt kellőképpen determinisztikus. Néha sikerült kérdést generálnia, néha nem, ezért megváltoztattuk a koncepciót és szöveg címadásra, kulcsszava generálására és összeggzésre használtuk amire sokkal alkalmasabbnak bizonyult.

Készítettünk egy Jupyter notebook-ot ami a megfelelő Pytorch, TensorFlow, Transformers, Accelerate könyvtárak segítségével GPU gyorsítással futtatunk.

Ehhez Python-ban egy REST API-t készítettünk ami képes kéréseket kezelni összeállítani, elküldeni a modell felé és az eredményeket visszaküldeni a Chrome plugin felé.

A Chrome plugin a böngészőben minden kijelölt szövegre meghívható, négyféle metódust képes futtatni és az eredményeket egy listában gyűjti is.

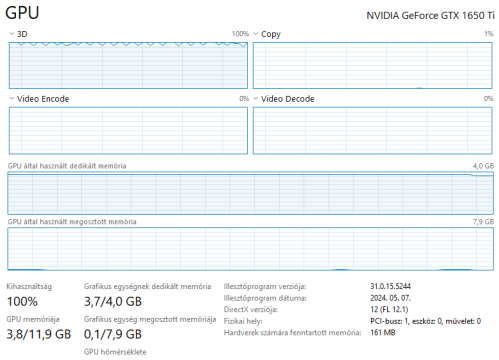
Tekintsük most át a fejlesztés folyamatát.

5.1. Előfeltételek

A telepítést a projekt repository-jában lévő TXT és Jupyter notebook tartalmazza, ezzel segítve a későbbi felhasználást. A modell futtatásához Python környezet szükséges. Mi ezt Windows rendszeren készítettük el, de más rendszeren is kialakítható. 3.10.11 verziójú vagy ennél régebbi Python-ra van szükség, mivel a nyelvi tanulási könyvtárak jelenleg ezen működnek. Rendszergazdai jog és külön meghajtó célszerű, mert a telepített könyvtárak elég nagyok. Célszerű virtuális Python környezetet kialakítani, hogy a telepített könyvtárak elkülönüljenek a lokális futtatókörnyezettől. Ezt követően célszerű a Jupyter notebook-ot telepíteni, hogy könnyebben dolgozhassunk a python projektekkel. A Chrome plugin fejlesztéséhez csak egy Visual Studio Code-ra és egy Chrome böngészőre van szükség.

5.2. Lokális GPU futtatás

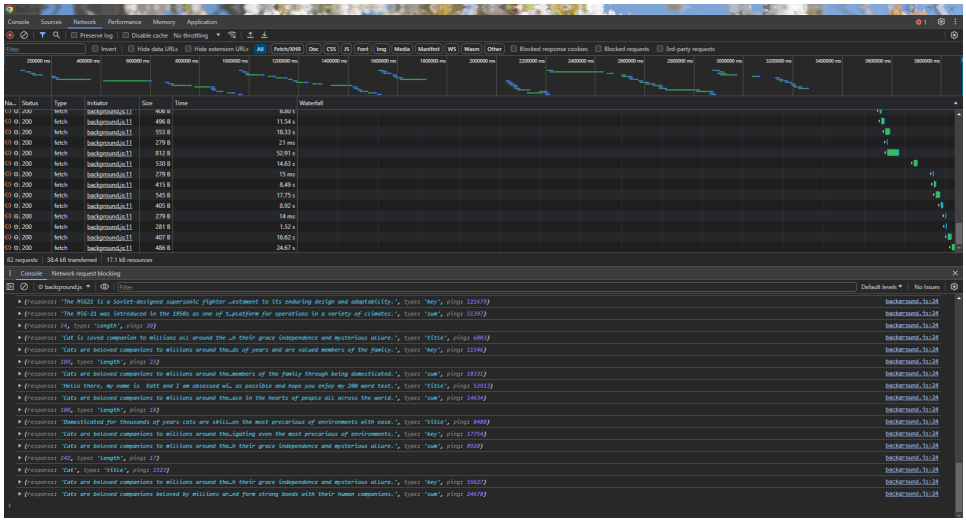
A futtatás előtt célszerű előkészíteni a rendszert arra, hogy GPU támogatással futtathassuk a modellünket jelentősen felgyorsítva azt. Az első lépés az Nvdia driver és a CUDA driver toolkit telepítése. Ezeket ellenőrizzük, ezután telepítsük az Accelerate, Pytorch, Tensorflow és Transformers könyvtárak GPU-s felhasználásra tervezett verzióit. Ezt követően klónozzuk a modell repository-ját és módosítsuk az eredeti scriptet, hogy GPU-n fusson a letöltött lokális modellel. Ezután már kérdéseket tehetünk fel a modellnek. Ezen lépések alaposan ki vannak fejtve a Jupyter notebook-ban és van egy SERVER notebook ami a magyarázat nélküli futtatást tartalmazza. Ezt célszerű futtatni a rendszer használatakor.



5. ábra. A megoldásunk GPU használata

5.3. Szerver

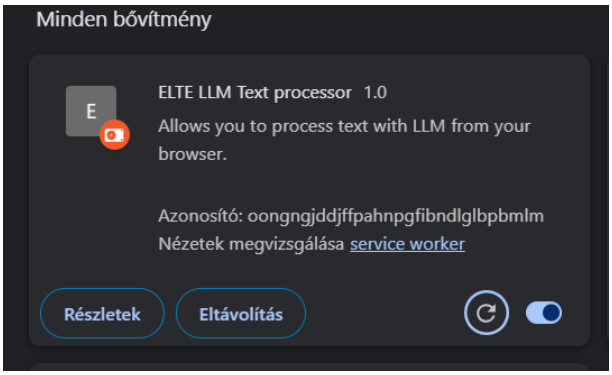
Ahhoz hogy a modellt összekapcsolhassuk a böngésző bővítménnyel egy szerverre van szükség, mivel a model Python környezetben fut és REST API implementálása mellett döntöttünk és JSON szabvány üzenetekkel kommunikálunk így a Flask könnyű szerverkönyvtár segítségével implementáltuk a REST API-t CORS-al a hitelesített kommunikációhoz. A szerveren négy végpontot hoztunk létre. Egyet a token számításhoz, egyet a címadáshoz, egyet a kulcsszavak generálásához és egyet az összegzéshez. A szerver a beérkező szövegen tisztítást végez, összekapcsolja a szükséges utasítással majd futtatja a modelt, végül visszatér az eredménnyel.



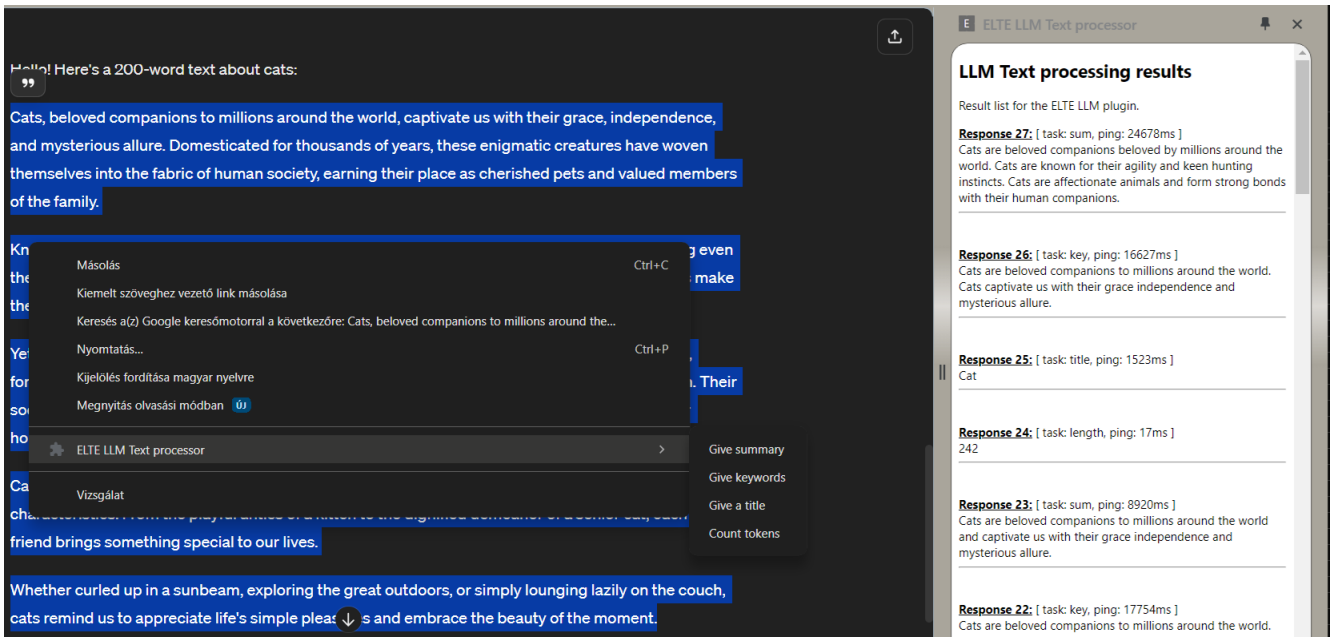
6. ábra. A szerver és a plugin közötti hívások

5.4. Chrome plugin

A modell kényelmes használatához egy Google Chrome böngésző bővítményt fejlesztettünk ami a kijelölt szövegek jobbgombos menüjében jelenik meg. Itt kérhetjük a tokenszámlálást, a címadást, az összegzést és a kucssző generálást. A kérés után megnyílik a plugin oldalablaka ahol egy rövid várakozás után megkapjuk a választ. A plugin listázza az eddigi válaszokat mik később törölhetőek. A kérés típusa és válaszideje is itt szerepel.

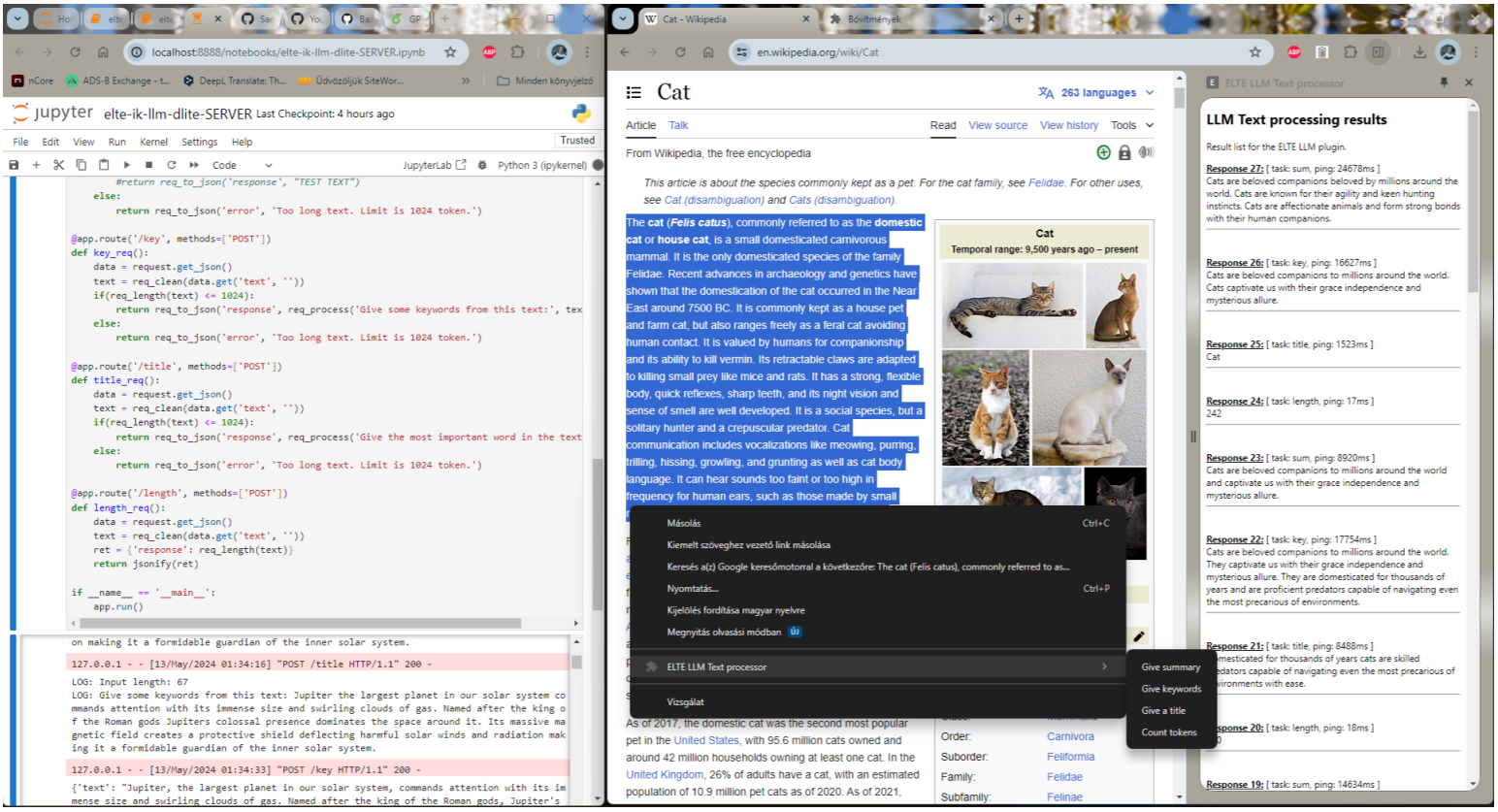


7. ábra. A plugin



8. ábra. A plugin

Itt látható a háttérben futó szerver, egy Wikipedia cikk amin kijelöltünk egy részletet és a plugin oldalmenüje amiben a válaszok találhatóak.



9. ábra. A teljes alkalmazás

6. Mérések

A fejlesztés után méréseket végeztünk. Három 200 szavas szöveget generáltattunk ChatGPT-vel. Egyet a macskákról könnyedd nyelvezettel, egyet a MiG-21-es vadászrepülőgépről közepesen nehéz nyelvezettel és egyet a Jupiter bolygóról nehéz nyelvezettel. Mindhárom szövegből vettünk egy rvid, egy közepes és egy teljes mintát és ezen futtattuk a rendszer mindhárom képességét és mértük a válaszidőket. Ezt a mérést kétszer is elvégeztük. Ezek eredményei láthatóak az alábbi táblázatokban.

Text set	tittle	key	sum
Jupiter - 67	4679	4296	22552
Jupiter - 132	6051	64459	64459
Jupiter - 235	9246	122059	179727
MiG-21 - 77	1141	8044	14202
MiG-21 - 139	56641	14366	27196
MiG-21- 249	31956	22501	68735
Cats - 54	3211	16622	16601
Cats - 100	9188	8133	13436
Cats - 242	1447	12751	12751

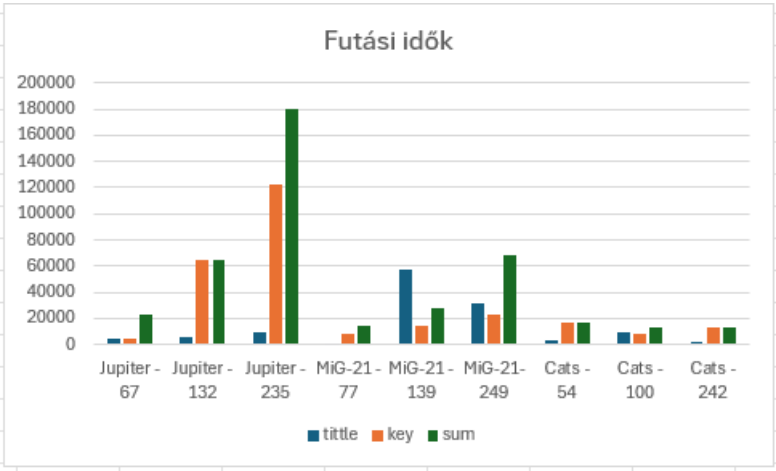
10. ábra. A megoldásunk futási ideje a három tesztszöveg három különböző méretű mintáján mindhárom funkció esetében. 1. teszteset

Text set	tittle	key	sum
Jupiter - 67	4222	7198	10184
Jupiter - 132	6154	36640	70686
Jupiter - 235	9311	199017	9416
MiG-21 - 77	23082	2727	15116
MiG-21 - 139	18406	20998	19795
MiG-21- 249	13934	121679	51397
Cats - 54	6803	18331	14634
Cats - 100	8488	17754	8920
Cats - 242	1523	16627	24678

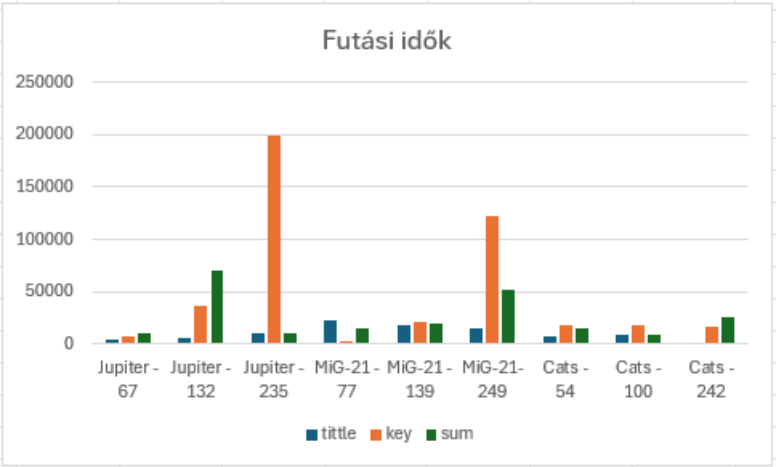
11. ábra. A megoldásunk futási ideje a három tesztszöveg három különböző méretű mintáján mindhárom funkció esetében. 2. teszteset

7. Eredmények

Értelmezzük a mérések eredményeit. Az előző adatokat az alábbi grafikonokon ábrázoltuk. Jól leolvasható, hogy a szöveg hosszával nagyjából lineárisan nő a futási idő, a szöveg komplexitásával viszont exponenciálisan. A második mérés viszont jelzi, hogy a modell hajlamos néha kiugró válaszidőkre is. Az is látható, hogy a címadás a legkönnyebb a kulcsszó keresés közepes a szöveg összegzés a leglassabb feladat.



12. ábra. A megoldásunk futási ideje a három tesztszöveg három különböző méretű mintáján mindhárom funkció esetében. 1. teszteset



13. ábra. A megoldásunk futási ideje a három tesztszöveg három különböző méretű mintáján mindhárom funkció esetében. 2. teszteset

8. Befejezés

Összegezve a tapasztalatokat, a modell kis mérete ellenére hatékonyan és aránylag gyorsan működik, de hajlamos a kiszámíthatatlan működésre és a kiugró futásidőkre, de az esetek többségében jó teljesítményt nyújt. Az általunk kialakított szerver és plugin kialakítás pedig könnyű hozzáférhetőséget és használhatóságot biztosít.