





# Wikipédia Cikkek Kategorizálása Mistral 7B Segítségével

1<sup>st</sup> Gergelyi Laura   
Department of Software Engineering  
Eötvös Loránd University (ELTE)  
Budapest, Hungary  
laura.gergelyi@gmail.com

2<sup>nd</sup> Tóth Botond   
Information Systems  
Eötvös Loránd University (ELTE)  
Budapest, Hungary  
tothbotond00@gmail.com

3<sup>rd</sup> Béres Gábor Kristóf   
Department of Software Engineering  
Eötvös Loránd University (ELTE)  
Budapest, Hungary  
beresgabo2002@gmail.com

4<sup>th</sup> Attila Kiss   
Information Systems  
Eötvös Loránd University (ELTE)  
Budapest, Hungary  
kiss@inf.elte.hu

**Abstract**—A természetes nyelvfeldolgozás (NLP) egyik klasszikus feladata a szövegek automatikus kategorizálása. Jelen tanulmány célja a Mistral 7B nyílt forráskódú nagy nyelvi modell (LLM) alkalmazása egyszerű angol nyelvű Wikipédia-cikkek témakörök szerinti besorolására és összegzésére. A projekt során a Kaggle-ről származó Simple English Wikipedia adathalmazt használtuk, amelyet előzetesen megtisztítottunk és struktúrált formában feldolgoztunk. A Mistral 7B modellt lokálisan, API használata nélkül, kvantált GGUF formátumban futtattuk a llama.cpp eszköz segítségével. A kategorizálás során hat előre meghatározott témakörbe soroltuk a cikkeket (történelem, tudomány, művészet, irodalom, technológia, általános információ), és törekedtünk az egyszavas, egyértelmű válaszformátum biztosítására. A projekt során részletes tapasztalatokat szereztünk a modell működéséről, válaszainak megbízhatóságáról, valamint a hosszabb futtatás során fellépő kihívásokról. A kísérlet eredményei alapján megállapítható, hogy a Mistral 7B kvantált változata CPU környezetben is hatékonyan alkalmazható oktatási és kutatási célú szövegfeldolgozási feladatokra.

**Index Terms**—Large language model, Wikipedia, Mistral 7B, Llama

## I. BEVEZETÉS

Az elmúlt években a nagy nyelvi modellek (LLM-ek) – mint például az OpenAI GPT, a Google PaLM vagy a Meta által fejlesztett LLaMA és Mistral modellek – egyre nagyobb szerepet kaptak a természetes nyelvfeldolgozás (NLP) és a mesterséges intelligencia kutatásában. Ezek a modellek hatalmas paraméterszámmal és kiterjedt tanulási kapacitással rendelkeznek, amelyek lehetővé teszik számukra, hogy különféle nyelvi feladatokat oldjanak meg, beleértve a szövegértelmezést, szöveggenerálást, kategorizálást, kérdés-válasz rendszerek működtetését, illetve akár szöveges összefoglalók készítését is.

Jelen tanulmány egy konkrét alkalmazási eseten keresztül vizsgálja a Mistral 7B modell képességeit, amely egy nyílt forráskódú, 7 milliárd paraméteres nagy nyelvi modell. A célunk az volt, hogy a modell segítségével egyszerű angol nyelven írt Wikipédia-cikkeket automatikusan soroljunk előre

meghatározott kategóriákba, illetve tömör összefoglalókat generáljunk ezek tartalmáról.

A feladathoz a Simple English Wikipedia Plain Text elnevezésű adathalmazt használtuk fel, amelyet előzetes tisztítás és feldolgozás után lokálisan tápláltunk be a modellnek. A feldolgozás során kizárólag lokális eszközöket alkalmaztunk, az API-használat teljes kizárásával, a llama.cpp nyílt forrású CLI-eszköz segítségével.

A projekt során részletes tesztelést végeztünk a modell válaszainak megbízhatóságáról, a kategóriák szerinti besorolás pontosságáról, valamint az összefoglalók tömörségéről és koherenciájáról. Az eredmények azt mutatják, hogy a Mistral 7B kvantált verziója hatékonyan képes működni CPU-környezetben is, és alkalmas struktúrált nyelvi feladatok ellátására oktatási és kutatási környezetben egyaránt.

## II. MIK AZOK A NAGY NYELVI MODELLEK?

A nagyméretű nyelvi modellek (LLM-ek) olyan gépi tanulási modellek, amelyek a természetes nyelv megértésére és generálására specializálódtak. Ezek a modellek transzformátor architektúrára épülnek, és hatalmas adatbázisokon való tanulás révén képesek különböző nyelvi feladatok elvégzésére. Az LLM-ek kulcsfontosságú tulajdonságai közé tartozik a kontextuális árnyalatok megragadása, koherens szövegek generálása, és a különböző területekhez való alkalmazkodás specifikus feladatokra való tanítás nélkül. Az LLM-ek fejlődése a számítástechnikai teljesítmény növekedésének, a nagy mennyiségű szöveges adatok elérhetőségének és az innovatív tanítási módszereknek köszönhető.

Az LLM-ek architektúrája általában több rétegű neurális hálózatokat tartalmaz, amelyek a szövegadatokat egy sor transzformáción keresztül dolgozzák fel. Ezek a modellek képesek megérteni a kontextust, megjósolni a szekvencia következő szavait, és olyan szövegeket generálni, amelyek kontextuálisan relevánsak és koherensek. A modelleket változatos adatbázisokon pre-tréningelik, lehetővé téve

számukra a nyelv széles körű megértését, amelyet később finomhangolással alkalmaznak specifikus feladatokra.

#### A. A nagy nyelvi modellek evolúciója

A nagyméretű nyelvi modellek fejlődése több jelentős mérföldkő mentén követhető nyomon:

- **Korai modellek:** A korai nyelvi modellek viszonylag egyszerűek voltak, és specifikus feladatokra, mint például a szófajok felismerése vagy a név entitások felismerése, összpontosítottak. Ezeket a modelleket általában korlátozott adatbázisokon tanították, és nem voltak képesek különböző feladatok általánosítására.
- **Pre-tréningelt nyelvi modellek (PLM-ek):** Az olyan modellek megjelenése, mint a BERT (Bidirectional Encoder Representations from Transformers) és a GPT (Generative Pre-trained Transformer) jelentős változást hozott. Ezek a modellek önfelügyelt tanulást alkalmaztak, ahol nagy korpuszokon tréningeltek, hogy sokoldalú nyelvi reprezentációkat hozzanak létre, amelyeket később különböző leágazó feladatokra finomhangolhattak.
- **Nagy nyelvi modellek (LLM-ek):** Az olyan modellek bevezetése, mint a GPT-3, amely 175 milliárd paramétert tartalmaz, jelentős ugrást jelent a területen. Ezek a modellek képesek zero-shot tanulásra, ahol képesek feladatokat végrehajtani anélkül, hogy explicit feladat-specifikus tréninget igényelnének, ehelyett kiterjedt pre-tréningelt tudásukra támaszkodnak.

A fejlődés a feladat-specifikus modellektől a pre-tréningelt modellekig, és végül a nagy méretű LLM-ekig a hardver, az optimalizálási algoritmusok és az adatgyűjtési technikák fejlődésének köszönhető. A finomhangolás, a transzfer tanulás és az emberi visszajelzések integrálása tovább javították a modellek teljesítményét és alkalmazkodóképességét.

#### B. A nagy nyelvi modellek általános felépítése

Az LLM-ek általános architektúráját számos kulcsfontosságú komponens és technika jellemzi:

- **Tokenizáció:** Ez a folyamat a szöveget kisebb egységekre, úgynevezett tokenekre bontja, amelyek lehetnek karakterek, szavak vagy szóelemek. A tokenizáció lehetővé teszi a modell számára, hogy különböző szöveg-hosszúságokat és komplexitásokat kezeljen.
- **Pozicionális kódolás:** Mivel a transzformátorok párhuzamosan dolgozzák fel a bemeneti szekvenciákat, pozicionális kódolásokat adnak a tokenekhez, hogy megőrizzék a szekvencia sorrendjéről szóló információkat. Ez lehetővé teszi a modell számára, hogy megértse a tokenek helyzetét egymáshoz képest.
- **Figyelmi mechanizmusok:** A figyelmi mechanizmusok lehetővé teszik a modell számára, hogy a bemeneti szekvencia releváns részeire összpontosítson. Különböző figyelmi formák, beleértve az önfelügyelt, a keresztfigyelmet, a ritkított figyelmet és a flash figyelmet, javítják a modell képességét a szövegen belüli függőségek és kapcsolatok megragadására.

- **Aktivációs függvények:** Nemlineáris aktivációs függvények, mint például a ReLU (Rectified Linear Unit) és a GeLU (Gaussian Error Linear Unit), nemlineáris elemeket vezetnek be a modellbe, lehetővé téve, hogy komplex mintákat tanuljon az adatokban.
- **Réteg normalizálás:** Olyan technikák, mint a Layer-Norm és az RMSNorm, a rétegek bemeneteinek normalizálásával stabil és hatékony tréninget biztosítanak. Ezek a normalizálási módszerek segítenek enyhíteni a belső kovariancia eltolódásokkal kapcsolatos problémákat és javítják a konvergencia sebességét.
- **Elosztott tanulás:** Az LLM-ek hatalmas mérete miatt elengedhetetlen az elosztott tanulási technikák alkalmazása. Az adatparallelizmus, a tenzorparallelizmus és a pipeline-parallelizmus gyakran használatosak a számítási terhelés elosztására több GPU vagy gép között, lehetővé téve a nagy modellek hatékony tréningjét.
- **Pre-tréning és finomhangolás:** Az LLM-eket kezdetben kiterjedt korpuszokon pre-tréningelik önfelügyelt tanulási technikákkal. Ez a pre-tréning fázis lehetővé teszi a modellek számára, hogy széles körű nyelvi megértést alakítsanak ki. Ezt követően kisebb, feladat-specifikus adatbázisokon finomhangolják őket, hogy alkalmazkodjanak adott alkalmazásokhoz vagy feladatokhoz.

Különböző architektúrális variációk, mint például az encoder-decoder modellek, a kauzális dekóder modellek és a prefix dekóder modellek, különböző feladatokhoz és alkalmazásokhoz illeszkednek. Az encoder-decoder modelleket gyakran fordítási és összefoglalási feladatokhoz használják, míg a kauzális dekóder modellek a nyelvgenerálási feladatokhoz alkalmasak. A prefix dekóderek képesek kezelni azokat a feladatokat, amelyek egy adott prefix vagy kontextus alapján történő kondicionálást igényelnek.

#### C. Jelenlegi kiemelkedő nagy nyelvi modellek

A nagy nyelvi modellek (LLM-ek) terén számos kiemelkedő modell létezik, amelyeket különböző kutatóintézetek és technológiai vállalatok fejlesztettek ki. Az alábbiakban bemutatunk néhány jelentős LLM-et és azok főbb jellemzőit:

- **OpenAI GPT-4:** Az OpenAI GPT-4 a nagy nyelvi modellek egyik legismertebb és legteljesítményesebb képviselője. A modell fejlesztése során jelentősen növelték a paramétereinek számát, és fejlett tréning-módszereket alkalmaztak, amelyek lehetővé teszik a GPT-4 számára, hogy komplex nyelvi feladatokat is nagy pontossággal hajtson végre. A GPT-modellek rendszeresen kiemelkedő eredményeket érnek el a különböző nyelvi benchmarkokon és versenyeken.
- **Anthropic Claude 3:** Az Anthropic Claude 3 modellel – beleértve a Haiku, Sonnet és Opus változatokat – különböző képességeket és teljesítményszinteket kínál. Fejlesztésük során nagy hangsúlyt fektettek a biztonságra, robusztusságra és az értékalapú igazításra. Ezek a modellek képesek kreatív szövegek generálására, kódolási feladatok elvégzésére, valamint multimodális inputok, például ábrák, képek és diagramok értelmezésére is.

- **Cohere Command-nightly:** A Cohere Command-nightly modell különösen hatékony az utasításszerű promptok feldolgozásában. 52 milliárd paraméterrel rendelkezik, és széleskörű internetes szövegkorpuszokon lett betanítva. Kiemelkedően teljesít szöveggenerálási, összefoglalási és keresési feladatokban, és több mint 100 nyelvet támogat, így jól alkalmazható nemzetközi környezetben is.
- **Google Gemini:** A Google által fejlesztett Gemini modellcsalád, különösen a Gemini Ultra, kiemelkedő nyelvi és multimodális feldolgozási képességekkel rendelkezik. Ezek a modellek képesek nemcsak szöveg, hanem kép, hang és videó adatok értelmezésére is. Erősségük a komplex logikai következtetésekben és többféle méretben való elérhetőségükben rejlik (Ultra, Pro, Nano).
- **Meta AI LLaMA és LLaMA 2:** A Meta AI LLaMA modelljei, mint például a LLaMA-13B és LLaMA-65B, kiváló teljesítményt nyújtanak alacsonyabb számítási erőforrás-igény mellett. A továbbfejlesztett LLaMA 2 modellcsalád 7B, 13B és 70B paraméteres változatokban érhető el, és jól alkalmazható szöveggenerálásra, összefoglalásra, valamint emberi beszélgetések szimulációjára. Külön hangsúlyt kapott a modellek biztonságossága és etikus használhatósága.
- **Mistral 7B:** A Mistral 7B modell egy nyílt forráskódú, 7 milliárd paraméterrel rendelkező LLM, amelyet a Mistral AI fejlesztett. Bár méretében kisebb, mint a GPT-4 vagy Claude 3 modellek, hatékony működésével, alacsony erőforrásigényével és kvantált változatának CPU-kompatibilitásával kiváló választás oktatási, kutatási és kísérleti célokra. A modell jól alkalmazható különféle nyelvi feladatokra, beleértve a szövegkategorizálást, összegzést, valamint a párbeszédes és utasításszerű inputok kezelését is. A nyílt GGUF formátumú változata lehetővé teszi, hogy API-használat nélkül, teljesen lokálisan fussanak rajta kísérletek.

Ezek a modellek eltérő erősségekkel és specializációkkal rendelkeznek, így különböző alkalmazási területeken kínálnak megoldásokat a természetes nyelv feldolgozásától a tartalomgeneráláson át a komplex logikai következtetésekig.

### III. MISTRAL 7B NYELVI MODELL

A **Mistral 7B** egy nyílt forráskódú nagy nyelvi modell, amely különféle természetes nyelvfeldolgozási feladatokhoz használható. A hatékony futtatásához azonban megfelelő hardver- és szoftverkörnyezet szükséges. Ez a bekezdés két fő módszert mutat be a Mistral 7B telepítésére és futtatására: a hivatalos Python-alapú mistral-inference interfész használatát (ami CUDA-képes GPU-t igényel), valamint egy kvantált változat futtatását llama.cpp segítségével, amely CPU-n és alacsony VRAM-mal rendelkező GPU-kon is működik.

#### A. A Mistral 7B kvantálási lehetőségei

A Mistral 7B különböző kvantálási formátumokban érhető el, amelyek optimalizálják a modell memória- és számítási igényét:

- **Q4\_K\_M:** 4 bites kvantált verzió, amely jelentősen csökkenti a VRAM-igényt, miközben a teljesítmény elfogadható marad.
- **Q5\_K\_M és Q6\_K:** Finomabb kvantálási szintek, amelyek jobb válaszmínőséget nyújtanak, de némileg több erőforrást igényelnek.

A kvantált modellek a llama.cpp rendszerrel kompatibilisek, így futtathatók akár CPU-n vagy alacsony VRAM-mal rendelkező GPU-kon is.

#### B. Mistral modellváltozatok és alkalmazási területek

A Mistral 7B két fő változatban érhető el:

- **Mistral 7B Base:** Nyers, finomhangolatlan változat, amelyet különböző feladatokra lehet tovább finomítani.
- **Mistral 7B Instruct:** Utasításkövető, felhasználóbarát változat, amely ideális párbeszédes alkalmazásokra és interaktív rendszerekre.

Ezek a változatok különféle felhasználási esetekben alkalmazhatók, mint például:

- Szövegkategorizálás és osztályozás
- Összegzés és kivonatolás
- Természetes nyelvű kérdés-megértés
- Oktatási és kutatási célú demonstrációk

#### C. A modell architektúrája és működése

A Mistral 7B modell alapja a **Transformer** architektúra, de néhány optimalizálással:

- **Sliding Window Attention:** Ez a figyelmi mechanizmus lehetővé teszi a hosszabb kontextuskezelést alacsonyabb számítási költség mellett.
- **Grouped Query Attention (GQA):** Hatékonyabb lekérdezésfeldolgozás nagyobb méretű modellek esetén.
- **Position Encoding:** Különleges pozicionális kódolási eljárás, amely javítja a kontextus megőrzését hosszú szekvenciák esetén.

A Mistral 7B modellt pre-tréning során hatalmas, publikus szövegkorpuszon képezték, és finomhangolás után specifikus feladatokhoz adaptálható.

#### D. Benchmark eredmények és teljesítmény

Habár a hivatalos benchmark eredmények még korlátozottan elérhetők, a közösségi tesztek és összehasonlítások alapján a Mistral 7B versenyképes teljesítményt nyújt a LLaMA 2-7B és GPT-J modellekkel szemben is. A kvantált változatok különösen népszerűek kisebb rendszereken való futtatáshoz, mivel lehetővé teszik a hatékony NLP-feladatvégrehajtást akár GPU nélkül is.

A nagy nyelvi modellek teljesítményét különböző témák mentén szervezett benchmarkokon keresztül mérik. Ezek a tesztek különféle kognitív és nyelvi képességeket vizsgálnak, és gyakran *0-shot*, *3-shot* vagy *5-shot* beállításban kerülnek kiértékelésre, attól függően, hogy a modell kap-e példákat a kérdés megválaszolása előtt. Az alábbiakban a leggyakrabban használt kategóriákat és azok összetevőit mutatjuk be.

- **Commonsense Reasoning (Józan ész alapú következtetés):** 0-shot értékelés, amely a következő

benchmarkok átlagát veszi: Hellaswag, Winogrande, PIQA, SIQA, OpenbookQA, ARC-Easy, ARC-Challenge, CommonsenseQA.

Ezek a feladatok a hétköznapi logikát és következtetést mérik, például: Melyik megoldás működhet a valóságban?, vagy Ki a névmas hivatkozása a szöveg alapján?

- **World Knowledge (Lexikális és világtudás):** 5-shot értékelés, két fő benchmark alapján: NaturalQuestions és TriviaQA.

Ezek a feladatok lexikális, enciklopédikus tudást igényelnek, amelyeket gyakran kvízkérdések formájában tesztelnek.

- **Reading Comprehension (Szövegértés):** 0-shot átlag a BoolQ és QuAC benchmarkokból.

A feladat célja eldöntendő kérdések megválaszolása adott szövegrészek alapján, illetve párbeszédalapú kérdés-válasz megértése.

- **Math (Matematika):** Kombinált értékelés: 8-shot GSM8K (maj@8) és 4-shot MATH (maj@4).

Ezek a benchmarkok matematikai szöveges feladatok megoldását vizsgálják. Az *maj@k* metrika azt jelzi, hogy a modell egy adott számú ismételt válaszkísérletből hányszor ad helyes többségi választ.

- **Code (Kódolás):** Átlag: 0-shot HumanEval és 3-shot MBPP.

Ezek a benchmarkok azt mérik, hogy a modell képes-e helyesen értelmezni és generálni programkódot, illetve megoldani programozási feladatokat.

- **Aggregált értékelések:**

- 5-shot MMLU – több tantárgyon átívelő kérdések (közgazdaságtan, történelem, biológia stb.)
- 3-shot BBH (Big-Bench Hard) – nehéz nyelvi, logikai és numerikus problémák
- 3–5-shot AGI Eval – mesterséges általános intelligenciát célzó értékelés, angol feleletválasztós kérdésekkel

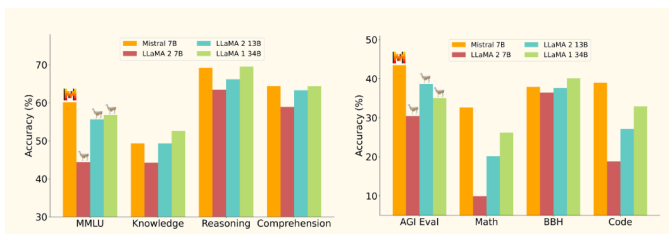


Fig. 1. Accuracy [1]

Model	Modality	MMLU	Hellaswag	WinoGrande	PIQA	Arc-e	Arc-c	NQ	TriviaQA	HumanEval	MBPP	MATH	GSM8K
LLaMA 2 7B	Pretrained	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	24.7%	63.8%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	Pretrained	55.6%	80.7%	72.9%	80.8%	75.2%	48.9%	29.0%	69.6%	18.9%	35.4%	6.0%	34.3%
Code LLaMA 7B	Finetuned	36.9%	62.9%	62.3%	72.8%	59.4%	34.5%	11.0%	34.9%	31.1%	52.5%	5.2%	20.8%
Mistral 7B	Pretrained	60.1%	81.3%	75.3%	83.0%	80.0%	55.5%	28.8%	69.9%	30.5%	47.5%	13.1%	52.1%

Fig. 2. Benchmark results [1]

## E. Telepítés

A hivatalos mistral-inference Python csomag használatához először telepíteni kell a megfelelő környezetet. Ajánlott Python 3.10 vagy 3.11 verzió használata, valamint egy CUDA-képes NVIDIA GPU legalább 16GB VRAM-mal. A telepítés során először létre kell hozni egy virtuális környezetet:

```
python -m venv mistral_env
source mistral_env/bin/activate
#Windows eseten: mistral_env/Scripts/activate
```

Ezután telepítsük a PyTorchot CUDA támogatással és a mistral-inference csomagot:

```
pip install torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cu121
pip install mistral-inference
```

A telepítés után a modellt az alábbi Python kóddal betölthető és futtatható:

```
import torch
from mistral_inference.transformer import
Transformer
from mistral_inference.generate import generate

model = Transformer.from_folder("./mistral-7B-
Instruct-v0.3/").to("cuda")
out_tokens, _ = generate(["Magyarazd el a gepi_
tanulast!"], model, max_tokens=64)
print(out_tokens)
```

A Mistral 7B futtatásának egyik leggyakoribb akadály a nem megfelelő GPU memória. A teljes modell futtatásához legalább 16GB VRAM szükséges, így sok fogyasztói GPU (pl. GTX 1650, RTX 3050) nem képes kezelni. Ha nincs telepítve a CUDA vagy a PyTorch rossz verziója van használatban, a következő hibák jelentkezhetnek:

```
AssertionError: Torch not compiled with CUDA enabled
```

Ha a GPU VRAM nem elegendő, az alábbi hibaüzenet jelenhet meg:

```
torch.OutOfMemoryError: CUDA out of memory
```

Ebben az esetben a legjobb alternatíva egy kvantált modell használata llama.cpp segítségével. A kvantált modellek csökkentett memóriamérettel rendelkeznek, például 4-bit pontossággal. A letöltéshez futtassuk az alábbi parancsot:

```
wget https://huggingface.co/TheBloke/Mistral-7B-v0
.1-GGUF/resolve/main/mistral-7b-v0.1.Q4_K_M.gguf
-P ./models/
```

Ezután llama.cpp klónozását és fordítását kell elvégezni:

```
git clone https://github.com/ggerranov/llama.cpp
cd llama.cpp
cmake -B build
cmake --build build --config Release
```

A sikeres fordítás után az alábbi paranccsal lehet elindítani az interaktív módot:

```
build/bin/main -m models/mistral-7b-v0.1.Q4_K_M.gguf
--interactive
```

Ez a módszer jelentősen kevesebb memóriát igényel, és nincs szükség sem PyTorchra, sem CUDA-ra. Azok számára, akik nagy teljesítményű NVIDIA GPU-val (16GB+ VRAM) rendelkeznek, a mistral-inference Python csomag a legjobb megoldás. Akik viszont alacsony VRAM-mal rendelkező GPU-val vagy CPU-val dolgoznak, egy kvantált modell futtatása llama.cpp segítségével egy életképes alternatíva, amely csökkenti a memóriahasználatot, miközben elfogadható teljesítményt nyújt.

#### IV. IV. ADATHALMAZ BEMUTATÁSA ÉS ELŐKÉSZÍTÉSE

Projektünk fókuszában a természetes nyelvfeldolgozási (NLP) feladatok állnak, és ehhez a célhoz választottuk a Kaggle platformon elérhető *Simple English Wikipedia Plain Text* [2] nevű adathalmazt. Ez a gyűjtemény a Simple English Wikipedia egyszerűsített nyelvezetén írt cikkek gazdag tárházát kínálja nyers, strukturált szöveges formátumban.

Ennek az adathalmaznak a kiválasztása tudatos döntés volt, mivel az egyszerű angol nyelvezet jelentősen hozzájárul a feldolgozás sikeréhez. A hagyományos Wikipédiával ellentétben a Simple English Wikipedia a közérthetőségre törekszik, kevesebb komplex mondat szerkezettel, egyszerűbb szókincsgel és világosabb fogalmazással. Ez a jellemző kritikus fontosságú az NLP feladatok szempontjából, mivel csökkenti a kétértelműséget, a szintaktikai bonyolultság okozta félreértéseket és megkönnyíti a szemantikai elemzést. Az egyszerű nyelvezet révén a gépi tanulási modellek, mint például a későbbi fázisban alkalmazott Mistral 7B, hatékonyabban képesek megragadni a szövegek lényegét és releváns mintázatokat tanulni.

Az adathalmaz eredeti formájában egyetlen nagy, plain text fájlként érhető el. Bár ez a formátum alapvetően strukturált, mivel a cikkek címei és tartalmi elkülönülnek, további tisztítási és rendezési lépéseket igényelt a közvetlen felhasználhatóság érdekében.

Az adathalmaz főbb jellemzői, amelyek ideálissá teszik a projekt számára, a következők:

- **Egyszerűsített Angol Nyelvezet:** A cikkek a Simple English Wikipedia stílusában íródtak, ami előnyös az NLP modellek számára a nyelvi komplexitás csökkentése révén.
- **Strukturált Szöveges Formátum:** Az adatok nyers szöveggént állnak rendelkezésre, ami egyszerűvé teszi a beolvasást és az alapvető szövegfeldolgozási műveleteket.
- **Cím és Tartalom Szétválasztása:** Az egyes Wikipédiás szócikkek címei és a hozzájuk tartozó tartalmak az eredeti fájlban is jól elkülöníthetők voltak, ami megkönnyítette a későbbi strukturálást.
- **Alkalmasság NLP Modellekhez:** A fent említett jellemzők együttesen biztosítják, hogy az adathalmaz könnyen előkészíthető és hatékonyan felhasználható különféle NLP feladatokhoz és modellek betanításához.

#### A. Tisztítási Lépések

Mielőtt az adathalmazt közvetlenül betápláltuk volna a kiválasztott gépi tanulási modellbe, elengedhetetlen volt egy átfogó tisztítási és előkészítési folyamat végrehajtása. Ennek célja az volt, hogy az adatok egységes, jól strukturált és a modell számára könnyen feldolgozható formátumba kerüljenek. A tisztítási lépések részletesen dokumentálva vannak egy Jupyter Notebookban, amely tartalmazza a végrehajtott Python kódokat és a lépésenkénti adatmanipulációt.

A legfontosabb tisztítási műveletek a következők voltak:

- **Szöveg Beolvasása és Előkészítése:** Az adathalmaz tartalmazó egyetlen nagy szövegfájl beolvasása volt az első lépés. Ezt követően a nyers szöveget előkészítettük a további feldolgozásra, ami magában foglalta a fájl feldarabolását az egyes, önálló Wikipédiás szócikkekre. Mivel a teljes adathalmaz egyetlen nagy fájlban volt, szükség volt egy mechanizmusra, amely felismeri az egyes cikkek határait.
- **Szócikkek Szétválasztása:** A cikkek elkülönítése kritikus lépés volt. Ezt úgy valósítottuk meg, hogy egy egyszerű, de hatékony heurisztikát alkalmaztunk: azonosítottuk azokat a sorokat, amelyek egyetlen szóból álltak és nagy kezdőbetűvel íródtak. Ezek a mintázatok jellegzetesen a Simple English Wikipedia cíkcímeit jelölték. Miután egy címet azonosítottunk, az azt követő összes sort (a következő címig vagy a fájl végéig) az adott cikk tartalmának tekintettük és így mentettük el a feldolgozás során. Ez a lépés eredményezte a címcímek és a hozzájuk tartozó tartalmak logikai párosítását.
- **Üres és Redundáns Sorok Eltávolítása:** A nyers szöveges fájlok gyakran tartalmaznak felesleges elemeket, mint például üres sorok vagy túlzottan sok whitespace karakter. Ezek a "zajok" negatívan befolyásolhatják az NLP modellek teljesítményét és növelhetik a feldolgozási időt. Ezért a tisztítási folyamat során szisztematikusan eltávolítottuk az összes üres sort és minimalizáltuk a felesleges whitespace-t a szövegben, egységesítve ezzel a bemeneti adat formátumát.
- **CSV Formátum Létrehozása:** Az előző lépések során szétválasztott és tisztított cíkcímeket és tartalmakat végül egy strukturált, táblázatos formába rendeztük. Minden sor egy-egy szócikknek felelt meg, jellemzően két oszloppal: az egyik a cikk címét, a másik pedig a teljes tartalmát tartalmazta. Ezt a strukturált adatot egy CSV (Comma Separated Values) fájlba mentettük el. A CSV formátum kiválasztása több szempontból is előnyös: széles körben elterjedt, könnyen olvasható ember és gép számára egyaránt, és számos adatelemző és gépi tanulási könyvtár (mint például a Pandas) natívan támogatja a beolvasását, megkönnyítve a további feldolgozást.

Az adattisztítás és előkészítés eredményeként egy jól strukturált, tiszta és egységes adathalmaz jött létre, amely tökéletesen alkalmas a Mistral 7B modell hatékony betanítására és későbbi tesztelésére. Az átláthatóság és reprodukálhatóság érdekében a teljes adatfeldolgozási pipeline, a

beolvasástól a CSV mentésig, részletesen dokumentálva van a mellékelt Jupyter Notebookban, lehetővé téve a folyamat teljes nyomon követését és szükség esetén módosítását.

## V. A PROBLÉMA MEGFOGALMAZÁSA

A projekt célja a Wikipédia szócikkek automatikus **kategorizálása** és **összegzése** a Mistral 7B nagy nyelvi modell segítségével. A célunk, hogy a cikkeket előre meghatározott témakörökbe soroljuk, valamint rövid összefoglalásokat generáljunk belőlük.

A kategorizálás során az alábbi főbb témakörökbe próbáljuk besorolni a cikkeket:

- Történelem (History)
- Tudomány (Science)
- Művészet (Art)
- Irodalom (Literature)
- Technológia (Technology)
- Általános információ (General Information)

A Wikipédia cikkek igen változatos témákat ölelnek fel, és egyes szócikkek több kategóriába is tartozhatnak. Ezért a nagy nyelvi modell használatával arra törekszünk, hogy a szövegek tartalmát megértve a lehető legpontosabb besorolást végezzük el.

A második cél az egyes cikkek **összegzésének generalálása**. Az eredeti Wikipédia cikkek gyakran hosszúak, és a felhasználók számára előnyös lehet egy tömörített, lényegre törő összegzés. Ehhez a Mistral 7B-t arra használjuk, hogy rövid, 3-5 mondatos összefoglalót készítsen az egyes cikkek főbb tartalmáról.

A kihívások között szerepel:

- A szövegek pontos kategorizálása különböző témák szerint
- Az összegzés generálásának minősége és tömörsége
- A modell hatékonysága és válaszideje nagy mennyiségű szöveg feldolgozásakor

A következő szakaszban bemutatjuk a módszertant és a megvalósítás lépéseit.

## VI. MISTRAL 7B KONFIGURÁCIÓJA ÉS KATEGORIZÁLÁSI METÓDUSA

A kiválasztott LLM, a Mistral 7B, megfelelő beállításokkal képes Wikipedia szócikkek kategorizálására. A következőkben bemutatjuk a konfigurációs lépéseket és a kód logikáját, amely biztosítja a megfelelő működést.

### A. A modell futtatása és konfigurációja

A Mistral 7B futtatásához a llama-cli interfészt használtuk. Mivel a modell **\*\*nem API-n keresztül működik\*\***, egy helyi GGUF formátumú modellt használunk. A legfontosabb beállítások:

- **Interaktív mód:** A `-cnv` flag engedélyezi az interaktív működést, így az LLM folyamatosan fogadja az új kérdéseket anélkül, hogy újra kellene indítani.
- **Kategorizálási utasítás:** A `-p` paraméter biztosítja, hogy a modell minden bemenetnél tudja, milyen válaszformátumot várunk el tőle.

- **Validációs mechanizmus:** Csak az előre meghatározott kategóriákat (History, Science, Art, Literature, Technology, General Information) fogadjuk el helyes válasznak.

### B. A kategorizálási algoritmus működése

A Python szkriptünk a következő lépésekből áll:

- 1) **A modell indítása:** Az LLM-et egyetlen példányban futtatjuk, amely folyamatosan fogadja a szövegeket és válaszokat ad.
- 2) **Cikkek beolvasása:** A Wikipedia szócikkeket egy CSV fájlból olvassuk be, és minden cikkhez generálunk egy bemeneti szöveget.
- 3) **Prompt küldése:** A szócikkek első 200 karakterét adjuk meg a modellnek, hogy a hosszú válaszok elkerülhetőek legyenek.
- 4) **A válaszok kiolvasása és tisztítása:**
  - Ha a válasz tartalmaz felesleges karaktereket, például a > jelet vagy az <|im\_end|> tokeneket, azokat eltávolítjuk.
  - A modell válaszát ellenőrizzük, és ha az nem felel meg az elvárt kategóriák egyikének, újra próbálkozunk.
  - Ha négy próbálkozás után sem kapunk megfelelő választ, akkor az adott szócikkhez az "Unknown" kategóriát rendeljük hozzá.
- 5) **Eredmények mentése:** Az összes feldolgozott szócikket és a hozzárendelt kategóriát CSV fájlba mentjük.

[illegible]

Fig. 3. Egy legenerált válasz egy minta prompt-al és az LLM válasza rá

[illegible]

Fig. 4. Prompt beadása a szükséges paraméterekkel

### C. Megoldás hatékonysága és megbízhatósága

A rendszer folyamatos futtatására azért van szükség, mert a `llama-cli` egy parancssori eszköz, amely nem rendelkezik klasszikus API támogatással. Az interaktív mód használatával jelentősen csökkentettük az egyes cikkek kategorizálásának időigényét, mivel nem kell minden egyes lekérdezés után újraindítani a modellt. Az `stdin` és `stdout` kezelésével a



Az optimalizáció során több problémát is kiküszöböltünk, például:

- A nem megfelelő válaszokat figyelmen kívül hagyjuk és újra kérdezzük.
- Megakadályoztuk a hosszú és ismétlődő generálásokat.
- Az LLM nem magyarázatokat ad, hanem egyértelműen egy szót választ a megfelelő kategóriák közül.

Ezzel a módszerrel az LLM sikeresen kategorizálja a Wikipedia szócikkeket, és a modell konfigurációja biztosítja a hatékony és pontos működést.

## VII. EREDMÉNYEK ÉS KIÉRTÉKELÉS

A modell futtatása során a Wikipédia cikkek egy részhalmazát sikeresen kategorizáltuk. Az eloszlás az alábbiak szerint alakult:

- **History:** 36.3%
- **General Information:** 32.0%
- **Science:** 20.8%
- **Art:** 9.4%
- **Literature:** 1.2%
- **Technology:** 0.3% (lényegében elhanyagolható)

Az eloszlásból jól látszik, hogy a történelmi és általános információs cikkek domináltak, míg az irodalom és technológia kevésbé volt jelen az adathalmazban. A következő ábra szemlélteti a kategóriák megoszlását:

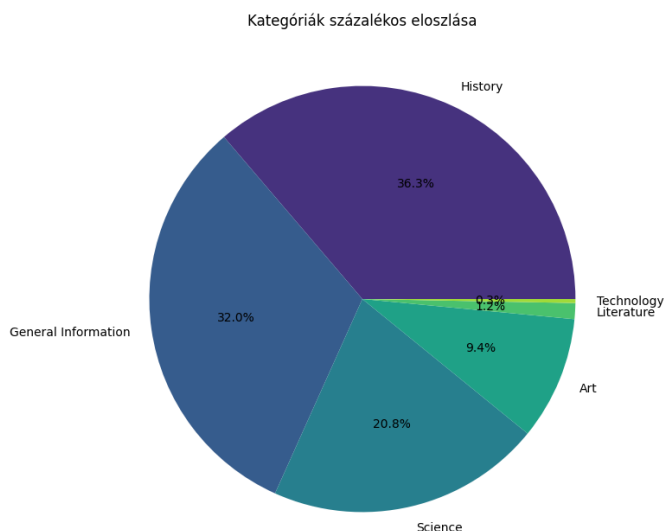


Fig. 5. A cikkek kategóriák szerinti megoszlása

A futtatás során több gyakorlati tapasztalatot is szereztünk a modell működéséről. A Mistral 7B kvantált változata jól teljesített, azonban hosszabb használat során megfigyeltük, hogy "elfárad", és hajlamos ismétlődő válaszokat adni. Ennek elkerülése érdekében bizonyos időközönként újra kellett indítani a modellt, hogy megőrizze a válaszainak frissességét és pontosságát.

A feladat sikeres megoldása után lehetőség nyílt további elemzésekre is. Például szófelhőket generáltunk a kategorizált cikkek tartalma alapján, melyek vizuálisan is jól megmutatták az adott kategóriákra jellemző szókinccset. Ez a jövőben kiindulópontként szolgálhat különféle gépi tanulási és adatbányászati technikákhoz, mint például K-means klaszterezés vagy SVM alapú szövegosztályozás.



Fig. 6. Szófelhő - History kategória

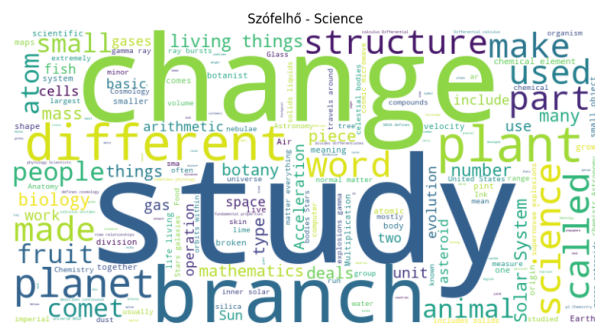


Fig. 7. Szófelhő - Science kategória

## VIII. ÖSSZEGZÉS ÉS JÖVŐBELI LEHETŐSÉGEK

A tapasztalatok alapján elmondható, hogy a Mistral 7B kvantált verziója CPU-környezetben is alkalmas lehet tanulmányi és kísérleti célú szövegfeldolgozási feladatokra.

A projekt sikeresen teljesítette fő célkitűzéseit. Sikerült egy nagy nyelvi modellt (LLM) lokálisan, külső API-któl függetlenül működésre bírni, és egy strukturált feladatot, a cikkek kategorizálását eredményesen végrehajtani a modell segítségével. A munka során értékes tapasztalatokat szereztünk a modell erőforrásigényéről, működéséről és a gyakorlati problémák (mint a válaszok pontossága vagy formázása) kezeléséről olyan egyszerű, de hatékony technikákkal, mint a reverse-prompt, input szűrés vagy többszöri próbálkozás.

A jövőben több irányban is érdemes lehet továbbfejleszteni a projektet a megszerzett alapokra építve:

- **Finomhangolt Kategória-besorolás:** Lehetőség van a jelenlegi kategóriák pontosítására, részletesebb alosztályok bevezetésére vagy teljesen új kategóriák definiálására a mélyebb elemzés érdekében.
- **Automatikus Összefoglalók Generálása:** A modell felhasználható lehet a kategorizált cikkek tartalmának tömör

összefoglalására, ami segíti a gyors áttekintést és a további feldolgozást.

- Adatbányászati Algoritmusok Tesztelése: A kategorizált és feldolgozott szövegeken különféle adatbányászati módszereket (pl. klaszterezés, hasonlóság-mérés) lehet kipróbálni a rejtett összefüggések és mintázatok feltárására.
- Modell Viselkedésének Mérése Hosszú Távon: Érdemes lehet vizsgálni a modell teljesítményének stabilitását és a válaszok konzisztenciáját ismételt vagy hosszabb ideig tartó használat során.

Összességében a tapasztalataink alapján a Mistral 7B modell kvantált verziója, még CPU környezetben futtatva is, ígéretes teljesítményt nyújt tanulmányi, kísérleti és kisebb léptékű szövegfeldolgozási feladatokhoz, ami hozzáférhetőbbé teszi a nagy nyelvi modellekkel való munkát.

#### REFERENCES

- [1] Mistral AI team. <https://mistral.ai/news/announcing-mistral-7b>. In *Performance in details*, 2023.
- [2] Plain text Wikipedia (SimpleEnglish). <https://www.kaggle.com/datasets/ffatty/plain-text-wikipedia-simpleenglish?resource=download>. 2024.