

Mondatszintű határozatrész-címkézés magyar bírósági határozatokon

Csányi Gergely Márk¹, Üveges István^{1,4}, Lakatos Dorina¹, Dóra Ripszám², Kornélia Kozák³, Nagy Dániel¹, Vadász János Pál²

¹GriffSoft Zrt., 1041 Budapest

²Ludovika Nemzeti Közszerológálati Egyetem, Információs Társadalom Kutatóintézet, UNESCO Tanszék, 1083 Budapest

³Ludovika Nemzeti Közszerológálati Egyetem, Államtudományi és Nemzetközi Tanulmányok Kar, Európai Köz- és Magánjogi Tanszék, 1083 Budapest

⁴ELTE Társadalomtudományi Kutatóközpont, 1097 Budapest

{gergely.csanyi,istvan.ueges,dorina.lakatos,daniel.nagy}@griffsoft.hu
{ripszam.dora,kozak.kornelia,vadasz.pal}@uni-nke.hu

Kivonat A cikk egy magyar bírósági határozatokon működő, mondat-szintű határozatrész-címkésző (Rhetorical Role Labeling-RRL) rendszert mutat be. Az RRL feladata, hogy a bírósági határozatok minden mondatát a határozatban betöltött szerepe szerinti címkével lássa el (pl. tényállás, bírói érvelés, döntés, felek érvelése stb.), ezáltal támogatva többek között a szemantikus keresést, per kimenetelének predikcióját vagy az automatikus összefoglalást. A munkában szakértők által kézileg annotált korpuszon hasonlítottunk össze több architektúrát (BiLSTM, Attention és BiLSTM+Attention) és egy lineáris SVM referenciamodellt, különböző beágyazási stratégiákkal (huBERT CLS vs. Jina v3, late chunkinggal és anélkül). A címkészlet nyolc osztályból állt. Eredményeink szerint a legjobb teljesítményt a huBERT CLS beágyazásokkal táplált BiLSTM érte el, mind dokumentumszintű pontosságban, mind (súlyozott és makró) F1-ben, érdemben felülmúlva az SVM-et. Meglepő módon a late chunking nem javította, hanem rontotta a mondat-szintű RRL pontosságát, ami arra utal, hogy a túl tág dokumentumkontextus zajt vihet a mondatvektorokba. A rendszer már éles környezetben is működik: RAG-alapú jogi információkeresést támogat a magyar bírósági határozatokon.

Kulcsszavak: rhetorical role labeling, retorikaszerep-címkézés, határozatrész-címkézés, late chunking, mondat-szintű osztályozás

1. Bevezetés

A retorikaszerep-címkézés (Rhetorical Role Labeling-RRL) olyan NLP feladat, amelyben a dokumentum egyes részeit (jelen esetben minden mondatát) a szövegben betöltött szerepe szerint osztályozzuk. A jogi területen azért érdekes probléma, mert a szerepek szerint felbontott határozatok jól hasznosíthatók több egyéb feladat esetében is. Ha képesek vagyunk felismerni egy szövegből, hogy mi az a szövegrész ami a tényállásról, vagy a bírói érvelésről szól, akkor ezeket a szövegrészeket felhasználhatjuk pl. jogeset predikcióhoz (feltételezve persze, hogy

hasonló bírósági ügyek hasonló ítélettel végződnek), de szemantikus kereséshez is, pl. a tényállásokra szűrve gyorsabban található hasonló ügyek, vagy kivonatok minőségét lehet javítani azzal, hogy csak a releváns szerkezeti egységeket használjuk fel a kivonatoláshoz. Tudomásunk szerint magyar nyelvre még nem készült hasonló megoldás, nemzetközi szinten pedig a legtöbb hasonló munka angol nyelvre készült (Bhattacharya és mtsai, 2019; Malik és mtsai, 2021; Bambroo és mtsai, 2025), pár nem angol mellett (Marino és mtsai, 2023; Aragy és mtsai, 2021). Ebben a munkában egy magyar bírósági döntéseken alkalmazható, élesben is működő RRL-osztályozót mutatunk be. Több architektúrát hasonlítottunk össze (BiLSTM és Attention-alapú modellek, lineáris SVM), és megvizsgáljuk a vektorizálás „late chunking” technikájának hatását is, amely a széles kontextusablakú beágyazási modellek segítségével képes a szövegrészeket átívelően kontextust biztosítani. A modelleket jogi szakértők által kézzel annotált, mondat szintű korpuszon tanítottuk és teszteltük.

2. Kapcsolódó irodalom

Bhattacharya és mtsai (2019) az indiai Legfelsőbb Bíróság öt jogterületről származó 50 ítéletén (összesen 9 380 mondat), mondat szintű RRL-feladaton dolgoztak 7 címkével. BiLSTM és BiLSTM-CRF modelleket vizsgáltak, a mondatok sorrendiségét a hierarchia és a CRF-tranzíciók révén hasznosítva. A mondatvektorokat `sent2vec`-ből származó, nagy jogi korpuszon előtanított beágyazásokkal állították elő.

Aragy és mtsai (2021) 70 darab portugál nyelvű polgári jogi beadványt, tehát nem bírósági határozatot címkézett fel mondat szinten. A mondatokat a BERT CLS tokenjével klasszifikálták, finomhangolva egy klasszifikációs réteget illetve a BERT egyes rétegeit is, tehát nem vették figyelembe a mondatok sorrendjéből származó plusz információt.

Malik és mtsai (2021) 100 angol nyelvű indiai ítéletből (50 versenyjogi, 50 adóügyi) álló, szakértőkkel annotált kb. 21 ezer mondatos korpuszt hoztak létre 7 retorikai címkével, mondat szinten, kifejezetten RRL-feladatra. A szerzők többfeladatos (Multi-Task Learning-MTL) modellt javasolnak, ahol a címkézéshez egy címkeváltozás predikció (label shift prediction) segédfeladat is társult a CRF mellett. Ez a megközelítés érdemben felülmúlta a mondat sorrendjétől független megoldásokat, bizonyítva a sorrendből származó extra információ fontosságát.

Marino és mtsai (2023) két korpuszon vizsgálták: egy kb. 1 500 olasz ítéletből álló ITA-RhetRoles korpuszon (5 címke) és a 275 dokumentumból álló BUILD angol korpuszon (13 címke), 96 ezer illetve kb 32 ezer mondattal. A szerzők hierarchikus modellt vizsgáltak (LEGAL-ToBERT: LEGAL-BERT fölé épített transzformer), amely a mondatok sorrendiségét és kölcsönös viszonyait is kiaknázza (pozicionális kódolással és mondat szintű encoderrel), és mindkét nyelven felülmúlta a sima LEGAL-BERT referenciamodellt.

Bambroo és mtsai (2025) mondat szintű RRL-t végeztek 7 címkével az indiai DIN (150 ítélet, 31 ezer mondat) és az angliai DUK (50 ítélet, 18 ezer mon-

dat) korpuszon. A javasolt MARRO modell BiLSTM-CRF-re épül multi-headed self-attentionnel, sent2vec vagy legal-bert-small beágyazásokkal és multi-task tanulással és label-shift segédfeladattal a modell a mondatok sorrendiségéből és a dokumentumon belüli távolabbi összefüggésekből is profitál.

3. Adathalmaz

3.1. Általános jellemzők

A célunk az volt, hogy az összes jelenleg nyilvánosan elérhető anonimizált bírósági határozatra elvégezzük az RRL címkézést, amely körülbelül 235 ezer dokumentumot jelent. A határozatok felépítése általában meglehetősen hasonló: a két fél, a bíróság, az előző bíróságok és az ügyek száma, valamint a tárgy információival kezdődnek, majd egy rövid bekezdés jön a döntésről. Ezt követi általában a tényállás és az előző bíróságok ítéleteinek ismertetése, majd a felek érvelései, újra a döntés, végül pedig a bíróság részletes érvelése a döntéssel kapcsolatban. A magyar bíróságok 2016 előtt másképp strukturálták ezeket a dokumentumokat, nem adtak egyértelmű fejezetcímeket a dokumentumokban, 2016 után pedig csak a Kúria volt köteles ilyen leíró fejezetcímeket adni, amit lassan az alacsonyabb szintű bíróságok is követtek. Mára már a fejezetcímeket tartalmazó megoldás vált az általánosabb szerkesztési móddá, amely egy neurális modell számára könnyen megtanulható címkézést tesz lehetővé. A továbbiakban a fejezetcímeket nem tartalmazó dokumentumokat a **régi típusúnak**, az ezeket tartalmazókat pedig **új típusú** dokumentumnak nevezzük.

A tanító, validációs és teszhalmazok jogterület szerinti eloszlását az 1. táblázat mutatja be. Eltérő volt az eloszlása a tanító és a teszteléshez használt halmazoknak. A tanítóhalmazban a jogterületek szinte egyenlően oszlottak el, a katonai büntetőügyeket büntetőügyként számolva, míg a teszhalmaz az egész, kb. 235 ezres korpusz eloszlását követte, hogy a tényleges pontosságot lehessen becsülni. A teljes korpuszban az új-régi típusú dokumentumok közti arány kb. 1:9-hez, mely fokozatosan egyenlítődik ki, mert az újabb dokumentumok már jellemzően új típusúak. A tanítóhalmaz minden jogterületen közel egyenlő számú régi és új típusú dokumentumot tartalmazott, kivéve a közigazgatási és katonai büntetőjogi eseteket. A régi típusú dokumentumok különösen fontosak voltak az értékelés szempontjából, mert az új típusú dokumentumokkal ellentétben nem tartalmaztak könnyen azonosítható fejezetcímeket, amelyek leegyszerűsíthették volna a címkézési feladatot, illetve a jelenlegi korpusz nagy többségben ilyen dokumentumokból áll. A dokumentumokat összesen hat jogi szakértő címkézte. Az annotátorok iránymutatásként megkapták a rendelkezésre álló címkék listáját (lásd 3.3. szakasz), amelyek egy jogász számára érthetőek, és azt az utasítást, hogy mondatonként csak egy címkét adjanak, több lehetséges címke esetén a legmegfelelőbbet kiválasztva. Végül egy 299 dokumentumból álló tanító+validálási és egy 120 dokumentumból álló teszhalmazt hoztunk létre.

Annotátorok közti egyetértést 10 dokumentumon (összesen 2734 mondat) számítottunk két annotátor bevonásával. Metrikaként az átlagos egyezést illetve

a Krippendorff alfa (Krippendorff, 2011) értéket választottuk. A részletes eredményeket az A. Függelék 5. és 6. táblázatai mutatják be. A nyolc címkéből hat jó egyezést mutatott ($\alpha > 0,67$) a Tényállás (0,3899) és a Bíróság döntése (0,4243) kategóriák szerepeltek rosszabbul.

1. táblázat. A tanító, validációs és teszhalmaz jogterületenkénti eloszlása

Jogterület	Tanító+validációs halmaz			Teszhalmaz			Arány [%]	
	Régi típus	Új típus	Össz.	Régi típus	Új típus	Össz.	Teszt	Teljes korpusz
Büntető	27	26	53	13	3	16	13,33	15,83
Gazdasági	28	28	56	12	3	15	12,50	12,41
Katonai büntető	5	0	5	3	0	3	2,50	2,01
Közigazgatási	49	24	73	21	3	24	20,00	20,19
Munkaügyi	26	26	52	10	4	14	11,67	8,34
Polgári	30	30	60	42	6	48	40,00	41,22
Összes	165	134	299	101	19	120	100	100

3.2. Mondatra bontás

A jogi dokumentumokban a szöveg mondatokra bontása kihívást jelent, mivel a jogi dokumentumok sokkal több pont karaktert tartalmaznak, mint más területek korpuszai. Ezek közé tartoznak a különböző ítélkezési gyakorlatra, ügyszámra vagy jogszabályra való hivatkozások (pl. II. Pfv.35.125/2010/4, 32/2008. (VII. 19.) IM rendelet 8. § stb.), valamint az anonimizálás eredményeként monogramok és három pont is gyakran előfordul. A mondatokat a Csányi és mtsai (2024) című cikkben leírt, kifejezetten bírósági dokumentumokhoz módosított heurisztikus szegmentálóval bontottuk szét. Ez a mondatszegmentáló nagyon jó eredményeket ért el jogi dokumentumok esetében, jobb teljesítményt nyújtott a HuSpaCy magyar változatában (Orosz és mtsai, 2023) található transzformer-alapú megoldáshoz képest, miközben lényegesen gyorsabb is.

3.3. Címkekészlet

A mondatokat az alábbi címkék egyikével címkéztük:

- **Tényállás:** minden mondat, amely leírja, hogy miről szól a jogvita.
- **Perelőzmény:** mivel a bírósági határozatok minden szintjét vizsgáltuk (Kúria, Törvényszékek, Ítéltáblák, Közigazgatási Bíróságok stb.), a dokumentum tartalmazhatnak információkat korábbi döntésekről is.
- **Felek érvelése:** a felek érveiről és kérdéseiről szóló mondatok.
- **Bíróság döntése:** Az a jellemzően egy-két mondat, ami a döntés maga, pl. A bíróság a felperes keresetét elutasítja.
- **Bírói érvelés:** a bírói érvelés a döntés jogi alapját illetően.
- **Perköltség:** mondatok arról, hogy az ítélet alapján melyik félnek mennyit kell fizetnie perköltségként.

- **Rendelkező rész:** az ítélet gyakorlati következményeit leíró mondatok, például hogy az alperesnek X összeget kell fizetnie kártérítésként a felperes részére.
- **Egyéb:** egyik fenti kategóriába sem tartozó részek, például aláírások, a dokumentum fejléce, keltezés, fejezetcímek stb.

A címkék eloszlását, a mondatszámot, arányukat az adatbázisban, valamint a mondatonkénti tokenek számát a 2. táblázat mutatja be. A tokenek számát a `jinaai/jina-embeddings-v3` huggingface modell segítségével számítottuk ki.

2. táblázat. Címkék eloszlása, tokenátlagok mondatonként

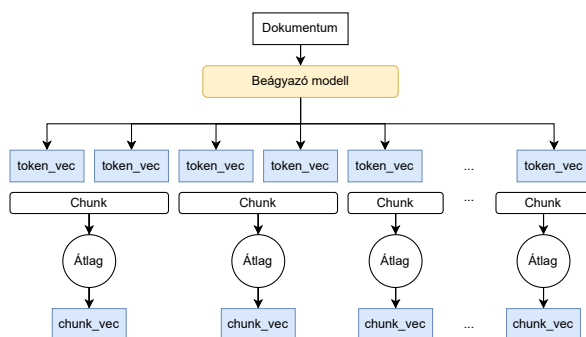
Label	Tanító és validációs adathalmaz			Teszthalmaz		
	Mondatszám	Arány	Token/mondat	Mondatszám	Arány	Token/mondat
Bíróság döntése	1657	0,0492	40,15	710	0,0402	47,70
Bírói érvelés	9864	0,2931	57,16	5899	0,3340	54,11
Felek érvelése	6639	0,1973	52,56	3198	0,1811	49,71
Egyéb	5518	0,1640	11,38	1380	0,0781	17,14
Perelőzmény	3735	0,1110	57,44	1653	0,0936	56,13
Perköltség	967	0,0287	57,08	450	0,0255	59,14
Rendelkező rész	441	0,0131	62,93	352	0,0199	55,45
Tényállás	4832	0,1436	49,32	4020	0,2276	49,52

4. Beágyazások

4.1. Jina beágyazások late chunkinggal és anélkül

A beágyazási modellek egy adott szöveget egy vektortérbe képeznek le. A korszerű, transzformer-alapú beágyazási megoldások azonban egy fix kontextusablakkal rendelkeznek. Hosszabb szövegek esetén gyakran szükséges a bemenetet kisebb darabokra felosztani, hogy mindegyik beférjen a modell kontextusablakába. Ezeknek a szövegrészeknek az egymástól függetlenül történő beágyazása azonban csökkenti a rendelkezésre álló dokumentumszintű kontextust.

Ennek kezelésére Günther és mtsai (2024) a JinaAI-tól egy kiváló, mégis egyszerű ötlettel állt elő, melyet late chunkingnak (utólagos felbontásnak) neveztek el. A koncepciót az 1. ábra szemlélteti. A late chunking során a teljes szöveget átadjuk a beágyazó modellnek, és kiszámításra kerülnek a token szintű beágyazások. Ezt követően történik csak a szöveg felbontása, vagyis az egyes chunkok kialakítása, mely során chunkvektorokat a chunkok tokenvektorainak átlagával képzünk. Így biztosítható, hogy az egyes szövegrészek közötti kontextus a beágyazásokban is megjelenjen. Ennek kiszámításához a beágyazó modellnek tokenszintű beágyazásokat is vissza kell adnia, és viszonylag nagyobb tokenablakkal kell rendelkeznie.



1. ábra. Late chunking

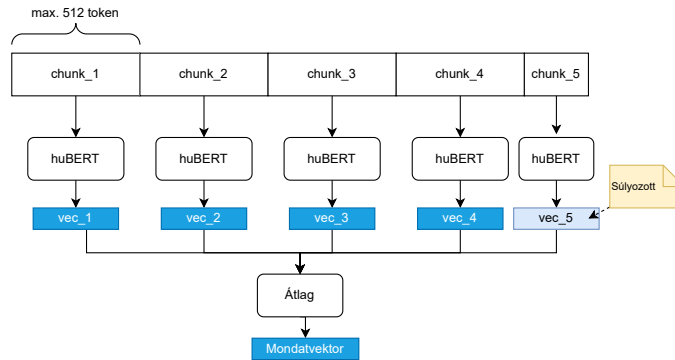
Több olyan beágyazási modell is van, amely akár 8192 tokent is képes lefedni, ám jellemzően ezek nem adnak vissza tokenszintű beágyazásokat, csak egy beágyazást a beadott szövegre, tehát ezekkel nem lehetséges a late chunking. Ilyen például az OpenAI `text-embedding-3-large` modellje, a Gemini `gemini-embedding-001` modellje. Tokenszintű beágyazásokat is visszaad például a Pekingi Mesterséges Intelligencia Akadémia BGE-M3 (Chen és mtsai, 2024) modellje, valamint a JinaAI `jina-embeddings-v3` (Sturua és mtsai, 2024) és `jina-embeddings-v4` modelljei (Günther és mtsai, 2025), valamint a Stella V5 modellje (Merola és Singh, 2025). Ebben a cikkben a Jina V3-as modelljét használtuk, a natív API segítségével, amelyben a late chunking beállítás egy egyszerű paraméterként elérhető.

4.2. BERT CLS

Második beágyazási modellként a huBERT (SZTAKI-HLT/hubert-base-cc) modell (Nemeskey, 2021) CLS token reprezentációját használtuk. Az előtanítás kizárólag magyar adatokon, többek között jogi dokumentumokon történt, ez némi előnyt jelentett feladatunkhoz, azonban a modellt nem finomhangoltuk. A modell maximális kontextusablaka 512 token, és 768 dimenziós CLS beágyazásokat hoz létre. A kontextusablaknál hosszabb mondatok kezelését a 2. ábra mutatja be. Ezeket a mondatokat maximum 512 token széles darabokra osztottuk, a szöveget csak a szavak határain bontva, a darabolt részek között átfedés nélkül. Minden egyes szövegrészhez kiszámítottuk a BERT CLS vektort. A kontextusablaknál hosszabb mondat beágyazása a felosztott adatok beágyazásainak átlaga volt, kivéve az utolsó darabot, ahol a vektort a tokenek számának és a kontextusablak arányával szoroztuk be, hasonlóan a Csányi és mtsai (2025) cikkhez.

4.3. Pozíciós jellemző

Ha a mondatokat úgy vektorizáljuk, hogy nem alkalmazzuk a late chunking módszerét, akkor a mondat helyzete a dokumentumban a kontextussal együttesen



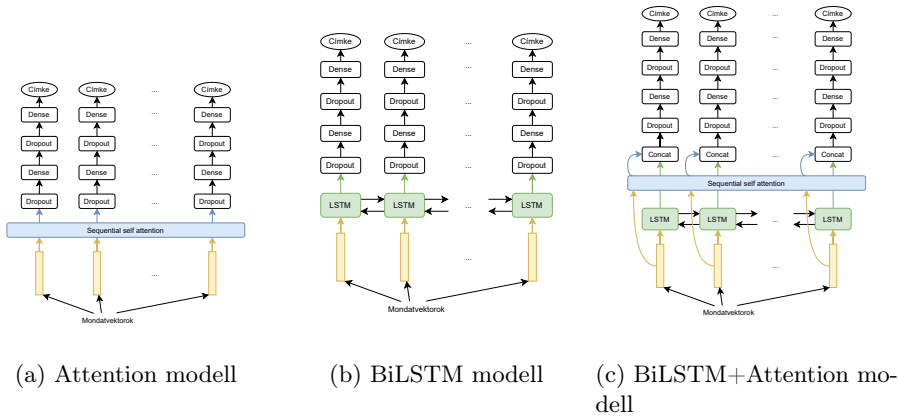
2. ábra. Kontextusablaknál hosszabb mondatok vektorizálása BERT CLS módszerrel

elveszik. Ezért kiszámítottuk a pozíciós jellemzőt, amely az adott mondat relatív helyzete a dokumentumban. Ez egy 0 és 1 közötti szám, amelyet plusz egy helyiértékként hozzáadtuk a már rendelkezésre álló beágyazásokhoz.

5. Osztályozási modellek

Az RRL egy szekvenciális osztályozási feladat, ahol a mondatok jelentése nem független egymástól, tehát a címkék sem függetlenek, így a mondat szövegben elfoglalt helye fontos információ az osztályozás során. Ennek igazolására kipróbáltuk referenciamodellként a lineáris SVM-et, amely nem képes kihasználni ezt az információt, valamint a szekvenciális címkézésre alkalmas neurális modelleket: BiLSTM (Hochreiter és Schmidhuber, 1997), Attention (Vaswani és mtsai, 2017) és BiLSTM+Attention hálókat. Az egyes architektúrák felépítése a 3. ábrán látható.

Minden architektúrában minden mondatot először beágyaztunk, majd ezek bekerültek az adott szekvenciális neurális modellbe. Minden szekvenciális kimeneti vektor egymásra helyezett Dense és Dropout rétegekbe, valamint egy mondatonkénti softmaxba tápláltunk be, nyolc neuront használva osztályozási réteggként. Az Attention architektúra (3a. ábra) self-attention-nel a mondat-sor felett hoz létre kontextusérzékeny reprezentációt. A BiLSTM (3b. ábra) a kétirányú mondatkörnyezettel ad kontextust, azonban hosszabb szekvenciákra nem működik nagyon jól. A BiLSTM+Attention (3c. ábra) pedig mindkét módszer erősségeit egyesíti. Az implementáció során a `keras` framework-öt használtuk (2.15.0 verzió), a szekvenciális self-attention-höz a `keras-self-attention` (0.51.0 verzió) könyvtárat, a keresztvalidáláshoz és a lineáris SVM-hez pedig a `scikit-learn` (1.6.1 verzió) könyvtárat használtuk (Pedregosa és mtsai, 2011).



3. ábra. Neurális modellek

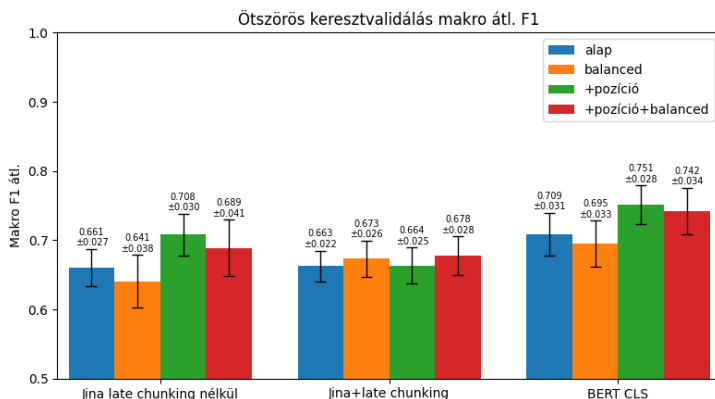
6. Eredmények

6.1. Lineáris kernelű SVM

Referenciaként a mondatvektorokat lineáris kernelű SVM-mel osztályoztuk. Finomhangoltuk a C paraméteret, amely $C=1$ beállításnál volt a legjobb, és kipróbáltuk a `class_weight=balanced` beállítást is, mivel az adatunk nem homogén eloszlású volt. Az SVM nem képes kihasználni a mondat szekvenciából származó plusz információkat, kivéve ha late chunkingot alkalmazunk. Ötszörös keresztvalidálást végeztünk a 299 dokumentumból álló tanítási és validálási halmazok felhasználásával, ügyelve a jogterület szerinti arányos mintavételezésre. Összehasonlítottuk a Jina és a BERT CLS beágyazásokat pozíciós jellemzőkkel és anélkül, valamint a `class_weight=balanced` beállítással. Mérőszámként a makro átlag F1 metrikát választottuk. Az eredményeket a 4. ábra mutatja.

A late chunking nélküli beágyazások (Jina late chunking nélkül és BERT CLS) hasonló mintát követtek: a `class_weight=balanced` beállítás használata rontotta a teljesítményt, azonban a pozíciós jellemző hozzáadása előnyös volt, a makro F1 átlagot 4,76%-kal (66,07%-ról 70,83%-ra) és 4,25%-kal (70,85%-ról 75,10%-ra) emelte. Ezzel szemben late chunking esetén a `class_weight=balanced` beállítás bizonyos mértékben javította az eredményeket, és a pozíciós jellemző hozzáadása is pozitív, de marginális hatással volt, míg a legjobb late chunkinggal elért eredmény jelentősen a late chunking nélküli legjobb eredmény alatt maradt (70,83% vs. 67,77%).

A Jina vektorok aluteljesítése meglepő a BERT CLS vektorokkal szemben, egyfelől mert ez egy finomhangolt beágyazás, másfelől pedig a late chunkinggal végzett vektorizálás számos adathalmazon nagyobb javulást hozott a rövidebb, mint a hosszabb chunkoknál (Günther és mtsai, 2024). Korpuszunk mondatai



4. ábra. Különböző vektorformák és tanítási paraméterek összehasonlítása: *pozíció*: mondat relatív pozíciója, *balanced*: `class_weight=balanced` beállítás.

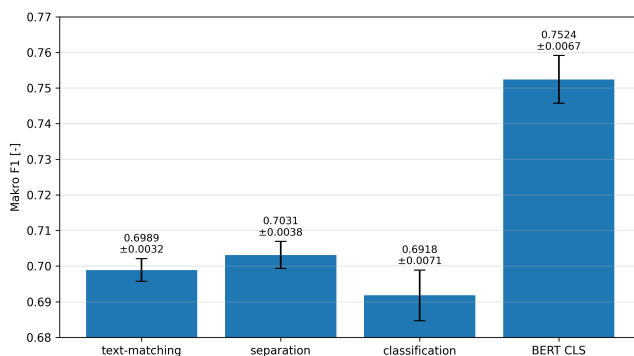
viszonylag rövidek voltak (átlagosan 50 token/mondat), ezért érdemi teljesítménynövekedést vártunk. Ezzel szemben egy friss tanulmányban Merola és Singh (2025) azt kapták, hogy Q&A feladatban a *late chunking* a visszakeresés teljesítményét rontotta, vagy legfeljebb csekély mértékben javította. A rossz teljesítmény miatt felmerült a Jina vektorizáló nem megfelelő beállítása. A Jina az alábbi vektorizálási feladatokat kínálja: *classification*, *text-matching*, *separation*, *retrieval.passage*, valamint *retrieval.query*, melyek közül a *classification* beállítást választottuk. Összehasonlítást végeztünk 5-szörös keresztvalidációval a tanító+validációs adatokon, már nem egy, hanem két ismétléssel, rögzített `random_seed` paraméterrel. A halmazok képzéséhez jogterület szerinti arányosított felosztást alkalmaztunk, mivel a mondat szintű osztályozás független a mondatok sorrendjétől. Összehasonlítottuk a *late chunking* nélküli Jina vektorokat a *separation*, *text-matching*, *classification* vektortípusokkal, illetve a BERT CLS beágyazásokkal. Az eredményeket az 5. ábra mutatja.

Érdekes módon a legjobb feladatnak nem a *classification* bizonyult, hanem a *separation* beállítás, de semelyik Jina beállítással sem sikerült megközelíteni a BERT CLS beágyazást.

Az eredmények tehát azt mutatták, hogy a mondatok jelentős része további szekvenciális információ kihasználása nélkül is helyesen osztályozható, illetve a *late chunking* gal kapott kontextus rontott a klasszifikáció során. Ugyanakkor bebizonyosodott, hogy a szekvenciális információ fontos, mert a pozíciós jellemző hozzáadásával az eredmények számottevően javultak.

6.2. Neurális modellek

Mivel a lineáris SVM-mel kapott eredmények arra utaltak, hogy a Jina beágyazások nem teljesítenek jól az RRL feladatunkban, a neurális architektúrákat



5. ábra. Jina `separation`, `text-matching` és `classification` beágyazási feladatok összehasonlítása a BERT CLS-sel lineáris SVM osztályozóval

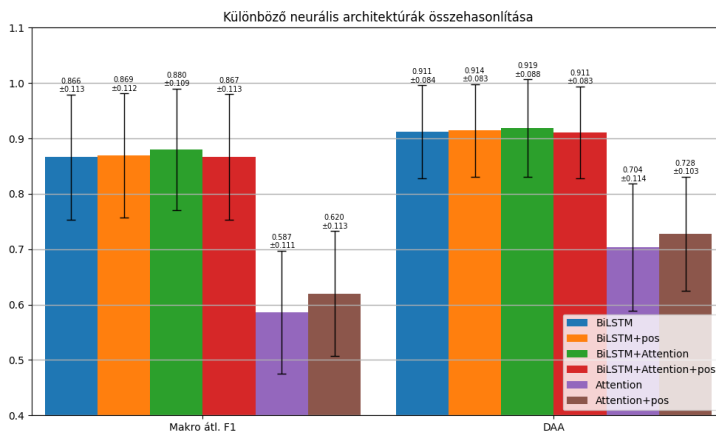
kizárólag a legjobban szereplő BERT CLS beágyazásokkal hasonlítottuk össze, pozíciós jellemző hozzáadásával, illetve anélkül. A tanítás során az adathalmazt 85% tanító és 15% validációs részre osztottuk, arányosított felosztással a jogterület és a régi ill. új dokumentumtípus szerint is. Minden epoch elején a dokumentumokat megkevertük (de a bennük lévő mondatokat nem), és kategorikus keresztentropia veszteségfüggvényt alkalmaztunk.

A tanítás során használt paramétereket a B. Függelék 7. táblázata tartalmazza, melyekhez rövidebb próbálgatásokat követve jutottunk el. Minden tanítást háromszor végeztünk el, három különböző véletlenszerű állapotot (random state) állítva be a tanító és validálási halmazok bontásához.

Fő összehasonlítási metrikaként a makro F1 átlagot és a dokumentumszintű átlagos pontosság (Document Average Accuracy-DAA) metrikát választottuk. Ezeket minden dokumentumra a validációs halmazon külön számoltuk ki, és ezek átlagát és szórását a 6. ábra mutatja be. Az eredmények alapján az Attention háló képes megtanulni a jogi ügyek szabályszerűségeit, de csak korlátozott mértékben. A BiLSTM és a BiLSTM+Attention hálók sokkal jobban megragadták a jogi dokumentumok szabályszerűségeit, mintegy 20-25 F1 pontos különbséggel megelőzve az Attention modellt. A pozíciós jellemző hozzáadásának hatása az Attention modellnél marginálisan, de javított, míg a másik két architektúrában elhanyagolható volt. Ezért csak a legjobban szereplő BiLSTM és BiLSTM+Attention hálókat hasonlítottuk össze a tesztalmazon.

6.3. Eredmények a tesztalmazon

A BiLSTM és a BiLSTM+Attention hálókat a korábbi szakaszban bemutatott beállításokkal újratanítottuk az egyesített tanító és validációs adatokon. Az így kapott modelleket ezután a tesztalmazon értékeltük ki, melyet a 3. táblázat mutat be. Az eredmények azt mutatták, hogy a BERT CLS+BiLSTM beállítás működött a legjobban, bár az összes beállítás viszonylag hasonlóan teljesített, és mindegyik jó teljesítményt nyújtott az RRL feladatban.



6. ábra. Neurális architektúrák összehasonlítása BERT CLS vektorokkal a validációs halmazon; **+pos**: pozíciós jellemzővel

3. táblázat. Eredmények a teszhalmazon

Beágyazás	Neurális modell	DAA	Accuracy	Makro F1	Súlyozott F1
BERT CLS	BiLSTM	0,9226	0,9247	0,8849	0,9252
BERT CLS+pos	BiLSTM	0,8926	0,8828	0,8356	0,8853
BERT CLS	BiLSTM+Attention	0,8806	0,8668	0,8209	0,8690
BERT CLS+pos	BiLSTM+Attention	0,8964	0,8731	0,8317	0,8751

6.4. Címkeszintű eredmények a teszhalmazon

A legjobb modellel számolt címkeszintű eredményeket a 4. táblázat tartalmazza.

4. táblázat. A legjobb modell címkeszintű eredményei

Címke	Pontosság	Fedés	F1	F1 Régi	F1 Új
Bírói érvelés	0,9747	0,9425	0,9583	0,9507	0,9915
Bíróság döntése	0,8341	0,8066	0,8201	0,8177	0,8421
Felek érvelése	0,9073	0,9407	0,9237	0,9078	0,9732
Egyéb	0,9939	0,8768	0,9317	0,9167	0,9625
Perelőzmény	0,9458	0,9178	0,9316	0,9131	0,9827
Perköltség	0,9389	0,9862	0,9620	0,9577	0,9783
Rendelkező rész	0,6144	0,6676	0,6399	0,6646	0,4267
Tényállás	0,8823	0,9432	0,9117	0,9114	0,9151

A címkék többsége esetében 0,9 feletti F1 értéket mértünk, két címke kivételével: a Bíróság döntése (0,8201) és a Rendelkező rész (0,6399) esetében. Jól

látható volt az is, hogy az új típusú dokumentumok esetében a modell jobban teljesített a régi típusú dokumentumoknál a Rendelkező rész kategória kivételével, alátámasztva, hogy az újabb típusú dokumentumok könnyebbek a kategorizálás szempontjából. Megállapítható volt még, hogy a régi dokumentumok esetében is nagyon jó eredményeket ért el a modell, a Bíróság döntése (0,8177) és a Rendelkező rész (0,6646) címkék kivételével mindenhol 0,9 feletti F1 értéket mértünk.

A Rendelkező rész címke a legkevesebbszer előforduló címke volt mind a tanító, mind a teszhalmazban. Ennek oka, hogy ez amolyan egyéb címke: csak azok a mondatok kapták ezt a címkét a határozat rendelkező részéből, amely egy bekezdésnyi szöveg a határozat elején, amelyek nem voltak besorolhatóak vagy a Perköltség vagy a Bíróság döntése kategóriába. Ezért ennek a címkének a fontossága sem nagy gyakorlati szempontból.

7. Összegzés

Bemutattuk tudomásunk szerint az első, magyar bírósági határozatokon működő, mondat szintű határozatrész-címkéző (Rhetorical Role Labeling-RRL) megoldást, amelyet egy újonnan összeállított korpuszon értékeltünk, és klasszikus, illetve neurális architektúrákkal vetettünk össze. Összhangban más nemzetközi kutatások eredményével azt tapasztaltuk, hogy a mondatok sorrendjéből származó információ jelentősen segíti a klasszifikáció pontosságát. A magyar BERT (huBERT) CLS-beágyazásokkal táplált BiLSTM adta a legerősebb összteljesítményt a teszhalmazon, egyértelműen felülmúlva a lineáris SVM alapmodellt, ami alátámasztja a szekvenciális információ jelentőségét. A visszakereséssel kapcsolatos irodalom friss eredményeivel ellentétben a „late chunking” rontotta a teljesítményt a mondat szintű RRL-ben, és a többnyelvű Jina v3 beágyazások sem bizonyultak jobbnak a magyar BERT CLS-nél. Ez arra utal, hogy a dokumentumkontextus „befecskenedése” fix mondatvektorokba zajt vihet a beágyazásokba, ezzel nehezítve a kategorizálást. A legjobb modellel a dokumentumok mondatai 92,2%-os átlagos pontossággal voltak osztályozhatók. A munka gyakorlati hatással is bír: a legjobb modell az Országos Bírósági Hivatalnál egy RAG-pipeline-ban teszi lehetővé a határozatrészek alapján történő szűrést, javítva a különböző jogi problémák kereshetőségét és magyarázhatóságát.

Hivatkozások

- Aragy, R., Fernandes, E.R., Caceres, E.N.: Rhetorical role identification for portuguese legal documents. In: Brazilian Conference on Intelligent Systems. pp. 557–571. Springer (2021)
- Bambroo, P., Adhikary, S., Bhattacharya, P., Chakraborty, A., Ghosh, S., Ghosh, K.: Marro: multi-headed attention for rhetorical role labeling in legal documents. *Artificial Intelligence and Law* pp. 1–30 (2025)
- Bhattacharya, P., Paul, S., Ghosh, K., Ghosh, S., Wyner, A.: Identification of rhetorical roles of sentences in indian legal judgments. In: *Legal knowledge and information systems*, pp. 3–12. IOS Press (2019)

- Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., Liu, Z.: M3-embedding: Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In: Findings of the Association for Computational Linguistics ACL 2024. pp. 2318–2335 (2024)
- Csányi, G.M., Lakatos, D., Üveges, I., Vági, R., Megyeri, A., Fülöp, A., Nagy, D., Vadász, J.P.: Bírósági határozatok automatikus mondatszegmentálásának hatékonyságmérése. In: XX. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2024). Szegedi Tudományegyetem, Informatikai Intézet (2024)
- Csányi, G.M., Lakatos, D.P., Vadász, J.P., Nagy, D., Üveges, I.: A kontextusablakon kihajolni nem veszélyes: jogi szövegek hatékony szemantikus keresése. In: XXI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2025). Szegedi Tudományegyetem TTIK, Informatikai Intézet (2025)
- Günther, M., Mohr, I., Williams, D.J., Wang, B., Xiao, H.: Late chunking: contextual chunk embeddings using long-context embedding models. arXiv preprint arXiv:2409.04701 (2024)
- Günther, M., Sturua, S., Akram, M.K., Mohr, I., Ungureanu, A., Wang, B., Eslami, S., Martens, S., Werk, M., Wang, N., és mtsai: jina-embeddings-v4: Universal embeddings for multimodal multilingual retrieval. arXiv preprint arXiv:2506.18902 (2025)
- Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* 9(8), 1735–1780 (1997)
- Krippendorff, K.: Computing krippendorff’s alpha-reliability (2011), <https://repository.upenn.edu/entities/publication/034a6030-c584-4d14-9d3d-7b7e8d16df20>
- Malik, V., Sanjay, R., Guha, S.K., Hazarika, A., Nigam, S., Bhattacharya, A., Modi, A.: Semantic segmentation of legal documents via rhetorical roles. arXiv preprint arXiv:2112.01836 (2021)
- Marino, G., Licari, D., Bushipaka, P., Comandé, G., Cucinotta, T., és mtsai: Automatic rhetorical roles classification for legal documents using legal-transformer over bert. In: CEUR WORKSHOP PROCEEDINGS. vol. 3441, pp. 28–36. CEUR-WS (2023)
- Merola, C., Singh, J.: Reconstructing context: Evaluating advanced chunking strategies for retrieval-augmented generation. In: International Workshop on Knowledge-Enhanced Information Retrieval. pp. 3–18. Springer (2025)
- Nemeskey, D.M.: Introducing huBERT. In: XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021). p. TBA. Szeged (2021)
- Orosz, G., Szabó, G., Berkecz, P., Szántó, Z., Farkas, R.: Advancing Hungarian Text Processing with HuSpaCy: Efficient and Accurate NLP Pipelines. In: Text, Speech, and Dialogue. pp. 58–69. Springer Nature Switzerland (2023)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., és mtsai: Scikit-learn: Machine learning in python. *the Journal of Machine Learning Research* 12, 2825–2830 (2011)
- Sturua, S., Mohr, I., Akram, M.K., Günther, M., Wang, B., Krimmel, M., Wang, F., Mastrapas, G., Koukounas, A., Wang, N., és mtsai: jina-embeddings-v3:

Multilingual embeddings with task lora. arXiv preprint arXiv:2409.10173 (2024)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017)

Függelék

A. Annotátorok közötti egyetértés

5. táblázat. Annotátorok közötti egyetértés

Metrika	Érték
Átl. egyetértés	0,8706
Krippendorff Alfa	0,7777

6. táblázat. Címkeszintű annotátorok közötti egyetértés

Címke	Krippendorff Alfa	Átl. egyetértés
Bíróság döntése	0,4243	0,9693
Bírói érvelés	0,7568	0,8815
Felek érvelése	0,9479	0,9832
Egyéb	0,6998	0,9682
Perelőzmény	0,6768	0,9203
Perköltség	0,9283	0,9971
Rendelkező rész	0,8066	0,9942
Tényállás	0,3899	0,8175

B. Tanítási paraméterek

7. táblázat. Tanítás során használt paraméterek

Paraméter	Érték
Epochok	max 200
Tanulási ráta	0,001
Dropout	0,4
Rekurrens dropout	0,4
Batch méret	32
LSTM cellák	128
Elosztott dense neuronok	32
Early stopping	validációs veszteség
Early stopping patience	10 epoch, legjobb modell marad
Optimalizáló	AdamW
Attention window	10