A TUDOMÁNY, AZ OKTATÁS ÉS A KÖZGYŰJTEMÉNYI KISZOLGÁLÁS ÚJ INFORMATIKAI SZINERGIÁI

NETWORKSHOP 2026
35. Országos Informatikai Konferencia

2026. március 31–április 2.
Debreceni Egyetem, Debrecen

Szerkesztette: Tick József, Kokas Károly, Holl András

HUNGARNET Egyesület
Budapest, 2026

HUN-REN
Magyar Kutatási Hálózat

NETWORKSHOP

Szerkesztette: Tick József, Kokas Károly, Holl András

Tipográfia és tördelés: Vas Viktória
Korrektúra: Danyi Melinda
Angol nyelvi lektor: Lukács Katalin

Networkshop 2026 konferencia előadásainak közleményei
Debreceni Egyetem, Debrecen
2026. március 31–április 2.

ISBN 978-615-6792-29-7
DOI: https://doi.org/10.31915/NWS.2026

Kiadja a HUNGARNET Egyesület
az MTA Könyvtár és Információs Központ közreműködésével
Budapest
2026

Borítókép: freepik.com

DOI: 10.31915/NWS.2026.24

A RAG RENDSZER LEHETŐSÉGEI A LEVÉLTÁRI KUTATÁSTÁMOGATÁSBAN

THE APPLICABILITY OF RAG TECHNOLOGY IN THE HUNGARIAN NATIONAL ARCHIVES

Varga Emese (varga.emese@mnl.gov.hu)
Havas Ádám (havas@helion.hu)
Bánki Zsolt (bankizsolt@mnl.gov.hu)

Absztrakt

A mesterségesintelligencia-alapú technológiák egyre hangsúlyosabban jelennek meg a közgyűjteményi digitális szolgáltatásokban. A felhasználói igények már nem csupán a digitalizált források elérésére, hanem azok mélyebb, tartalmi feltárhatóságára, az eddigi kutatási módszertanokkal nehezen átlátható összefüggések felszínre hozatalára és célzott kereshetőségére irányulnak. Erre a kihívásra kínál új lehetőséget a RAG (Retrieval-Augmented Generation) technológián alapuló szolgáltatás.

Az alkalmazás egy felhasználói kérdésre levéltári források alapján szemantikus keresést futtat, majd egy nagy nyelvi modell (LLM) segítségével választ generál. A megbízható válasz előfeltétele az intézmény által biztosított jó minőségű forrás.

A tanulmány a Magyar Nemzeti Levéltár és a HELION Kft. együttműködésében fejlesztett alkalmazás levéltári integrálhatóságát mutatja be, tekintettel arra, hogy a RAG-alapú megoldás miként illeszkedik a meglévő online tartalomszolgáltatásokhoz. Fókuszba kerül a RAG fejlesztésének és levéltári implementálásának folyamata is.

E téren bemutatásra kerül, hogy a kétlépcsősen működő RAG-alkalmazás az első nyelvi modellhívás során megvizsgálja a kontextust, majd a felhasználói kérdésből kulcsszavakat és név entitásokat (NER) nyer ki, amelyek a dokumentumkeresés alapját képezik. FAISS (Facebook AI Similarity Search)-alapú szemantikus keresést kombinál Oracle adatbázis context indexre támaszkodó szűréssel. A keresés eredményeként kapott szövegrészek adják a második nyelvi modellhívás bemenetét. Mindkét híváshoz az OpenAI gpt-4.1 modelljét használja, a kérdésre adott végleges válasz streamelt formában érkezik.

Kulcsszavak: mesterséges intelligencia, RAG, szemantikus keresés, nagy nyelvi modellek, levéltári digitalizáció

Abstract

Artificial intelligence-based technologies are becoming increasingly prominent in public collection digital services. User demands are no longer limited to accessing digitized resources, but also include deeper content exploration, the uncovering of connections that are difficult to discern using existing research methodologies, and targeted searchability. A service based on RAG (Retrieval-Augmented Generation) technology offers a new opportunity to meet this challenge.

The application runs a semantic search based on archival sources and then generates answers to user questions using a large language model (LLM). A prerequisite for reliable answers is the high quality of the sources provided by the institution.

The article will discuss the archival integration of the application developed in collaboration between the Hungarian National Archives and HELION Kft., with a focus on how the RAG-based solution fits into existing online content services. It will focus on the development of RAG, the process of its implementation in archives, and the experiences of its developers.

In this context, we will demonstrate that the two-stage RAG application examines the input during the first language model call, then extracts keywords and named entities (NER) from the user's question, which form the basis for document search. It combines FAISS (Facebook AI Similarity Search)-based semantic search with Oracle database context index-based filtering. The text fragments obtained as a result of the search provide the input for the second language model call. It uses the OpenAI gpt-4.1 model for both calls, and the final answer to the question is delivered in streamed form.

Keywords: artificial intelligence, RAG, semantic search, large language models, archival digitization

Bevezetés

Napjainkban a közgyűjteményi digitális tartalomszolgáltatásokkal szemben már nem csupán a puszta elérhetőség és a nyílt hozzáférés az elvárás. A kombinált keresési metódusok – például a kulcsszavas és a fazettás keresés – mellett egyre nagyobb igény mutatkozik a szemantikus keresésre, az AI-alapú tartalomelemzési lehetőségekre[1] és a kontextualizált válaszokat nyújtó rendszerekre is.

Ezek az AI-alapú eszközök a tudományos kutatás fontos eszközévé válnak, főként a szakirodalom-keresés és -elemzés, illetve a tartalomgenerálás terén, de a torzítások veszélye jelentős, emiatt a kutatóknak kritikus szemlélettel kell kezelniük az AI által generált ajánlásokat.[2] Egy kutatástámogató szolgáltatásnál ezért fontos, hogy a válaszok ellenőrzött forrásokra épüljenek, és hivatkozásokkal legyenek alátámasztva. Erre kínál megoldást a RAG technológia.

A RAG (Retrieval Augmented Generation[1]) egy külső dokumentumgyűjteményből először visszakeresi a felhasználói kérdés szempontjából releváns információkat, majd egy nagy nyelvi modell (LLM[2]) segítségével választ generál a felhasználói kérdésekre.[3] Nemzetközi példák mutatják, hogy a RAG rendszerek történeti dokumentumokra is alkalmazhatók. Különböző kísérletek vannak levéltári anyagokon[4] és történeti újságcikkeken[5] alapuló rendszerek kialakítására, illetve különböző típusú kulturális örökségi adatok RAG általi, egységes tudásbázisba való szervezhetőségére.[6]

A technológia magyar levéltári használhatóságát az MNL és a Helion kft. által fejlesztett webalkalmazással demonstráljuk, amely Nagy Imre első kormányának minisztertanácsi jegyzőkönyveire[3] vonatkozó kérdések megválaszolását teszi lehetővé, és ezáltal egy új módszert kínál a levéltári anyagok kutatásához. Az alkalmazásba a tisztított, forráskiadványokhoz[4] készült PDF dokumentumokat használtuk. A technológia alkalmas arra, hogy nemcsak faktuális, hanem értelmező, definíciós, generatív kérdésekre is releváns választ adjon, így rámutathat összefüggésekre, lehetséges kapcsolatokra személyek, intézmények

1 A technológia működési elvét *visszakeresésre épülő válasz generálás*ként írhatjuk le magyarul a legpontosabban.

2 LLM: Large Language Model. Bővebben lásd: Naveed, Humza et.al.: A Comprehensive Overview of Large Language Models, *ACM Transactions on Intelligent Systems and Technology*, 2025/5, 1–72. https://doi.org/10.1145/3744746. [Megtekintve: 2026.07.08.]

3 Az eredeti forrásdokumentumok elérhetők az Adatbázisok Online felületén, lásd: Nagy Imre első kormánya 1953. július 4–1955. április 18., https://adatbazisokonline.mnl.gov.hu/adatbazis/minisztertanacsi-jegyzokonyvek-1944-1965/hierarchia. [Megtekintve: 2026.05.06.]

4 A forráskiadványok megjelenései: Nagy Imre első kormányának minisztertanácsi jegyzőkönyvei I. 1953. július 10. – 1954. január 15., szerk. Baráth Magdolna, Gecsényi Lajos, Nagy Imre Alapítvány, Magyar Nemzeti Levéltár, Budapest, 2018.; Nagy Imre első kormányának minisztertanácsi jegyzőkönyvei II. 1. rész, január 29. – 1954. július 2., szerk. Baráth Magdolna, Gecsényi Lajos, Nagy Imre Alapítvány, Magyar Nemzeti Levéltár, Budapest, 2022.; Nagy Imre első kormányának minisztertanácsi jegyzőkönyvei II. 2. rész, 1954. július 6. – 1954. október 26. Nagy Imre Alapítvány Magyar Nemzeti Levéltár, Budapest, 2022.

és események között, illetve időbeli és térbeli kontextusba helyezheti az információkat. Ugyanakkor nem helyettesíti a forráskritikát és a levéltári anyagok alapos kutatását, amelyek továbbra is a kutató feladatai maradnak.

A RAG levéltári alkalmazhatósága

A levéltári anyagok AI-módszerekkel történő kutatását, vizsgálatát a tech cégek által kínált megvalósításokkal szemben olyan megoldással kell lehetővé tenni, amit az MNL képes kontrollálni. Az MNL igényeire szabott alkalmazás fejlesztése mellett sem magától értetődő, hogy annak miért kell a RAG technológiát használnia. Miért nem lehet a modellnek egyben átadni az egész dokumentumkorpuszt, és az alapján várni a válaszokat?

Napjaink nagy nyelvi modelljei elméletileg képesek hatalmas dokumentummennyiség fogadására egyetlen promptban, de ennek alkalmazása komoly problémákat vonna maga után:

- **Költség:** Az API[5]-k token[6]-alapú elszámolásúak. Sok millió token „beküldése" minden egyes kérdésnél rendkívül drága lenne, míg a RAG csak a releváns részeket (a visszakeresés eredményeit) küldi el, töredék áron.
- **Késleltetés:** Minél nagyobb a kontextus, annál lassabb a modell válaszideje. Például egy 1 millió tokenes kérés feldolgozása percekig is eltarthat (az alkalmazásunk által kezelt dokumentumok összes terjedelme kb. 7–8 millió token).
- **Figyelemvesztés:** Bár a modellek fejlődnek, még mindig megfigyelhető, hogy a hatalmas szövegek közepén elrejtett információkat kevésbé hatékonyan találják meg, mint a szöveg elején vagy végén lévőket.

A RAG működési elve egyszerű: a felhasználó által feltett kérdés alapján futtatni kell egy (vagy több) keresést, ezek eredményei alapján egy nagy nyelvi modell (LLM) választ ad. A megvalósítás több döntési pontot hozott:

- Milyen LLM-et használjunk? Helyben működőt vagy szolgáltatásként igénybevehetőt?
- Hogyan működjön a keresés?
- A több felhasználóval párhuzamosan futó kommunikációt az alkalmazás miképp különítse el úgy, hogy eközben mindenki számára megmaradjon a saját kontextusa?

5 API (Application Programming Interface, magyarul alkalmazásprogramozási felület)

6 A nagy nyelvi modellek nem szavakkal, hanem tokenekkel dolgoznak, amelyek szavak, szótöredékek vagy írásjelek lehetnek. Angol nyelven 1 token nagyjából 4 karakternek vagy 0,75 szónak felel meg.

A RAG technológia levéltári alkalmazhatósága szempontjából a válasz minőségét tartottuk a legfontosabbnak. A helyben működő LLM-mel szemben fontos érv volt a jelentős hardverköltség, valamint a hardver beszerzésének és üzembe helyezésének időigénye. Ezek alapján a szolgáltatásként igénybe vehető LLM mellett döntöttünk, ami API-hívások formájában használható. 2025-ben, a fejlesztés idején közzétett LLM összehasonlításokban nagyon jól szerepelt az OpenAI ***gpt-4.1*** modellje, ezért az alkalmazásba ezt építettük be.

A forrásanyag előfeldolgozása

Kész megoldások egész sora[7] használható PDF fájlok RAG rendszerbe illesztésére. Ezek azonban csak a legegyszerűbb felépítésű PDF-eknél adnak elfogadható eredményt. Beépített algoritmusaik nem értelmezik a szövegeket és könnyen vezethetnek tartalmi szempontból rossz szerkezetű eredményhez.

A prototípusban használt közel 3000 oldal összterjedelmű PDF fájlokból úgy nyertük ki a szöveget, hogy annak tördelése, szerkezete és tartalma megfeleljen a szemantikus keresés és a nagy nyelvi modell számára. Az anyagot 10.487 szövegdarabra bontottuk: 380-tól 1300 karakterig terjedő méretűek, átlag 1113 karakteresek (kb. 400 token). Az átfedés kb. 150 karakter.

A szemantikus kereséshez mindegyik szövegdarabhoz egy-egy vektort rendeltünk. Igen magas (1536) dimenziószámú vektorokról van szó, amiket az OpenAI ***text-embedding-3-large***[8] modelljével állítottunk elő. Ezt a folyamatot szokás *beágyazás*nak nevezni, mert ezzel egy szöveget (szót, mondatot, dokumentumot) beemelünk egy matematikai térbe.

A visszakeresés megvalósítása

A hasonló jelentésű szövegdarabok közel kerülnek egymáshoz ebben a matematikai térben, amire ráépíthető a szemantikus keresés: A felhasználói kérdés kulcsszavai és kifejezései alapján ugyanezzel a módszerrel vektort képezünk. Ezután megkeressük azokat a szövegdarabokat, amelyek vektorai közel állnak a kérdés vektoraihoz, vagyis tartalmilag a legrelevánsabbak. Az LLM ezekből a kiválasztott szövegrészekből generálja a választ. Az alkalmazásba a FAISS (Facebook AI Similarity Search[9]) keresőmotort építettük be, amelyet a keresés hatékonyságának és megbízhatóságának fokozása érdekében szemantikus kereséssel kombináltunk, az Oracle saját Text keresőmotorja segítségével.

---

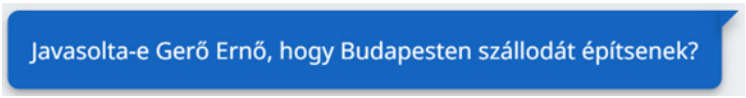
7 Például: pypdf, PyMuPDF, PyMuPDF4LLM, pdfminer.six, pdfplumber, pypdfium2, OCRmyPDF, docTR, Unstructured, Docling, stb.

8 Bővebben lásd: https://developers.openai.com/api/docs/models/text-embedding-3-large [Megtekintve: 2026.07.08.]

9 A FAISS (Facebook AI Similarity Search) a Meta által fejlesztett, nyílt forráskódú szoftverkönyvtár, amely lehetővé teszi nagy mennyiségű adathalmaz gyors átkutatását és hasonlósági keresését. Bővebben lásd: https://ai.meta.com/tools/faiss/ [Megtekintve: 2026.07.08.]

Kulcsszavak kinyerése, NER (Named Entity Recognition)

A kombinált keresés hatékonyságához nagyon fontos, hogy a felhasználó által megfogalmazott kérdésből kiemeljük azokat a szavakat és különösen azokat a tulajdonneveket, amikre a keresést futtatni kell. A folyamat első lépése során (az 1. LLM hívás keretében) történik meg ez, ahogyan azt az alábbi egyszerű példa mutatja egy adott felhasználói kérdés esetében:

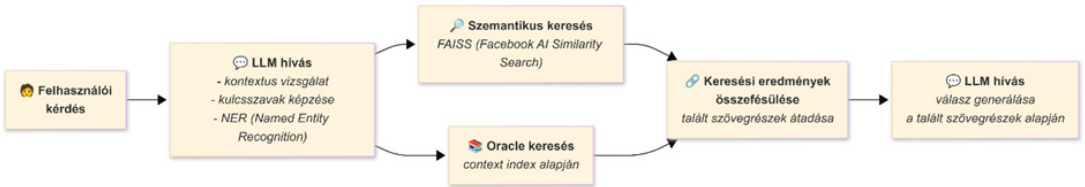


Az 1. LLM hívás eredménye (NE = named entity):

```
Kulcsszavak
├── 🧍 Gerő Ernő
│   └── Típus: személy (NE)
├── 📍 Budapest
│   └── Típus: hely (NE)
└── 🏨 szálloda építés
    └── Típus: egyéb (nem NE)
```

A válasz generálásának folyamata

Az LLM-hívások folyamatát, a válasz generálásának logikáját az alábbi ábra mutatja be:



Fontos eleme a folyamatnak az LLM-hívások paraméterezése és a prompt, amit a felhasználói kérdéssel és a keresési eredményekkel egyidőben kap meg a modell.

Előzmények (kontextus) figyelembevétele

Az alkalmazás minden egyes felhasználói munkamenethez (session-höz) két párhuzamos szálon követi az előzményeket.

1. A kontextusvizsgáló (és kulcsszókinyerő) LLM-hívás folyamatosan követi a saját előzményeit az alkalmazás használatának megkezdésétől a böngészőlap bezárásáig vagy kilépésig (illetve az F5 billentyű lenyomásáig).
2. A választ generáló LLM-hívás addig követi az előzményeket, amíg a kontextusvizsgálat alapján szükséges. Az előzményektől független, új kérdés hatására törli őket (ez természetesen nem azt jelenti, hogy a felületről eltűnnek).

Az előzmények figyelembevétele és szükség szerinti törlése két szempont miatt fontos:

- A párbeszéd folyamán a kérdések gyakran implicit módon hivatkoznak az előző kérdésre, ilyen esetekben fontos az előző kérdéshez kapott kulcsszavak továbbvitele.
- Ha viszont a kérdés újnak tekinthető, akkor éppen az a fontos, hogy előzmények nélkül, csak a szükséges kulcsszavak felhasználásával fusson le a keresés. Ennek költségvonzata is van, el kell kerülni a fölösleges tokenfelhasználást.

A válaszok értékelése

A demó alkalmazás gyakorlati használhatóságát levéltáros és történész szakemberek bevonásával is teszteltük. A vizsgálat célja az volt, hogy különböző típusú tesztkérdések által kiderüljön, hogy a kutatók/felhasználók a chatbot-alapú szolgáltatást mennyire tekintik valódi segítségnek a kutatómunkájukban. Elégedettek-e a jelenlegi demóverzióval a publikus bevezetéshez? Relevánsak és pontosak-e a válaszok?

A tesztelők összesen 516 kérdést tettek fel a rendszernek, 1-től 5-ig adható pontszámmal értékelték a válaszokat, amelynek átlaga 4,3 lett.

Emellett az alábbi 7 kategória egyikébe is besoroltak minden egyes választ:

Válasz besorolása	**Részarány (%)**
értékelhetetlen, hibás	1,0
téves, a forrásokban foglaltaknak nem felel meg	1,6
hiányos, nem tartalmazza a lényeget	3,3
hallucinációt tartalmaz	1,6
elfogadható, de nem elég tartalmas	11,6
jó, tartalmazza a lényeget	42,8
kitűnő, tartalmas és részletes	38,2

Az eredmények arra utalnak, hogy a rendszer működése alapvetően megfelel a felhasználói elvárásoknak. A válaszok 81%-át jónak vagy kitűnőnek értékelték, a hallucinációk aránya alacsony volt, negatív értékelés főképp a válaszok hiányosságára vonatkozott.

A kifejezetten gyenge értékelések aránya mindössze 2,6% volt, ami arra utal, hogy a teljesen sikertelen válaszok ritkán fordultak elő.

A visszajelzések által kirajzolódtak a kutatói elvárások egy RAG-alapú szolgáltatással szemben. Fontos szempont volt a válasz strukturáltsága, pontossága, tartalmassága. A tesztelők pozitívan értékelték, ha a modell jelezte, hogy a források alapján nem tud teljeskörű választ adni a feltett kérdésre. Különösen fontos szempont volt még a forrásokhivatkozások pontossága, a forráskörnyezet láthatóvá tétele, és a teljes forrásanyag közvetlen elérhetősége a kutatók számára.

A fejlesztés tapasztalatai

A fejlesztés bebizonyította, hogy a ma elérhető és használható RAG technológia alkalmas arra, hogy hatékonyan segítse a levéltári dokumentumok kutatását. Közel 3000 oldalnyi dokumentumból másodpercek alatt nyerhetünk ki olyan információkat, amelyekhez a hagyományos (digitális) keresési technikákkal hosszadalmas keresgélésre és a talált szövegrészek értelmezésére, feldolgozására lenne szükség.

Mindemellett fontos, hogy a felhasználók ismerjék e technika korlátait: nem várható mindig tökéletes és hiánytalan válasz. Bizonyos típusú kérdések eleve fel se tehetők a RAG technológia jellemzői miatt: pl. Hányszor szólalt fel Gerő Ernő? Ez a technika nem képes arra, hogy a teljes tartalmat végig vizsgálva megszámlálja a felszólalásokat, vagy bármilyen más szöveg előfordulásait.

A RAG technológia széleskörű használatbavételéhez a rendelkezésre álló dokumentumok előfeldolgozása jelentős munkaigénnyel fog járni, ugyanis a különböző PDF fájlok szerkezete teljesen heterogén, azokat nem úgy állították elő, hogy erre a célra módosítás nélkül alkalmasak legyenek. A lábjegyzetek – ha vannak – elkülönítése külön nehézséget jelent.

Fontos, hogy az alkalmazásba épített prompt tartalmán nagyon sok múlik. Kis módosítás gyakran jelentős következményeket von maga után, újra kell tesztelni sok-sok kérdéssel. A fejlesztés jelentős részben a modellparaméterek és a prompt variálásában merül ki, ezek kipróbálása, tesztelése időigényes. A fejlesztési folyamat nem mindig lineáris, néha inkább a „Trial and error" jellemzés illik rá. Az egyik nyelvi modellnél bevált paraméterek, promptok nem biztos, hogy jók egy másik modellváltozathoz is. Új modellre áttéréskor újra kell kezdeni a tesztelést, módosítani a promptot és a modellparaméterek beállítását.

Az alábbiakban összefoglaljuk mindazokat a szempontokat, amelyeket figyelembe kell venni egy RAG rendszer tervezésekor és fejlesztésekor. Jó eredményt csak akkor várhatunk, ha az alábbi feltételek mindegyike teljesül:

- A dokumentumkorpusz legyen megfelelően előfeldolgozott és feldarabolt. A szövegdarabok mérete és az átfedés nagysága is fontos (a teljesen automatizált, kész előfeldolgozó megoldások nem mindig adnak jó eredményt).

- Magas dimenziószámú vektorok generálása egy magyar nyelvet jól kezelő modellel.
- Gyors működést biztosító, lehetőleg kombinált (szemantikus és szöveges) keresés implementálása a tulajdonnevek kiemelt kezelésével (NER).
- Magyarul is kifogástalanul teljesítő, jó minőségű nagy nyelvi modell használata.
- Prompt kialakítása az alábbi szempontok alapján:
  - a használt nagy nyelvi modell tulajdonságai,
  - a dokumentumok tartalma,
  - a RAG rendszer célja.
- A modellparaméterek finomhangolása, amely a prompt fejlesztésével, finomításával és rengeteg teszteléssel egyszerre végzendő.

Távlati célok

Az eddigi tapasztalatok és a kutatói visszajelzések alapján kirajzolódnak a fejlesztés további lépései. Elsőként a demóverzió élesítése a cél, egy jól definiált korszak iratainak közzétételével. Egyúttal folyamatosan újabb és újabb irategyütteseket teszünk a RAG-on keresztül kutathatóvá, hogy minél szélesebb körű tudásbázist alakítsunk ki. Jelen terveink alapján a minisztertanácsi iratokat középtávon az MTI[10] és az MSZMP[11] források követhetnék.

A különböző típusú források integrálása nemcsak a rendszer által feldolgozható információk mennyiségét növeli, hanem hozzájárul ahhoz is, hogy a felhasználók összetettebb történeti összefüggéseket tárhassanak fel. A forrásanyagok bővítésének egyik alapfeltétele a jó minőségű, OCR-rel[12] feldolgozott szövegállományok előállítása, hiszen a hibás vagy torzított szöveg rontja a visszakeresés pontosságát és a generált válaszok megbízhatóságát. Ehhez egyrészt szükség van a karakterfelismerési hibák csökkentésére, másrészt a szövegek strukturáltabb feldolgozására, például bekezdések, címsorok vagy egyéb tartalmi egységek pontosabb azonosítására.

Emellett fontos fejlesztési irány, hogy a rendszer által generált válaszokban szereplő hivatkozásokat közvetlenül az Adatbázisok Online[13] megfelelő rekordjaira mutató linkekkel kapcsoljuk össze. Ez nemcsak az átláthatóságot és az ellenőrizhetőséget növeli, hanem lehetőséget ad arra is, hogy a felhasználók a generált válaszokból kiindulva az eredeti forrásokban folytathassák a kutatást.

10 MTI (Magyar Távirati Iroda) forrásai, lásd: Adatbázisok Online: https://adatbazisokonline.mnl.gov.hu/adatbazis/magyar-tavirati-iroda. [Megtekintve: 2026.05.06.]

11 MSZMP (Magyar Szocialista Munkáspárt) jegyzőkönyvei.

12 Az OCR (Optikai Karakterfelismerés) technológia a képeken, beszkennelt dokumentumokon vagy PDF-fájlokban található gépelt, kézírott vagy nyomtatott szöveget géppel olvasható, szerkeszthető és kereshető szöveggé alakítja.

13 https://adatbazisokonline.mnl.gov.hu/

Irodalomjegyzék

[1] Király Péter: Adat a könyvtárban. Hagyomány és újítás a 21. századi könyvtárban. Erdélyi évszázadok. A kolozsvári magyar történeti intézet évkönyve III. Bolyai Társaság – Kolozsvár, Egyetemi Műhely kiadó, 2018, 61.

[2] Hajdu Krisztián: A mesterséges intelligencia szerepe a kutatástámogatásban. Különleges Bánásmód Interdiszciplináris folyóirat, 2025/11,15. DOI https://doi.org/10.18458/KB.2025.2.7

[3] Lewis, Patrick et.al.: Retrieval Augmented Generation for Knowledge-Intensive NLP Tasks, ArXiv, 2025, 2.

[4] Lin, Claire et.al.: A Preliminary Study of RAG for Taiwanese Historical Archives. Proceedings of the 37th Conference on Computational Linguistics and Speech Processing (ROCLING 2025), 2025, 45–62.

[5] Tran, The Trung et.al.: Retrieval Augmented Generation for Historical Newspapers. JCDL '24: Proceedings of the 24th ACM/IEEE Joint Conference on Digital Libraries, 2025, 1–5. DOI: https://doi.org/10.1145/3677389.3702542

[6] Kelly, Paul – Schild, Jonathan – Jafari, Amir: FolkRAG: a retrieval-augmented generation system for cultural heritage materials. Neural Computing and Applications, 37., 2025/24, 20281–20297. DOI: https://doi.org/10.1007/s00521-025-11455-4