DOI: 10.31915/NWS.2025.6

TUDOMÁNYOSZTÁLYOZÁS AZ MTMT-BEN: MI ÉS MÁS LEHETŐSÉGEK

Micsik András, Tanácsi Roland, Kukucska Ákos, Nagy Kadosa
HUN-REN Számítástechnikai és Automatizálási Kutatóintézet

Absztrakt

A Magyar Tudományos Művek Tára (MTMT) 2.0 rendszere 2018-as indulása óta az OECD által gondozott Frascati-hierarchia szerinti kategorizálást támogatja a tudományterületi besorolásokhoz. A tudományosztályozás adatmező kitöltése a szerző vagy az adminisztrátor feladata, azonban ez a lépés gyakran kimarad, ami alacsony számú kategorizált forrásközleményhez vezet.

Kutatásunk során kimerítő vizsgálatokat végeztünk arra vonatkozóan, hogyan lehetne a hiányzó tudományterületi besorolásokat nagy számban pótolni. Különféle mesterséges-intelligencia- (MI) és statisztika alapú módszereket, valamint a már létező tudományterületi besorolásokat használtuk fel. Itt bemutatjuk az elvégzett kísérletek eredményeit.

Kulcsszavak: tudományosztályozási rendszerek, automatikus klasszifikáció, mesterséges intelligencia, BERT-modellek, nagy nyelvi modellek

Abstract

Scientific Classification in the Hungarian Science Bibliography (MTMT): AI and Other Possibilities

Since its launch in 2018, the Hungarian Science Bibliography (MTMT) has supported categorization according to the Frascati hierarchy maintained by the OECD for scientific field classifications. Filling out the scientific classification data field is the responsibility of the author or administrator; however this step is often omitted leading to a low number of categorized publications. In our research, we conducted exhaustive investigations nto how missing scientific field classifications should be supplemented in large numbers. We utilized various artificial intelligence- (AI) and statistics-based methods, as well as existing scientific field classifications. Here, we present the results of the experiments conducted.

Keywords: research classification, classification with AI, training dataset preparation, BERT, LLM

Bevezetés

A Magyar Tudományos Művek Tára (MTMT) keretein belül a kutatóknak lehetőségük van megadni közleményeik témaköri besorolását. Ezt az adatot többféleképpen is fel lehetne használni, de a legfontosabb talán az lenne, hogy az egyes tudományterületek újdonságait nyomon lehessen követni, a terület dinamikáját fel lehessen mérni.

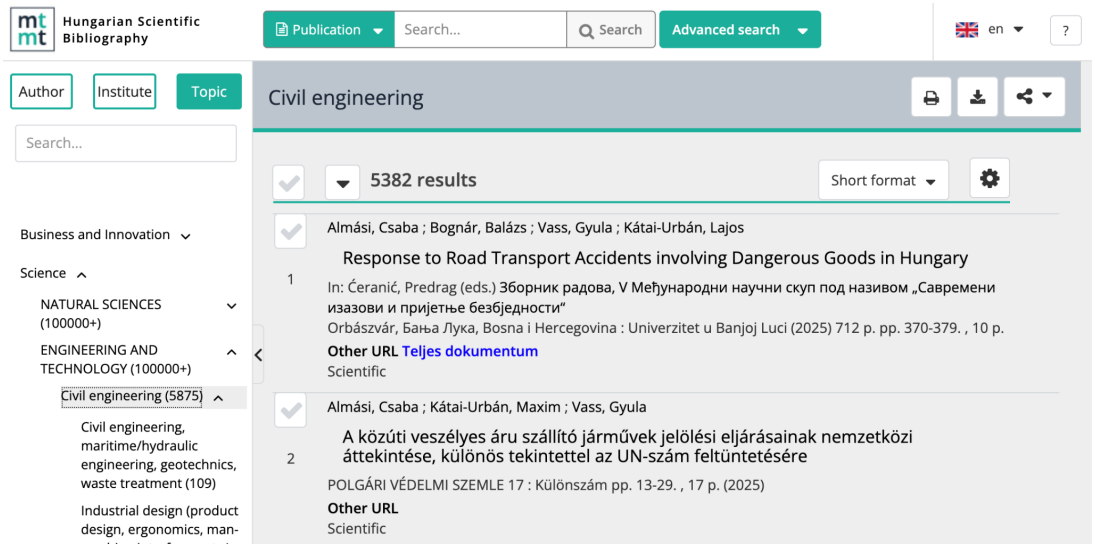
Sajnos a mai napig a forrásközlemények (tehát a magyar szerzőségű publikációk) 80%-a nincs besorolva tudományágakba, és az idéző közleményeknél még sokkal rosszabb ez az arány. Célul tűztük ki ezen besorolások tömeges és visszamenőleges pótlását, amelyhez számos mesterséges intelligencia nyújtotta módszert is megvizsgáltunk az alkalmazhatóság szempontjából, és ezen eredményeinkről számolunk be itt.

Az MTMT az OECD által gondozott Fields of Research and Development taxonómiát (másik nevén Frascati-hierarchiát) alkalmazza. Ez a taxonómia a tudományágakat 6 hierarchiaszinten több, mint 3000 különböző tudományos alterületre bontja, amelynek első két szintjét a lábjegyzetben megadott forrásban tekinthetjük át[1]. A Frascati kézikönyv[2] a kutatás-fejlesztési (K+F) statisztikák gyűjtésének és felhasználásának nemzetközileg elismert módszertana, amelynek első verzióját az olaszországi Frascatiban alakította ki egy szakértői csoport, innen ered tehát az elnevezés.

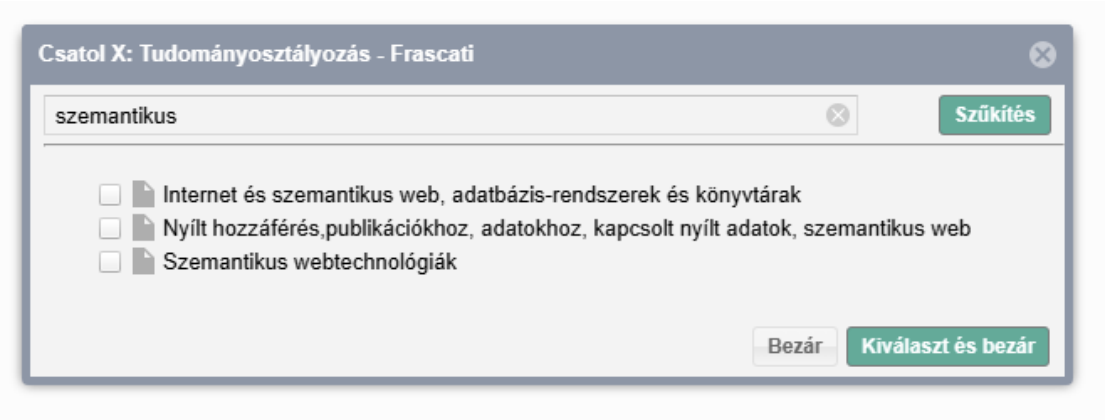
Az MTMT-ben rendelkezésre áll a teljes Frascati-rendszer angol és magyar nyelven is. A nyilvános felületen böngészhető a hierarchia (1. ábra), és az egyes területekhez rendelt közlemények számát is látjuk a neve mellett. A névre kattintva megtekinthetjük a közleménylistát a legfrissebbtől a régebbiek felé haladva. A szerkesztői felületen menüből indítható művelettel utólag, vagy már a felvitel közben is kiválaszthatjuk a megfelelő tudományterületeket, legfeljebb ötöt kijelölve. Akinek a teljes hierarchialista nehezen kezelhető, keresés alapján is meg tudja találni a témaköröket. A 2. ábra is szemlélteti azt a problémát, hogy a szakértőnek sem mindig egyszerű a megfelelő elemeket azonosítani a találatok között.

1 https://www.arrs.si/en/gradivo/sifranti/sif-frascati.asp

2 http://oe.cd/frascati



1. ábra. Tudományosztályozás szerinti böngészés az MTMT nyilvános felületén



2. ábra. Tudományosztályozás megadása az MTMT szerkesztői felületén

Mivel a szerzőket nem sikerült megfelelően ösztönözni a Frascati-besorolások kitöltésére, más megoldások után kellett néznünk. Az egyik lehetőség az lett volna, hogy más rendszerekből szerezzük be ezeket az információkat, azonban a Web of Science és a Scopus is más-más besorolási rendszereket használ, emellett ezek a besorolások sem megbízhatóak, mivel leginkább a publikáló folyóirat témakörein alapulnak és nem a konkrét cikk tartalmán.

A másik lehetőség egy mesterségesintelligencia-alapú automatizált klasszifikációs rendszer fejlesztése, amely a már meglévő osztályozások mintázatai alapján képes a hiányzó tudományosztályozásokat pótolni. Az ilyen klasszifikációs rendszerek fejlesztése az alábbi tipikus szakaszokból áll:

1. A megfelelő minőségű és mennyiségű tanító- és tesztadatok előállítása;
2. MI-modellek betanítása, azok minőségi teljesítményének mérése (ezt ismételve mindaddig, amíg az elvárásainknak megfelelő modelleket nem kapunk);
3. A legjobb modellek alapján a tömeges osztályozási módszer alkalmazásának megtervezése és végigvitele.

A továbbiakban az 1. és 2. lépés részletes ismertetésével foglalkozunk.

A tanítóadatok összeállítása

A gépi tanulási projektek sikeressége nagymértékben függ a tanítóadatok minőségétől, előkészítésétől és tisztításától (ezt a saját bőrünkön mi is megtapasztalhattuk). A kiindulási adatbázist az MTMT-ben már besorolt rekordok képezték, amelyek elemzése során több kihívással is szembesültünk. Az elsődleges probléma a Frascati-besorolások egyenlőtlen lefedettsége volt. Míg egyes tudományterületekhez több százezer közlemény tartozott, addig más, specifikusabb témakörökben alig vagy egyáltalán nem szerepelt rekord. Ezenkívül a meglévő besorolások jelentős része túlságosan általános, jellemzően 3. szintű kategóriákat tartalmazott (pl. Medical and Health Sciences).

Az alulreprezentált kategóriák adatainak pótlására a **Web of Science (WoS)** platformját használtuk fel. A WoS-kategóriák egyetlen hierarchia nélküli listában majdnem 300 elemből állnak. Ezek az elemek a Frascati-hierarchiában 4., 5. vagy 6. szinten álló taxonómia elemeknek feleltethetők meg. Hozzávetőleg 30 elem esetében nem találtunk megfelelő Frascati-besorolást (pl. development studies). Sok esetben csak megfogalmazásbeli eltérések voltak, mint pl. "Engineering, chemical" vs. "Chemical engineering'". Komolyabb eltérésre példa a "Media and communications" vs. "Communication".

A tanító- és tesztadatok végleges összeállítása a következőben leírt szempontok alapján történt. Elsősorban a Frascati-hierarchia 4. szintjén található 41 tárgyszót választottuk, mint lehetséges tudományosztályozási kimenetet, amelyeket a 6 fő, 3. szintű szülőtematika szerint csoportosítottunk (ezek: Bölcsészettudományok, Mezőgazdaság-tudományok, Műszaki és technológiai tudományok, Orvos- és egészségtudományok, Társadalomtudományok, Természettudományok). A finomabb, 5. és 6. szintű besorolásokat a 4. szintre vezettük vissza. Minden közleményhez egyetlen legjellemzőbb területet társítottunk. Ahol több terület is egyértelműen érintve volt, azokat a megfelelő multidiszciplináris kategóriákba tettük, vagy ha ilyen nem volt, akkor elhagytuk. Az egyetlen osztályt eredményező klasszifikációs feladatot **többosztályos osztályozásnak** (angolul **multiclass classification**) nevezzük, míg ahol több osztályt is lehet választani egyszerre, az a **többcímkés osztályozás** (multilabel classification). Ez utóbbi esetben biztosítani kell, hogy a kettő- és háromelemű osztálykombinációkból is elegendő számban legyenek tanítóadatsorok (az MTMT

rendszerében maximum hármat lehet hozzárendelni egy közleményhez), különben a kombinációs eseteket nem tudja a modell megtanulni.

A modellek tanító adataiként a közlemények címe és a hozzájuk tartozó kulcsszavak szolgáltak, mivel ezek általában rendelkezésre állnak. Olyan adatokra alapoztuk a tanítást, melyek általában szerepelnek az MTMT adatbázisában, és ez nem mondható el a kivonatokról (elérhetőség <25%). A folyóiratok, könyvek címének felhasználásával is próbálkoztunk, de azt tapasztaltuk, hogy túl sok a kivétel ezek esetében, amikor ezek a plusz adatok félre viszik a tanulást (pl. a folyóirat multidiszciplináris fókusza miatt).

Felismertük, hogy mind az MTMT, mind a WoS besorolási adatai tartalmaznak ún. zajt, azaz olyan eseteket, amelyek nehezítik vagy félrevezetik a tanítási folyamatot. Egyszerűbb esetben ide tartoznak a nagyon rövid címek, vagy ha hiányoznak a kulcsszavak. De ugyanígy félrevezető és zajnak tekintendő, ha túl sok kulcsszó van a cikkhez (pl. 15 vagy több). Erre az egyik legjobb példa az analízis kulcsszó használata olyan esetben, amikor bármilyen adatot elemezni kellett. Ilyenkor az MI előszeretettel tanulja meg, hogy a matematikához sorolja ezeket a publikációkat.

A szerző vagy adminisztrátor által megadott besorolások is lehetnek hibásak vagy félrevezetőek, egyrészt gondatlanságból, másrészt – amint egy korábbi példában mutattuk – a szakértőnek is nehéz lehet a megfelelő besorolás kiválasztása. Az ilyen esetek számát csökkenteni akartuk a tanítás hatékonysága érdekében, ezért létrehoztunk kétféle adatminőségi szűrőt. Az első Python-kód, amely egyszerű szabályok alapján szűri a tanító adatsorokat (pl. túl sok kulcsszó, túl rövid cím, nem angol nyelvű kulcsszavak stb.). A második szűrő egy nagy nyelvi modellt (LLM) alkalmaz, amely elbírálja, hogy egyértelmű-e a besorolás. Az LLM feladata volt annak ellenőrzése, hogy a bemeneti szöveg és a hozzárendelt címke tartalmilag konzisztens-e. A későbbiekben az eredményeket ismertető táblázatokban a tisztított elnevezést használjuk a fenti két módon szűrt tanító adatok esetében.

Automatikus osztályozási módszerek

A feladat egy szöveg osztályozási (text classification) probléma, amelyre számos megoldás létezik, és ezekből próbáltuk a főbb megoldástípusokból a legígéretesebbeket adaptálni a tudományos publikációk automatikus osztályozására. A projekt során alkalmazott módszereket három fő csoportba sorolhatjuk:

1. Célszoftver alkalmazása (pl. Annif) [1]
2. Sequence classification feladatra specializált Transformer-modellek finomhangolása (pl. SciBERT) [2]
3. Általános célú nagy nyelvi modellek (LLM) alkalmazása system prompt segítségével (pl. Llama3, Gemma) [3]

A vizsgált módszerek közös jellemzője, hogy mindegyik egy előtanított modellen alapszik. Az előtanítás során használt korpusz alapvetően befolyásolja, hogy a modell a kiválasztott célra megfelelően finomhangolható-e. Az általános szövegeken (pl. Wikipédia, hírek) tanított modellek (pl. BERT) szignifikánsan gyengébb eredményeket értek el, mert alig találkoztak még a tudományos nyelvezettel és terminológiával. Ez igaz abban az esetben is, hogyha újabb modellekről van szó, például a 2024 végén megjelent **ModernBert**-ről, amely a feladatunkat a legjobb esetben is csak **0,57 F1-értékkel** tudta megoldani [4].

Annif

Az Annif egy nyílt forráskódú szoftver, amelyet a Finn Nemzeti Könyvtár fejlesztett ki automatizált osztályozásra. Az eszköz MI- és gépi tanulási algoritmusok segítségével javasol tárgyszavakat vagy osztályozási azonosítót digitális dokumentumokhoz. Működése során a bemeneti szöveget elemzi, és egy előre definiált szótár (tezaurusz) alapján tesz javaslatot a besorolásokra. A szoftver nagy előnye a rugalmasság, mivel többféle algoritmust és modellt támogat, sőt ezeket kombinálni is tudja az ún. Ensemble-modellek segítségével.

SciBERT

A SciBERT a Google BERT architektúrájára épülő nagy nyelvi modell, amelyet kifejezetten tudományos szövegek feldolgozására specializáltak. A modellt egy hatalmas és sok tudományterületet lefedő publikáció-korpuszon tanították. Ennek köszönhetően a SciBERT lényegesen jobban kezeli a tudományos zsargont és a komplex szakterületi kontextusokat, mint az általános célú modellek. Fő alkalmazási területei a tudományos entitások felismerése, a szövegosztályozás, valamint a szakirodalmi összefoglalók készítése.

Általános célú előtanított LLM-ek

Itt egy általános célú nagy nyelvi modellt (pl. Llama3) egy specifikus **system prompt** segítségével utasítunk a tudományos közlemények osztályozására. A prompt egyértelműen definiálja a feladatot, a lehetséges kimeneti osztályokat (pl. „natural sciences", „engineering and technology" stb.) és a válasznak a formátumát. A módszer előnye, hogy nem igényel külön modelltanítást, így azonnal alkalmazható [3]. Hátránya viszont a jelentős hardverigény lokális futtatás esetén, illetve a drága előfizetés, ha üzleti megoldást alkalmazunk. További korlátot jelenthet, hogy a modell általános célú előtanítása során arányaiban jóval kevesebb tudományos szöveggel találkozott.

Kísérleteink

A klasszifikációs modellek teljesítményét leginkább a tanítóadatok minősége és mennyisége befolyásolja, különös tekintettel az adathalmaz zajszintjére és a kimeneti osztályok eloszlásának kiegyensúlyozottságára. A különböző módszerek objektív összehasonlítása érdekében az összes kísérletet egységesített és gondosan előkészített adathalmazokon végeztük el. A kísérleti adathalmazok minden osztályból 1026 példányt tartalmaztak a tanító- (train) és 116 példányt a kiértékelő (eval) részben. Összesen 7 különböző tanítóadat-halmazt hoztunk létre, amelyek a Frascati kézikönyvben meghatározott Fields of Science and Technology főcsoportjait, valamint a 6 főcsoport alcsoportjait fedték le[3].

Az **Annif** esetében a szoftver rendkívül érzékenynek bizonyult a tanítóadatok zajosságára és kiegyensúlyozatlanságára. Korábbi, szűretlen adatokon végzett kísérleteink során a modell hajlamos volt a leggyakoribb osztályt (Biological sciences) prediktálni, mivel a tanítóadat-halmazban ez az osztály dominált. A szoftver nem biztosít lehetőséget az osztályok súlyozására, amellyel a kiegyensúlyozatlanság hatása kompenzálható lenne. A legjobb eredményt egy **nn-ensemble** backend adta, amely egy **TF-IDF** és egy **omikuji-bonsai** modellt kombinált. Az 1. táblázatban látható eredmény egy hiperparaméter optimalizálást követően a 3. ábrán olvasható paraméterekkel lett előállítva.

A **SciBERT** finomhangolása a széles körben elterjedt Transformer architektúrának köszönhetően egyszerűen megvalósítható és testreszabható volt. Lehetőség nyílt többek között a loss függvény módosítására többcímkés osztályozáshoz (esetünkben BCEWithLogitsLoss), valamint az osztályok súlyozására az adathalmaz kiegyensúlyozatlanságának kezelésére. Emellett a kiértékelési folyamat is rugalmasan alakítható egyedi metrikák definiálásával. A loss függvény egy olyan matematikai függvény a gépi tanulásban, ami számszerűsíti, hogy a modell előrejelzése mennyire tér el a tényleges értéktől. Esetünkben a **Trainer** objektum leszármaztatásában meg tudtunk változtatni a tanítás során használt egyes műveleteket, például a **compute_loss** eljárásban az alapértelmezettől eltérő loss függvényt választottunk; multiclass esetben **CrossEntropyLoss**[4] és multilabel esetben pedig **BCEWithLogitsLoss**[5] függvényt.

3 https://en.wikipedia.org/wiki/Fields_of_Science_and_Technology

4 CrossEntropyLoss — PyTorch 2.8 documentation

5 BCEWithLogitsLoss — PyTorch 2.8 documentation

```
[tfidf-frascati]
name=frascati tfidf
language=en
backend=tfidf
analyzer=snowball(english)
vocab=frascati
access=private

[omikuji-bonsai-frascati]
name=frascati Omikuji Bonsai
language=en
backend=omikuji
analyzer=snowball(english)
vocab=frascati
cluster_balanced=False
cluster_k=100
max_depth=3
access=private

[nn-ensemble-frascati]
name=NN ensemble frascati tfidf-bonsai
language=en
backend=nn_ensemble
sources=tfidf-frascati:1,omikuji-bonsai-frascati:8
vocab=frascati
nodes=392
dropout_rate=0.4809760681960051
epochs=50
learning_rate=0.0010379317530691426
```

3. ábra. projects.cfg tartalma az Annif-hoz

A finomhangolás során a SciBERT-nél nem alkalmaztunk hiperparaméter keresést, mivel tapasztalatilag ezek csak egy nagyon kis mértékben tudták növelni a modell pontosságát, ellenben roppant időigényesek voltak. Ehelyett úgynevezett default értékeket alkalmaztunk a paraméterekre a következők szerint:

- epoch=100,
- batch_size=2,
- gradient_accumulation_steps=1,
- learning_rate=2e-05,
- weight_decay=0.01,
- warmup_ratio=0.1.

A tanítás során annak érdekében, hogy ne menjen feleslegesen 100 iterációt, úgynevezett Early stoppingot alkalmaztunk, amely azt figyelte, hogy az előző 5 tanítási cikluson belül az F1-érték alapján történik-e javulás, és ha nem, akkor leállította az egész tanítási folyamatot. Megfigyeltük, hogy általában 10–12 epoch után az esetek nagy részében a tanítási folyamat leállt, és szinte soha nem érte el a 20. iterációt. A tanítások a HUN-REN Cloud GPU-s virtuális gépén futottak (NVIDIA V100, 16GB VRAM-al). A VRAM használat főként

a *batch_size* értékétől függött: 2-es értéknél kb. 4 GB volt. Az 1. és 2. táblázatban szereplő tanítások átlagosan 30-45 perc alatt fejeződtek be, például a 2. táblázatban a természettudományi modell tanítása ~5000 tanító sorral kb. 45 percig tartott.

Az általános célú LLM-ek esetében a modell tanítás nélkül, úgynevezett **zero-shot** módon hajtja végre az osztályba sorolást [3]. Bár ezen modellek finomhangolása lehetséges, a rendkívül magas számítási erőforrásigény miatt ezt a megoldást elvetettük. A kísérleteket a HUN-REN **GenAi4Science**[6] szolgáltatáson keresztül, egy specifikus system prompt segítségével végeztük, az elterjedt prompt megfogalmazási alapelvek szerint[7]. Megfigyelhető volt, hogy a modellek, különösen a kisebb paraméterszámúak, rendkívül érzékenyek a prompt megfogalmazására.

A modellek teljesítményét, vagyis hogy mennyire sikeresen találják meg a megfelelő Frascati-besorolást, az általánosan elterjedt F1-metrikával[8] mértük. Kísérleteink főbb eredményeit az 1. és 2. táblázat foglalja össze. A főcsoport szintű osztályozás során (1. táblázat) a SciBERT-modell a tisztított adathalmazon érte el a legmagasabb, **0,97-es F1-értéket**. Nem sokkal maradt le az Annif és a Mistral, és itt kiemelendő, hogy a Mistral tanítás nélkül érte el a 0,90-es F1 értéket. A mérések során a többosztályos (multiclass) tanítások rendre jobb eredményeket hoztak, amely az 1. táblázatban is látszik, ahol a legjobb eseteket tüntettük fel.

Módszer	**Adatok**	**Legjobb modell**	**Legjobb F1**
Annif	Tisztítatlan, 3-as szint, multiclass	nn_ensemble(tfidf:1,omikuji:8)	0,81
Annif	Tisztított, 3-as szint, multiclass	nn_ensemble(tfidf:1,omikuji:8)	0,90
BERT	Tisztítatlan, 3-as szint, multiclass	SciBERT	0,88
BERT	Tisztított, 3-as szint, multiclass	SciBERT	0,97
LLM	Tisztítatlan, 3-as szint, multiclass	Llama3.3:70B	0,75
LLM	Tisztított, 3-as szint, multiclass	Mistral-Small3.1:24B	0,90

1. táblázat: 3-as szintű Frascati-besorolások klasszifikációs eredményei

A 4. szintű, részletesebb klasszifikációs eredmények (2. táblázat) szintén a tisztított, LLM-mel validált adathalmazok fölényét mutatják. A modellek teljesítménye minden vizsgált témakörben szignifikánsan javult a zajosnak feltételezett adatok kiszűrésével.

6 https://science-cloud.hu/genai4science

7 Build A LLM-Based Text Classifier| Prompt Engineering

8 Egyéni szövegbesorolási értékelési metrikák - Azure AI services | Microsoft Learn

Témakör	Legjobb modell	Tisztítatlan adatok, F1	Tisztított adatok, F1
természettudományi	SciBERT	0,85	0,92
mérnöki	SciBERT	0,59	0,82
orvosi	BiomedBERT	0,68	0,85
mezőgazdasági	SciBERT	0,80	0,91
társadalomtudományi	SciBERT	0,59	0,78
bölcsészettudományi	SciBERT	0,71	0,87

2. táblázat: 4-es szintű Frascati-besorolások BERT-alapú klasszifikációs eredményei

A második feladatnál kipróbáltuk az egyes tudományterületekre specializált BERT-modellek teljesítményét is, azt remélve, hogy így még jobb eredményeket érhetünk el. Az adott témakörre specializált BERT-modellek között leginkább orvosi témakörben voltak elérhető és működő modellek, mint például a Microsoft BiomedBERT (régebbi nevén PubMedBERT [5]) vagy a DMIS LAB BioBERT modellje. Ebben a vetélkedésben végül a BiomedBERT 0,5 %-kal megelőzte a SciBERT-et.

A mérnöki és a társadalomtudományi területeken elért alacsonyabb F1-értékek magyarázata, hogy ezekben a kategóriákban több mint 10 lehetséges alosztály található, ami növeli az osztályozási feladat komplexitását. Várhatóan ezen területek esetében a modellteljesítmény további, nagyobb mennyiségű tanítóadat bevonásával javítható lesz.

Összefoglalás

Az itt ismertetett kísérletekhez szükségünk volt tanítási és értékelési adatokra, különben az eredmények nem lettek volna mérhetőek, összehasonlíthatóak. Azt tapasztaltuk, hogy ezen adatok minőségének és mennyiségének biztosítása kulcsfontosságú a jó eredmények eléréséhez. Ezért automatikus módszerekhez folyamodtunk a tanítóadatok letöltése, válogatása, ellenőrzése során. Saját fejlesztésű szűrőkódok, valamint LLM igénybevételével biztosítottuk, hogy a tanítóadatok egyértelműek, jellemzőek és egyenletes eloszlásúak legyenek. A megfelelő osztályozási megoldás kiválasztása is nehéz volt, mivel a tudomány nyelvezete nagyon változatos, szókincse dinamikusan bővül. A felhasznált modellek „alaptudása" a döntő, esetünkben csak a tudományos szövegeken tanult modelleket tudtuk megfelelően finomhangolni. Mivel azt láttuk, hogy a BERT-jellegű modellek osztályozó képessége korlátosabb, ezért két lépésre osztottuk a feladatot: először a fő tudományági besorolást adja meg a modell, majd azon belüli alkategóriára tesz javaslatot a második modell. Így összesen hét modellt kellett betanítani, de ezeknek a kisebb modelleknek a betanítása és kezelése egyszerűbb és gyorsabb, mint egy nagy, monolitikus modell esetében.

Az itt ismertetett megoldást még nem tartjuk elég jónak ahhoz, hogy tényleges alkalmazásba kerüljön, akár csak olyan szinten, hogy a szerzők számára a felhasználói felületen javaslatot tegyen a tudományterület rovat kitöltésére, de tovább folytatjuk erőfeszítéseinket ennek a nem egyszerű feladatnak a jobb megoldására.

Köszönetnyilvánítás

Köszönetet mondunk a HUN-REN Cloud felhő és GenAI4Science szolgáltatásainak használatáért [6], amely segített nekünk a jelen cikkben közzétett eredmények elérésében.

Bibliográfia

[1] Golub, K., Suominen, O., Mohammed, A.T., Aagaard, H. and Osterman, O. (2024), „Automated Dewey Decimal Classification of Swedish library metadata using Annif software", Journal of Documentation, Vol. 80 No. 5, pp. 1057–1079. https://doi.org/10.1108/JD-01-2022-0026

[2] Beltagy, I., Lo, K. and Cohan, A., 2019. SciBERT: A pretrained language model for scientific text. arXiv preprint https://doi.org/10.48550/arXiv.1903.10676

[3] Wang, Z., Pang, Y., and Lin, Y., 2023. "Large Language Models Are Zero-Shot Text Classifiers", arXiv preprint, https://doi.org/10.48550/arXiv.2312.01044

[4] Philipp Schmid: Fine-tune classifier with ModernBERT in 2025, https://www.philschmid.de/fine-tune-modern-bert-in-2025

[5] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. ACM Trans. Comput. Healthcare 3, 1, Article 2 (January 2022), 23 pages. https://doi.org/10.1145/3458754

[6] Héder, M., Rigó, E., Medgyesi, D., Lovas, R., Tenczer, Sz., Török, F., Farkas, A., et al. "The Past, Present and Future of the ELKH Cloud." Információs Társadalom 22, no. 2 (August 31, 2022): 128. https://doi.org/10.22503/inftars.xxii.2022.2.8