Eötvös Loránd Tudományegyetem

Társadalomtudományi Kar

Mesterképzés

LSTM ÉS GRU NEURÁLIS HÁLÓZATOK TELJESÍTMÉNYÉNEK ÖSSZEHASONLÍTÁSA ÁLHÍREK OSZTÁLYOZÁSÁBAN KÜLÖNBÖZŐ ELŐFELDOLGOZÁSI STRATÉGIÁKKAL

Konzulens:
Buda Jakab Máté

Készítette:
Könye Máté

HLCZBE

Survey statisztika és adatanalitika szak

2025, április

EREDETISÉGNYILATKOZAT

Alulírott Könye Máté az ELTE TáTK a Survey Statisztika és Adatanalitika hallgatója fegyelmi felelősségem tudatában nyilatkozom, és aláírásommal igazolom, hogy a(z) LSTM és GRU neurális hálózatok teljesítményének összehasonlítása álhírek osztályozásában különböző előfeldolgozási stratégiákkal
című szakdolgozat/diplomamunka **saját, önálló szellemi munkám,** az abban hivatkozott, nyomtatott és elektronikus szakirodalom felhasználása a szerzői jogok szabályainak megfelelően történt.

Tudomásul veszem, hogy szakdolgozat/diplomamunka esetén plágiumnak számít:

- a szó szerinti idézet vagy fordítás közlése idézőjel és hivatkozás megjelölése nélkül;
- a tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát, valamint a HKR 74/A–C. §-aiban foglalt
rendelkezéseket megismertem, és tudomásul veszem, hogy

- a szakdolgozatom/diplomamunkám szövege a benyújtása után plágiumellenőrző szoftverrelvizsgálható,
- plágium esetén szakdolgozatom/diplomamunkám értékelés nélkül visszautasításra kerül,
- plágiumvétség esetén fegyelmi eljárás indítható.

Budapest, 2025. 04.12.

.....Könye Máté.....
a hallgató aláírás

Absztrakt

Az álhírek terjedése jelentős közegészségügyi, társadalmi és politikai kockázatokat hordoz, különösen a digitális médiatérben. Jelen tanulmány célja két fejlett rekurrens neurális hálózati architektúra, a Long Short-Term Memory (LSTM) és a Gated Recurrent Unit (GRU), teljesítményének összehasonlítása bináris álhírklasszifikációs feladatban. A modellek értékelése különféle szöveg-előfeldolgozási stratégiák (például lemmatizálás, stopword-kezelés, numerikus adatok átalakítása) mentén történt, GloVe szóbeágyazások felhasználásával. Az elemzésben több, egymástól független és tematikusan eltérő angol nyelvű hírkorpusz szolgált tanító- és teszthalmazként. Az eredmények arra utalnak, hogy bizonyos előfeldolgozási lépések, mint a számok szöveges formára hozása és a stopszavak megtartása, szignifikánsan növelhetik a prediktív teljesítményt. A GRU modellek jobb teljesítményt nyújtottak a 2016-os cikkeket tartalmazó teszthalmazokon, míg a legfrissebb, 2025-ös hírcikkeken az LSTM architektúra bizonyult megbízhatóbbnak és pontosabbnak. Az eredmények a neurális architektúrák és előfeldolgozási módszerek közötti kölcsönhatás jelentőségére világítanak rá, és irányt mutathatnak hatékonyabb automatizált álhírszűrő rendszerek fejlesztéséhez.

Kulcsszavak: álhírklasszifikáció, LSTM, GRU, előfeldolgozás, GloVe, rekurrens neurális háló, természetesnyelv-feldolgozás, szövegosztályozás

Tartalomjegyzék

Bevezetés

A rekurrens, visszacsatolt neurális hálózatok a szekvenciális adatok feldolgozásában a 21. században az egyik legfontosabb szerepet töltötték be. Ez a hálózattípus a hagyományos, rétegezett neurális hálózatok továbbfejlesztett változata, amely képes a bemeneti adatok időbeli függőségeinek kezelésére azáltal, hogy belső állapotát, memóriáját megőrzi és visszacsatolja a következő lépésekhez. Ezeknek is, legfőképpen a fejlesztett architektúrái vannak alkalmazva, mint a hosszú-rövid távú memória (LSTM), kapuzott rekurzív egység (GRU) és ezeknek fejlesztett verzió, mint a Bi-LSTM és Bi-GRU. Az LSTM és GRU típusú neurális hálózatokat számos kutatásban sikeresen alkalmazták különféle feladatokra, elsősorban képfeldolgozás és természetes nyelvfeldolgozás (NLP) területén. NLP-ben belül ezek az alkalmazási területek többek között a szentimentelemzés, a gépi fordítás, valamint a szövegklasszifikáció. Noha a GRU architektúra az LSTM-nél mintegy másfél évtizeddel később került bevezetésre, több empirikus vizsgálat alapján a GRU gyakran hasonló vagy jobb teljesítményt nyújt bizonyos NLP-feladatok esetében. Mindazonáltal a két modell gyakran egymás mellett is használatos, mivel eltérő problématípusokra eltérő mértékben érzékenyek, így az alkalmazásuk összehasonlítása továbbra is releváns kérdéskör a kutatásokban. Ez azon okból kifolyólag van, mert bár igaz, hogy GRU-nak vannak előnyei LSTM-el szemben, nincs általános konszenzus, melyik modell erősebb és lenne érdemesebb használni. Mindazonáltal, ezeket a neurális hálózati modelleket számos tényező erősen befolyásolja, legfőképp az adatbázis, korpusz, amin maga a modell tanul, illetve az adatbázisban szereplő adatok melyeket az NLP területén számos kombináció révén lehet kialakítani. Korábbi tanulmányok révén körvonalazódott egy kiindulási alap, hogy a természetesnyelv feldolgozás területén milyen módszereket érdemes alkalmazni az elemezni kívánt korpuszon, de közös megegyezés erről sincsen, mert ez a célfeladattól és kutatási területtől is függhet, pontosan milyen adatra van szükség a szövegből.

Ezen okokból kifolyólag az említett két népszerű Rekurrens neurális hálózat típusát, Long Short-Term Memory (LSTM) és Gated Recurrent Unit (GRU), vizsgálom meg és hasonlítom össze hatékonyságukat különböző adatfeldolgozási lépések alkalmazása révén, igaz és álhírek klasszifikációja során. Kutatásom során a következő kulcskérdéseket járom körbe. Először is, melyik modell adja a legpontosabb eredményt a kiértékelési szempontokat tekintve. Az

általam vizsgált kiértékelési szempontok, a gépi- és mélytanulási kutatások során gyakran elemzett értékek: *pontosság, precizitás, recall* és *F1-score*. Emellett a konfúziós mátrix vizsgálata, azon belül a hamis negatív és hamis pozitív cikkek átnézése, hogy miből adódhat a rossz besorolás Továbbá, megvizsgálni, hogy más előfeldolgozási módszereket alkalmazva mint, a stopszavak bent hagyása vagy kivétele, lemmatizálás vagy stemming, illetve a számok különböző kezelésével milyen változásokat hoz a két modellnél. A szakdolgozat, így hozzájárulhat annak feltárásához, hogy az egyes modellek mely előfeldolgozási lépésekkel kombinálva bizonyulnak hatékonyabbnak az álhírek osztályozásában. Emellett, irányt mutathat arra vonatkozóan, hogy magasabb predikciós pontosság érdekében milyen előfeldolgozási stratégiák alkalmazása lehet célszerű az álhírek felismerésének tekintetében. Végül, eltérő, független, más forrásokból gyűjtött adatbázisokon végzett teszteléssel értékelhető, hogy a vizsgált modellek közül melyek képesek hasonlóan kedvező teljesítményt nyújtani eltérő adatkörnyezetben is. Fontos azonban megjegyezni, hogy az elemzés elsősorban irányadó következtetéseket enged levonni, mivel az adathalmaz és a számítási erőforrások korlátjai nem teszik lehetővé általános érvényű megállapítások megfogalmazását.

A következőkben, először az elméleti hátteret alapozom meg, az NLP bemutatásával, és a nyelv, mint adat ismertetésével, hogy hogyan is tekinthető adatnak az általános nyelvi beszéd. Majd a továbbiakban a nyelvifeldolgozás különböző lépéseit mutatom be, mi miért fontos, mi elengedhetetlen vagy mi csak ajánlott a vizsgált szakirodalmak szerint. Majd ebből áttérek a szövegklasszifikációra, álhírek osztályozására és annak kihívásaira. Végül a saját kutatási eredmények bemutatása előtt még a rekurrens neurális hálózatok leírása, kitérve külön a LSTM és GRU részleteire korábbi kutatások bemutatásával.

Elméleti Háttér

Az NLP bemutatása, aktualitása

A Természetes Nyelvfeldolgozás (Natural Language Processing, NLP) napjaink egyik legdinamikusabban fejlődő tudományága, amely az emberi nyelv digitális adatok formájában történő elemzésére és feldolgozására összpontosít. Az NLP alkalmazási területei ma már rendkívül sokrétűek, a beszédfelismeréstől kezdve a szöveganalíziseken át a gépi fordításig, és

szerves részét képezi az olyan modern technológiáknak, mint a mesterséges intelligencia (Eisenstein, 2019, p. 1).

Mindemellett az NLP egyedi helyet foglal el a tudományos diszciplínák sorában, mivel egyesíti a nyelvészet, a matematika, a statisztika és a pszichológia elemeit. Ahogy Jurafsky és Martin (2025, p. 2) rámutatnak, az NLP feladata a nyelv mélyebb megértéséhez szükséges formalizált modellek és algoritmusok fejlesztése, amelyek figyelembe veszik a nyelv sokrétűségét a fonetikától a pragmatikáig. A technológia gyökereit a 20. század közepéig, Alan Turing és Noam Chomsky munkáiig lehet visszavezetni. Turing algoritmikus számítási modellje (1936) és Chomsky generatív nyelvtana (2002) alapvető fontosságúak az NLP fejlődésében.

Az NLP aktualitása különösen hangsúlyossá vált a mesterséges intelligencia és a gépi tanulás forradalmának köszönhetően. A hagyományos szabályalapú megközelítéseket felváltotta a statisztikai alapú NLP, amely a korpuszokból tanult preferenciákat használja a szintaktikai, szemantikai és pragmatikai kétértelműségek feloldására (Manning és Schütze, 1999). Az elmúlt évtizedekben a mélytanulás töltötte be a vezető szerepet az NLP-ben, különös tekintettel a rekurrens neurális hálózatokra a 2010-es években, amelyek specializált modellekként a szekvenciális adatok feldolgozására képesek (Goldberg, 2017, p. 133). Viszont az elmúlt időszakban nagyobb hangsúly került a transformerekre és azok figyelemmechanizmusára, amelyek párhuzamos feldolgozási képességük és hatékony tanulási architektúrájuk révén kiszorították a rekurrens megközelítéseket számos NLP-feladatban (Patwardhan és mtsai, 2023).

Nyelv és digitalizáció

Az én kutatásomban, több ok miatt is angol nyelvű cikkekkel és hírekkel foglalkozom, amiket a következőkben kifejtenék. Először is, más nyelvhez képest, ezen a nyelven található a legtöbb és legnagyobb adatbázisok nyelvi kutatások számára (Bender, 2019), így könnyen lehet keresni kutatási céltól függően megfelelő adatbázist, amennyiben nem rendelkezünk saját erőforrásokkal egy önálló készítéséhez. Emellett, az angol nyelvhez kapcsolódnak olyan különböző korpusz adatbázisok és programozási csomagok, amelyek kifejezetten az angol nyelvvel való kutatást és dolgozást segítik elő (Bird és mtsai, 2009; Miller, 1995; Marcus, Santorini, & Marcinkiewicz, 1993; Montani & Honnibal, 2018). Ez tekinthető előnyösnek, hiszen így a kutatók nagyszáma angol nyelvfeldolgozással dolgozik, ezzel egyre jobban

előrehajtva ennek a területnek a fejlődését, de ezáltal más nyelvek digitális feldolgozását és elemzését hátráltathatják, lassíthatják (Bender, 2019). Magától értetődően, azon nyelvek rendelkeznek a legoptimálisabb háttérrel, egy hozzám hasonló elemzés készítésével, melyet minél többen beszélnek angol mellett, mint például mandarin, spanyol és német (Bender, 2019).

Szöveg előfeldolgozás

Korábban említett, angol nyelvi felhasználásra készült programozási csomagok azért fontosak a neurális hálózatok készítéséhez, mert a modell tanítása előtt, a szöveget megfelelő formátumba, kell alakítani (Sharma és mtsai., 2024). Ezeket az átalakításokat különböző módszerekkel és lépésekkel lehet megvalósítani, annak függvényében milyen adatokkal, szöveggel szeretnénk pontosan dolgozni.

Normalizálás

Fontos kiindulási pont, a normalizálás és annak különböző lépései, ami szinte az összes hasonló területen végzetet kutatásban megjelenik (Arora & Kansal, 2019; Yolchuyeva és mtsai, 2018). Egyik ilyen kezdeti lépése a lowercasing, kisbetűzés. Lowercasing esetén a teljes szöveg kisbetűs formába kerül, aminek számos előnye van a természetes nyelvfeldolgozásban. Először is, biztosítja a konzisztenciát az adathalmazban, ezáltal csökkenti a szókincs méretét és annak dimenzionalitását (Jurafsky & Martin, 2025, p. 23). Illetve, az egységesített forma egyszerűsíti a további szövegfeldolgozási lépéseket (Chai, 2023).

Normalizálás egy másik fontos lépése a számok kezelése, ami a mai napig jelentős kihívást jelent a rekurrens neurális hálózatok használatában (Thawani és mtsai, 2021). Számos megközelítése van ennek a technikának, a leggyakoribb megoldások közé tartoznak a számok eltávolítása zajként (Petridis, 2024; Thawani és mtsai., 2021), helyettesítésük egy általános tokennel (pl. "[numtoken]"), szövegként való kezelésük tokenizációhoz (Thawani és mtsai., 2021), vagy olyan numerikus beágyazások alkalmazása, amelyek megőrzik a számok nagyságrendjét és skáláját (Zhou és mtsai, 2020). *Mivel a számok a tényalapú szövegek lényeges elemei, és a megbízható hírek jellemzően konkrét alapot nyújtanak numerikus összehasonlításokkal és pénzügyi kifejezésekkel* (de Oliveira, Pisa, Lopez, de Medeiros, & Mattos, 2021). Figyelmen kívül hagyásuk csökkentheti a modell teljesítményét, míg hatékony

feldolgozásuk javíthatja az álhírek azonosításának pontosságát. Kutatásomban, hogy én pontosan mely lépéseket végzem el, a módszertani fejezetben fejtem ki pontosan.

Ezeket elvégezve, még számos káros zaj szerepel a feldolgozandó szövegekben. A zaj olyan irreleváns vagy fölösleges elemeket foglal magában, mint a HTML címkék, túlzott központozás, speciális karakterek, emojik, ismétlődő karakterek (Al Sharou és mtsai, 2021; Petridis, 2024). Ezen további elemek helyes eltávolítása javítja a tokenizációt, a mondatszegmentálást és az osztályozási teljesítményt, ezáltal növelve az NLP-modellek hatékonyságát.

A speciális karakterek, például a hashtagek, a felhasználói megjelölések (@), az emojik és a pénznemek vagy százalékjelek gyakran eltávolításra vagy helyettesítésre kerülnek a szöveg egyszerűsítése érdekében. Korábbi kutatások azt mutatják, hogy ez a lépés általában növeli a modell hatékonyságát, azonban egyes alkalmazásokban ezek az elemek fontos információval bírhatnak (Michel & Neubig, 2018). Például az emojik és az internetes szleng kifejezések fontosak lehetnek a szentimentelemzés során (Michel & Neubig, 2018), míg a hírcikkek feldolgozásánál többnyire feleslegesek, mivel nem igazán szerepelnek benne. Ezért az előfeldolgozási lépéseket mindig az adott NLP-feladat követelményeihez kell igazítani annak érdekében, hogy ne vesszen el lényeges kontextus.

Szegmentálás és nyelvi szűrés

A zaj kiszűrés lépéseit folytatva, a stopszavak eltávolítása szintén lényeges lépés, mivel a gyakran előforduló, de alacsony információtartalmú szavak, például magyarban „a", „és”, "az" vagy az „is", angolban pedig „the”, „and”, „of” csökkenthetik a modell teljesítményét (Jurafsky & Martin, 2025, p. 24). Ezen szavak eltávolítása csökkenti az adathalmaz dimenzionalitását, és különösen hasznos az osztályozási, klasszifikációs vagy összegzési feladatokban, mivel segít a modellnek a lényegesebb tokenekre összpontosítani (Orebi & Naser, 2025). Ugyanakkor a túlzott stopword-szűrés veszélye, hogy eltávolíthat olyan nyelvtani elemeket, amelyek nélkülözhetetlenek a mondatszerkezet és a szövegösszefüggések értelmezéséhez, különösen a párbeszédmodellezésben vagy a szintaktikai elemzésben (Chai, 2023). Éppen ezért a két különböző megközelítés miatt, vizsgálom meg kutatásomban a stopword-ök jelentőségét is, hogy az én eredményem, melyik állítást támasztja alá az álhírek területén.

A zajok eltávolítása után, rá lehet térni a tokenizálásra. A tokenizálás a természetes nyelvfeldolgozás (NLP) egyik alapvető lépése (Bird és mtsai., 2009; Webster & Kit, 1992),

amely a szöveget egyéni tokenekre bontja, biztosítva annak strukturált feldolgozását további szintaktikai és szemantikai elemzésekhez (Chai 2023; Jurafsky & Martin 2025, p. 4). Ez a folyamat elengedhetetlen az adatok megfelelő szegmentálásához, lehetővé téve a szekvenciális adatok hatékony feldolgozását (Jurafsky & Martin,2025, p. 4; Webster & Kit, 1992). Bár az egyszerű szóközalapú tokenizálás elegendő lehet az alapvető szöveges feldolgozási feladatokhoz, nem képes megfelelően kezelni az összetett kifejezéseket, például az idiómákat vagy több szóból álló állandósult szókapcsolatokat, mint például a „kitalálni" angol megfelelője, „figure out" (Webster & Kit, 1992). Ezen problémák kiküszöbölésére fejlettebb tokenizálási módszereket, például részszó- vagy byte-páros kódolást alkalmaznak olyan esetekben, ahol mélyebb nyelvi elemzésre van szükség (Jurafsky & Martin 2025, p. 20).

Ezt követően a szavak további egységesítése történhet szótövezéssel. A szótövezés egyik fajtája a stemming, ami a szavakat a tövükre redukálja a toldalékok eltávolításával, ami egyszerűsíti a szöveg reprezentációját, de gyakran nem létező vagy torzított szavakat eredményez, például angolban „táncol".„dancing" szót „tánc"-„danc" alakra csonkíthatja (Haviana & Mulyono, 2023; Jurafsky & Martin, 2025, p. 24). Bár a stemming számítási szempontból gyors és hatékony, mivel kizárólag szabályokon alapul, nem veszi figyelembe a grammatikai szerkezetet, ami jelentésvesztéshez vezethet (Haviana & Mulyono, 2023). Ezzel szemben a lemmatizálás a szavakat szótári alakjukhoz igazítja, figyelembe véve azok nyelvtani kontextusát (Bird és mtsai., 2009; Jurafsky & Martin, 2025, 23.o), például a „aludt" („slept") szót „alszik" („sleep") alakra alakítva. Bár a lemmatizálás pontosabb, mint a stemming, nagyobb számítási kapacitást igényel, mivel mélyebb morfológiai elemzést végez (Elov, Khamroeva, & Xusainova, 2023). Továbbá, a lemmatizálás során nyelvtani kategóriák (POS-taggelés) alkalmazása szükséges, amelyhez a Penn Treebank címkézési rendszerét használják (Sahala & Lindén, 2023; Straka, Straková, & Hajič, 2019), majd pontos morfológiai elemzés érdekében a nyelvtani kategóriákat olyan lexikai adatbázisokhoz, mint a WordNet, kapcsolják (Miller, 1995). A stemmingel összehasonlítva a lemmatizálás nyelvészetileg megalapozottabb módszert kínál, amely megőrzi a szavak eredeti jelentését, viszont nagyobb számítási kapacitást igényel, ami hátrányt jelenthet nagyméretű szövegkorpuszok esetében (Haviana & Mulyono, 2023). A legújabb kutatások, például Haviana és Mulyono (2023) vizsgálata szerint a stemming negatívan befolyásolta a szövegklasszifikációs modellek teljesítményét, míg a

lemmatizálás nem hozott szignifikáns javulást. Ezt a két módszert a kutatásomban én is összehasonlítom, hogy alá tudom-e támasztani ez az eredményt.

Miután a szöveges adatok előfeldolgozása megtörtént, a következő lépés az adatok numerikus reprezentációjának kialakítása. A neurális hálózatok nem képesek közvetlenül a nyers szöveget értelmezni, ezért szükséges az adatok olyan módon történő átalakítása, amely lehetővé teszi számukra a hatékony tanulást. Az alábbiakban bemutatásra kerülnek azok a lépések, amelyek biztosítják a szöveg megfelelő formába hozását a modell számára.

Szöveg egységesítése neurális hálókhoz

Először is, fontos biztosítani, hogy a modell bemeneti adatai egységes hosszúságúak legyenek, mivel bár egyes neurális hálózatok, mint az RNN típusú modellek, képesek változó hosszúságú bemenetek feldolgozására, a gyakorlatban a tanítás hatékonysága érdekében gyakran egységes hosszúságú szekvenciákra van szükség. Ehhez kell alkalmazni a padding (kibővítés) és truncation (csonkítás) technikákat, amelyek segítenek a különböző hosszúságú szövegek egységesítésében. Meglehetősen kevés szakirodalom szerepel ennek az előfeldolgozási lépésnek leírásáról. Dwarampudi és Reddy (2019) tanulmányában szerepel leírással a bővítés és csonkítás 2-2 típusa. Mindkét módszer célja, hogy egy megadott hosszúságú szekvenciát érjünk el, ami lehet hosszabb vagy rövidebb, mint a vizsgált sorozat (Dwarampudi & Reddy, 2019; Yadav mtsai., 2020). Ha a sorozat rövidebb, mint a megadott egység, akkor bővíteni kell 0-val vagy [PAD] tokennel (Dwarampudi & Reddy, 2019; Yadav mtsai., 2020). Ezt a bővítést el lehet végezni az adatok előtt vagy után (Dwarampudi & Reddy, 2019). Hasonlóan a csonkítást is lehet elölről vagy hátulról kezdeményezni, hogy elérjük a kívánt hosszt (Dwarampudi & Reddy, 2019). Azért fontos megkülönböztetni a pre- és postpadding-et, mert Lopez-del Rio mtsai. (2020) kutatásában is bizonyították, hogy a bővítés pozíciója hatással van a végső modell eredményére. Emellett Dwarampudi és Reddy elemzésében (2019) és az az eredmény jött ki, hogy érdemesebb prepadding-et használni, mivel hatásosabb volt ez a módszer az LSTM esetében, ezért az én előfeldolgozásomban is csak ezt a módszert alkalmazom mindkét modell architektúra esetében.

Végső előkészítési lépésben, miután a szövegek hosszbeli egységesítése megtörtént, következik a szavak reprezentációja vektorok segítségével. A szóreprezentációk matematikai objektumok, gyakran vektorok, amelyek nyelvi jellemzőket kódolnak, ahol minden dimenzió

egy adott tulajdonságnak felel meg, ami segít a szavak közti kapcsolatok megragadására (Turian és mtsai, 2010). Ezeket a reprezentációkat, a szóbeágyazások úgy használják, hogy többdimenziós vektorokat (én modelljeimben 300 dimenziós) rendelnek a szavakhoz, amelyek geometriai tulajdonságai tükrözik a szemantikai hasonlóságokat (Garg és mtsai, 2018; Jurafsky & Martin, 2025; Mikolov és mtsai, 2017). A szóbeágyazások révén, a szavak közti kapcsolatok még tágabb összefüggésekben is értelmezhetők: például Garg és munkatársai, (2018) példának hozzák azt, hogy a „London" és „Anglia" közötti különbség ugyanolyan mintázatot követ, mint a „Párizs" és „Franciaország" közötti viszony, és ez így az ilyenfajta analógiák felismerésére is alkalmasak. A vektoralapú feldolgozásra különböző módszerek léteznek, például a TF-IDF, amely egy gyakran alkalmazott súlyozási eljárás a szövegek reprezentálására, különösen klasszikus gépi tanulási algoritmusok esetén, vagy a fejlettebb szóbeágyazási modellek, mint a Word2Vec (Mikolov és mtsai, 2013b) és a GloVe (Jurafsky & Martin, 2025, 123.o .; Pennington és mtsai, 2014). A szakdolgozatomban a GloVe szóbeágyazást használtam, ezért ezt mutatom be részletesebben.

Glove, vagyis Global Vectors for Word Representation, egy olyan szóbeágyazási módszer, ami hidat képez két korábbi domináns megközelítése között (Pennington és mtsai, 2014). Pontosan fogalmazva a globális mátrix faktorizációs módszerek, mint például a LSA (Latent Semantic Analysis) (Deerwester és mtsai., 1990), és a helyi kontextusablak-alapú módszerek közt, mint amilyen a skip-gram modell Mikolov és munkatársai (2013a) munkájában. Penningtonék azt is kiemelik, hogy míg LSA analógiás feladatokban alulteljesít, addig a skip-gram modell jól teljesít ezekben, de nem használja ki teljes mértékben a statisztikai információkat, mivel a globális előfordulási gyakoriságok helyett kizárólag helyi kontextusablakokra épít. Ezen hiányosságok kiküszöbölésére fejlesztették ki Pennington és munkatársai (2014) a GloVe modellt, ami egy súlyozott legkisebb négyzetek módszeren alapuló felügyelet nélküli tanulási eljárás, ami a szavak globális előfordulási statisztikáira épít. Az előfordulási mátrix használatával, amely megragadja a szavak közötti átfogó statisztikai kapcsolatokat, a GloVe olyan vektorokat hoz létre, amelyek jelentésteli belső szerkezettel rendelkeznek. Ennek eredményeképpen a tanulmányban (Penningtion mtsai., 2014) a modell 75%-os pontosságot ér el az analógiás feladatokat tartalmazó afdatbázison. Firoozi és társai (2022) kimutatták, hogy a GloVe hatékonyan integrálja a statisztikai és a kontextuális

információkat a szavak vektoriális reprezentációjának számításakor, így ez egy rendkívül megbízható beágyazási technika.

Pennington és munkatársai (2014) bizonyították, hogy pusztán az előfordulási valószínűségekre fókuszálás helyett az együttes előfordulási arányok jobban képesek megkülönböztetni a releváns és irreleváns szavakat. Empirikus eredményeik azt is alátámasztják, hogy a GloVe modell magasabb Spearman-féle rangkorrelációs értékeket ér el a szótári hasonlósági feladatokon, mint a Word2Vec. Ez azt jelzi, hogy a szóvektorok által számolt hasonlósági pontszámok jobban korrelálnak az emberi megítélésekkel. A GloVe számos szempontból előnyösebbnek bizonyult: a tesztek során jobb teljesítményt nyújtott öt különböző szószerűségi adathalmazon[1] (pl. WS353, MC, RG, SCWS, RW), ráadásul ezt fele akkora korpuszméret mellett érte el, mint a Word2Vec CBOW* modell (Pennington és mtsai, 2014). Korábbi kutatások (Abualigah és mtsai, 2024; Firoozi és mtsai, 2022; Polat & Cankurt, 2023), köztük olyanok is, amelyek álhírek felismerésével foglalkoztak (Kishwar & Zafar, 2021; Shrivastava & Sharma, 2021) azt is kimutatták, hogy ez a szóbeágyazási módszer hatékonyan működik különböző RNN modellekkel, köztük LSTM-el is, van ahol majdnem 100%-os pontosságot ér el a modell (Abualigah és mtsai, 2024), de van több 80% körüli eredmény is (Kishwar & Zafar, 2021; Shrivastava & Sharma, 2021). Ezek az eredmények függhetnek a használt adatbázison, szöveg előfeldolgozási lépéseken, illetve, az is lehet befolyásoló tényező, hogy statikus, vagy dinamikus a GloVe, vagyis, hogy tanítható-e vagy sem (Firoozi és mtsai, 2022). Firoozi és kutatótársai (2022) tanulmányában azt vizsgálták, hogy a finomhangolt modern szóbeágyazási technikák befolyásolhatják-e a mélytanulás pontosságát, aminek eredményéül az jött ki, hogy nemhogy csak javít a modell pontosságán, de segít növelni a modellek számítási hatékonyságát a tanítási idő szempontjából.

Ezekből az okokból kifolyólag döntöttem amellett, hogy dinamikus GloVe szóbeágyazást használom, hogy a modellem a lehető legpontosabb eredményt érje el. A következőkben a szövegklasszifikációra térek ki, és arra, hogy hogyan lehetséges elkülöníteni az álhíreket az igaziaktól.

[1] ***word similarity datasets**: olyan szópárokat tartalmazó teszthalmazok, amelyekhez emberi értékelés társul a szavak jelentésbeli hasonlósága alapján. Ezeket használják a szóvektorok minőségének objektív mérésére.*

Szövegklasszifikáció

A szövegklasszifikáció a természetes nyelvfeldolgozás egyik alapvető feladata, amely során egy osztályozó algoritmus előre meghatározott osztályokba sorolja a szövegeket (Hu és mtsai., 2020). Az osztályozási folyamat egyik legfontosabb lépése a korábban említett előfeldolgozás és annak különböző lépései (Pimpalkar és mtsai., 2021). Ez azért fontos, mert hatékony előfeldolgozás növeli a modell pontosságát és segíti az osztályozó tanulási folyamatát, mint ahogy korábbi kutatások bizonyították (Pimpalkar és mtsai., 2021). A szövegklasszifikáció széles körben alkalmazott módszer, amely számos területen, például webes keresésben, szentimentanalízisben és dokumentumok rendszerezésében is használatos (Hu és mtsai., 2020). Felhasználási területein is látni, hogy több csoportokba is lehet sorolni a vizsgált szövegeket, viszont az álhírek felismerésének esetében az egyik legelterjedtebb megközelítés az, hogy osztályozást bináris kategóriákra bontják, ahol a modellek azt tanulják meg, hogy egy adott szöveg valódi vagy hamis hírt tartalmaz-e (Oshikawa és mtsai., 2018).

Álhírek szövegklasszifikációs megközelítése

Az álhírekhez kapcsolódó dezinformációnak több típusával találkozhatunk az életben, illetve az interneten, közösségi médiákon. Több kutatásban is kifejtik ezeket a típusokat, milyen kapcsolata van az álhírekkel, illetve azokra milyen különböző NLP módszerek és kutatási megközelítések vannak (Oshikawa és mtsai., 2018; Su és mtsai., 2020). Az egyik ilyen hasonló, az álhírekkel szinte egybeépült probléma (Mridha és mtsai., 2021) a pletyka, szóbeszéd (rumor) felismerése. Jelenleg az NLP területén nincs konzisztens definíciója a szóbeszédek detektálásának (Oshikawa és mtsai., 2018), viszont a kettő közti, álhír és szóbeszéd közti különbséget le lehet írni úgy, hogy míg az álhírek szándékosan kreáltak, a pletykák csak meg nem erősített információk kérdéses forrásokból, ami a megtévesztés célja nélkül terjed (Mridha és mtsai., 2021; Su és mtsai., 2020). Egy másik kapcsolódó terület az álláspont meghatározása (stance detection). Az álláspont felismerés módszere annak előrejelzése, hogy egy szöveg támogatja vagy ellenez különböző állításokat, konzisztencia alapján, nem igazságtartalmat tekintve. Ez a módszer különbözik az álhírek detektálásától, de lehet ennek alfeladata, abban a tekintetben, hogy segíthet a dokumentumokból származó megfelelő bizonyítékok azonosításához és a téves információk elemzéséhez szükséges hitelességi jellemzők kinyeréséhez (Hu és mtsai., 2020; Oshikawa és mtsai., 2018).

Az álhírek behatárolásának sincsen pontosan meghatározott definíciója NLP területén, viszont több kutatásban is próbálták már lehatárolni (de Oliveira és mtsai., 2021; Mridha és mtsai., 2021; Su és mtsai., 2020; Traylor és mtsai., 2019). De Oliveira és munkatársai tanulmányában (2021) azt írják, hogy kezdetben az álhír kifejezés olyan hamis vagy szenzációhajhász információkra utalt, amelyeket releváns hírekként terjesztettek, de mára már összemosódott a közösségi médián is terjedő hamis hírekkel. Másik tanulmányokban (Mridha és mtsai., 2021; Traylor és mtsai., 2019) úgy jellemzik az álhíreket, mint olyan hírek, amelyek szándékosan kreált dezinformációkat vagy hamis állításokat tartalmaz, azzal a céllal, hogy a fogyasztókat megtévessze és mindezalatt abban a színben van feltűntetve, mint egy legitim hír. Célja lehet ezeknek az álhíreknek, korábbi megtörtént példákra alapozva, hogy a félrevezetés által hatással legyen a közvéleményre és közvélekedésre, formálni tudja a regionális és nemzeti párbeszédet, kárt okozzon vállalatoknak és egyéneknek, illetve befolyásolni tudja a választásokat, mint állítólagosan a 2016 Egyesült államokbelit (Traylor és mtsai., 2019).

Ezekből kifolyólag érzékelhető annak fontossága, hogy képesek legyünk detektálni az álhíreket. Egyik ilyen módszere a tényellenőrzés (Su és mtsai., 2020; Traylor és mtsai., 2019), viszont ez túlságosan időigényes, vagy nehéz automatizálni. Éppen, ahogy Traylor (2019) is kiemeli az automatizálás és annak gyorsaságának szempontjából érdemes a természetes nyelvfeldolgozási megközelítése.

A nyelvi megközelítés azért lehetséges, mert számos kutatás bizonyította milyen eltérések szerepelnek igaz és álhírek közt, mind címben, mind szövegben (de Oliveira és mtsai., 2021; Horne & Adali, 2017; Rashkin és mtsai., 2017; Traylor és mtsai., 2019). Horne és Adali (2017) kutatásukban kifejtik, hogy a címek közt, szignifikáns különbség van, ami erősebb, mint a főszövegben. Álhíreknél a címek hosszabbak, több a nagybetű, kevesebb funkciószavak és egyszerűbb szavakat használnak (Horne & Adali, 2017). Az álhírek egyik erősebb nyelvi különbsége, hasonlóan a címekhez, a hosszhoz is kapcsolható, viszont ebben a környezetben a hosszaság mutatja egy cikk igazságértékét, vagyis az álhírek rövidebbek tartalmilag (Horne & Adali, 2017). Emellett, korábbi kutatások alapján az álhírek szóhasználatának jellemzői közé tartozik, hogy ismétlődő nyelvezetet használ (Horne & Adali, 2017), olyan szavakkal, amik adott témában kevésbé relevánsak (de Oliveira és mtsai., 2021), illetve több társas vonatkozású, negatív érzelmű és cselekvést leíró szavakat használ a sűrűbben előforduló

névmások, panaszok és tagadó állítások mellett (de Oliveira és mtsai., 2021; Horne & Adali, 2017; Rashkin és mtsai., 2017). Továbbá nagyobb számban megtalálhatóak olyan kifejezések és szavak amíg a felnagyításban, túlzásban segíthetnek, mint a fokozott melléknevek és módosító határozószavak (Rashkin és mtsai., 2017). Ezzel szemben kevesebbszer szerepelnek főnevek, a hihető hírekhez képest, amelyekre az jellemző, még hogy gyakrabban használnak határozottabb kifejezéseket és kevésbé jellemző rájuk a bizonytalanságot kifejező szóhasználat, ami arra utal, hogy egyértelműbben, magabiztosabban írják le az eseményeket (Rashkin és mtsai., 2017).

Az említett nyelvi különbségekből látható, hogy a klasszifikációhoz megfelelően el tudnak különülni a kategóriák. Ez azért fontos, mert a klasszifikáció pontossága a besorolás részletességétől függ, és attól, hogy mennyire különül el a két vizsgált csoport (Dogra és mtsai., 2022). Abból az okból kifolyólag írom, hogy két csoport, mert mint korábban is említettem, a kutatásomban a bináris besorolási rendszert (álhír vagy igazi) alkalmazom, ami az egyik legelterjedtebb az álhírekkel foglalkozó klasszifikációs kutatások közt (Dogra és mtsai., 2022; Oshikawa és mtsai., 2018; Su és mtsai., 2020). Viszont, ez a módszer nem mindig a leghatékonyabb, mert egy hír lehet részben igaz és részben hamis is. Ennek megoldására hozza példának Su és munkatársai (2020) a LIAR adatbázist, ami rövid politikai állításokból áll, amiket olyan címkékkel láttak el, hogy ’pants on fire’ (teljesen hamis), false, barely-true, half-true, mostly-true és true (Wang, 2017). Ezeknek a klasszifikációs adatoknak mindegyik esetben a legnagyobb kihívása a helyes és pontos címkézés, ahol is minden cikket vagy állítást el kell látni a megfelelő leírással, ami nagyon időigényes és nehéz feladat.

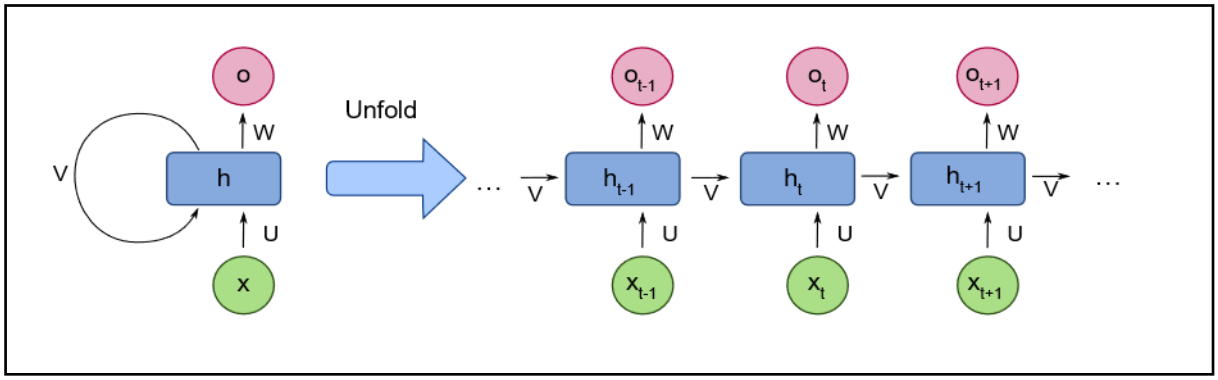
Ilyen klasszifikációs feladatok megoldásához többfajta megközelítés is van, mint a gépi-, vagy mélytanulási módszerek. Ilyen egyik legnépszerűbb mélytanulási módszer a szakdolgozatomban használt rekurrens neurális hálózat, és annak továbbfejlesztései, mint az LSTM és GRU.

Rekurrens Neurális Hálózatók

Az RNN, egy speciális neurális hálózat, ami az egyik legoptimálisabb megoldás szövegek feldolgozására, mert direkt a szekvenciális adatok vizsgálatára lett létrehozva (Prasad és mtsai, 2023). Emiatt számos területen alkalmazzák, mint a nyelvi modellezésben, fordításokban, kép- és hangfelismerésben (Poluru & Syed, 2023; Prasad és mtsai, 2023). Az RNN fő jellemzője,

amely megkülönbözteti más neurális hálózatoktól, például a feedforward neural network-től (FNN), az adatok szekvenciális feldolgozása. A rejtett állapot (hidden state) nem csupán a korábbi bemenetek információinak megőrzésére szolgál, hanem az időbeli összefüggéseket is kódolja (Shah és mtsai, 2022). A hálózat sajátossága, hogy minden időpillanatban ugyanazokat a rétegsúlyokat és rétegstruktúrát használja, vagyis a rétegek visszatérően ugyanazok maradnak. Ez a rekurrens, vagyis visszacsatolt működési elv teszi lehetővé, hogy a hálózat figyelembe vegye a korábbi állapotokat az új eredmények kiszámításánál. Egy ilyen alapmodell felépítése három rétegből áll. A folyamat első lépése a bemeneti vektor (x), amely a hálózatba érkező adatokat reprezentálja. Ezt követi a rejtett állapotot leíró vektor (h), amely a hálózat memóriáját és a korábbi állapotokból származó információkat hordozza. Végül a kimeneti vektor (o) adja a feldolgozás aktuális eredményét. Ez a kimenet, szöveggenerálási feladatok esetén, a következő időlépés bemeneteként is szolgálhat (Mridha és mtsai, 2021). Ez a folyamat az *első ábrán* látható.

1. *Ábra*: *Rekurens neurális hálózat (RNN) vizualizációja: Összenyomott és kifejtett forma*



Forrás: https://en.wikipedia.org/wiki/Recurrent_neural_network,
Letöltve: 2025.03.19

Ez alapján matematikai felírását a rejtett állapotnak, a következőképpen lehet felírni: (Noguer i Alonso, 2024):

1. $$h_t = f(Vh_{t-1} + Ux_t + b).$$

Ahol is, *h* mutatja a rejtett állapotot, *t* az idő lépést, *V* és U a súlymátrixot, *b* pedig a biast. A f egy nemlinearitást biztosító aktivációs függvény, ami azért szükséges, mert enélkül, az RNN csak egy lineáris rendszer lenne, amely nem tudna összetett kapcsolatokat modellezni (Mienye és mtsai, 2024). A f helyére szokás tanh, ReLU vagy Sigmoid függvényt használni, amik különböző eredényeket hozhatnak.

A kimenet kiszámításának egyenlete (Noguer i Alonso, 2024):

2. $$o_t = g(Wh_t + c)$$

Ebben az esetben, a h_t az aktuális időlépés kimenete, V hasonlóan a kimenti súlymátrix, *c* a kimeneti bias, míg *g* az aktivációs függvény (például sigmoid bináris klasszifikációnál) (Noguer i Alonso, 2024). A két képlet alapján alapján jól látható, hogy az előző időlépés rejtett állapota (h_{t-1}) befolyásolja az aktuális állapotot (h_t), ezáltal hozzájárulva az RNN memóriahatásához és lehetővé téve a szekvenciális mintázatok modellezését.

A modell tanítása során a súlyok (U, W, V) és a bias értékek kezdetben véletlenszerűen vannak inicializálva, vagy előre meghatározott értékeket kapnak. Ezek az értékek gradiensalapú tanulási módszerek segítségével frissülnek, jellemzően a *Backpropagation Through Time* (BPTT) algoritmus alkalmazásával (Pascanu, Mikolov és Bengio, 2013). A BPTT a klasszikus visszaterjesztéses algoritmus időbeli kiterjesztése, amely az időlépések mentén számítja ki a veszteségfüggvény gradienseit az egyes súlyokra vonatkozóan (Werbos, 1990;).

A veszteségfüggvény a modell által generált előrejelzés ($\hat{y}_t$) és a valódi címke (y_t) közötti eltérést méri. Bináris osztályozási problémák esetén az egyik leggyakrabban alkalmazott veszteségfüggvény a bináris keresztentrópia (Das és mtsai, 2020):

3. $$L = \frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)]$$

A gradiens alapú optimalizálás során a hálózat súlyai az alábbi módon frissülnek (Du és mtsai, 2019):

4. $$W^{(h)}(k) = W^{(h)}(k-1) - \eta \frac{\partial L(\theta(k-1))}{\partial W^{(h)}(k-1)}$$

ahol η a tanulási ráta.

Az RNN egyik fő korlátja ebben a tanulási folyamatban mutatkozik meg: hosszú szekvenciák esetén a gradiens eltűnhet vagy robbanhat. Amennyiben ez az érték túl alacsony, a súlyok minimálisat, nullához közeli értékkel változnak, vagy túl nagyot. Az előbbi esetben az úgynevezett eltűnő gradiens probléma lép fel, amely megakadályozza, hogy a modell hatékonyan tanuljon hosszútávú összefüggéseket. Ennek oka, hogy a távoli időpillanatokból származó gradiens összetevők fokozatosan elhalványulnak, így nem jutnak el a korábbi rétegekhez, ezáltal a hosszabb távú információk elvesznek a tanulási folyamat során. Főleg a

hosszabb szekvenciális adatoknál jelentkezik ez, épp ezért fontos volt erre megoldásokat keresni. Erre megoldásként fejlettebb RNN architektúrákat hoztak létre, mint az LSTM és a GRU hálózati modellek. Ezek úgynevezett kapukat használnak, amelyek szabályozzák, hogy az információ milyen mértékben maradjon meg a hosszabb távú tanulás során. A kapuk működésének alapja az additivitás elve, amely lehetővé teszi, hogy a gradiens értékei stabilabbak maradjanak, így az eltűnő és felrobbanó gradiens problémák hatása csökkenjen. Ennek eredményeként az LSTM (Gers és mtsai, 2000; Hochreiter & Schmidhuber, 1997) és GRU (Cho és mtsai, 2014) modellek hatékonyabban képesek hosszútávú összefüggéseket megtanulni és jobb teljesítményt nyújtani.

Bajaj (2017) kutatásában, mind három hálózati modellt használta álhírek klasszifikációjára más módszerek mellett, ahol bár precizitás terén jó teljesítményt nyújtott az alap RNN, a másik két modell megmutatta, miért is érdemes őket választani, magasabb eredmény érdekében. Továbbá, a két fejlesztett modelltől eltekintve, az alap RNN túlteljesítette a Feedforward Networkot és a logisztikus regressziót is, ezzel mutatva, hogy remek teljesítményt tud nyújtani.

Saleh és munkatársai (2021) négy különböző adatbázisra megnézve alkalmazták a szimpla RNN és LSTM modelleket álhírek klasszifikációjához. Két-két modellt alkalmaztak, egyet egy réteggel és egyet kettővel. Az LSTM mind a négy esetben jobb eredményeket adott, és amíg az eredménye javult a két rétegű esetében. a szimpla RNN-nél ez romlott. Az *1. táblázatban*, ahol a kapott eredményeket szemléltetem, ebből a kutatásból csak a két népszerűbb adatbázis (FakeNewsNet, ISOT) értékeit illesztem be a két réteges eredményekkel. A kapott eredményekből az látható, hogy az adatbázis megválasztása milyen eltérő eredményeket tud hozni.

Malhotra és Mahur (2022) egy másik kutatásban coviddal kapcsolatos álhírek klasszifikációban alkalmazta a három eddigiekben is említett modellt. Különböző előfeldolgozási lépéseken végighaladva, egy 10000 adatot tartalmazó adatbázison végezte el az elemzést. A kapott eredmények itt azt mutatják, hogy a három modell közt minimális különbség van, mindegyik esetében 92% körüliek a vizsgált mutatók.

A bemutatott kutatások eredményei egyértelműen rávilágítanak arra, hogy a szimpla RNN bizonyos esetekben versenyképes teljesítményt tud nyújtani, az LSTM és GRU, általában jobb

eredményeket érnek el. A következőkben ezen két fejlettebb architektúra működését, mutatnám be jobban, sajátos előnyeikkel és hátrányaikkal további szakirodalmak segítségével.

1. Táblázat: *RNNS, LSTM és GRU eredmények összehasonlítása*

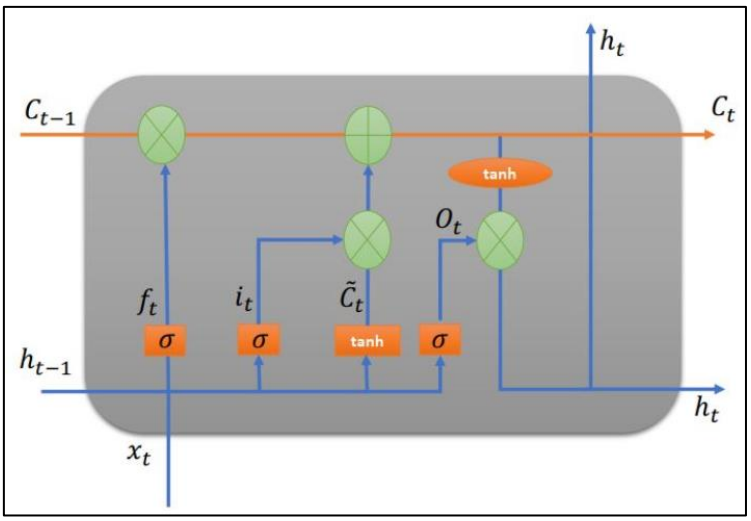
Forrás	Adatbázis	Modell	Pontosság	Precizitás	Recall	F1
Bajaj (2017)		RNN		91%	56%	70%
		GRU		89%	79%	84%
		LSTM		93%	72%	81%
Saleh és munkatársai, 2021	FakeNewsNet	RNN	79,84%	80,23%	79,84%	74,91%
		LSTM	86,43%	85,9%	86,43%	85,95%
	ISOT	RNN	79,84%	80,23%	74,91%	79,84%
		LSTM	99,78%	99,78%	99,78%	99,78%
Malhotra és Mahur (2022)	COVID-19 Fake News	RNN	92,38%	92,35%	92,14%	92,14%
		GRU	92,9%	92,95%	92,9%	92,9%
		LSTM	92,89%	92,92%	92,9%	92,89%

LSTM

A Long Short Term Memory alapjait tekintve egy régi módszer, Hochreiter és Schmidhuber 1997-ben publikálta az első tanulmányt róla. Az alap RNN hibáira több lépésben találtak megoldást. Egyik ilyen a Constant Error Carousel (CEC) mechanizmus, vagyis a memory cell bevezetése, ami a modell számára lehetővé tette, hogy az információt hosszú ideig tárolja a cellaállapot révén, ezzel megakadályozva a gradiens eltűnését vagy túl nagyra növését. Ezzel páhuzamban bemutatták a kapu egységeket (gated units), amelyek hatékonyabbá tették a hosszútávú függőségek kezelését. Viszont, ez a modell még csak bemeneti (input) és kimeneti (output) kapuval rendelkezett, amit később egészítettek ki az elfelejtési (forget) kapuval (Gers és mtsai, 2000). Erre azért volt szükség, mert az alap LSTM modell belső cellái, a folyamatos adatok feldolgozásával, korlátlanul növekedhetnek. A forget kapu, ezt a problémát úgy oldja meg, hogy adaptívan tanul elfelejteni, vagyis célzottan törli vagy csak csökkenti az olyan memória cella értékét amelyek már nem relevánsak (Gers és mtsai, 2000). Ezáltal a modell önállóan szabályozza a saját memória frissítését külső mechanizmus nélkül.

Így a három kapu, amelyek a *2. ábrán* a szigma (σ) aktivációs függvénnyel szerepelnek, az alábbi módon szabályozzák az információáramlást az LSTM cellán belül. Először is a felejtő kapu (f_t) meghatározza, hogy a korábbi cellaállapotból (C_{t-1}) mennyi információt kell megőrizni vagy elfelejteni, majd a bemeneti kapu (i_t) kontrollálja, hogy az aktuális bemenetből (x_t) és az előző rejtett állapotból (h_{t-1}) mennyi új információ kerüljön a cellaállapothoz (C_t). Az új információt először egy úgynevezett jelölt cellaállapot ($\tilde{C}_t$) generálja, amelyet egy hiperbolikus tangens (tanh) aktivációs függvény hoz létre, majd ezt a bemeneti kapu szabályozza, hogy milyen mértékben épüljön be a cellaállapotba. Végül pedig a kimeneti kapu (o_t) szabályozza, hogy a frissített cellaállapotból (C_t) milyen információ kerüljön a következő időlépés rejtett állapotába (h_t), amelyet a hiperbolikus tangens (tanh) aktivációval transzformálva továbbít a következő réteg vagy időlépés számára.

2. Ábra: *LSTM belső architektúrája: Kapuk és állapotfrissítések*



Forrás: Mateus és mtsai, 2021

A kapuk és cellaállapotok pontos kiszámítását a következő egyenletek írják le (Jozefowicz és mtsai, 2015; Mateus és mtsai, 2021):

5. $$f_t = \sigma(U_f x_t + V_f h_{t-1} + b_f)$$
6. $$i_t = \sigma(U_i x_x + V_i h_{t-1} + b_i)$$
7. $$o_t = \sigma(U_o x_t + V_o h_{t-1} + b_o)$$
8. $$\tilde{C}_t = tanh(U_C x_t + V_c h_{t-1} + b_c)$$
9. $$C_t = \sigma(f_t \times C_{t-1} + i_t \times \tilde{C}_t)$$
10. $$h = o_t \times tanh(C_t)$$

Amelyekben σ a szigmoid aktivációs függvényt, tanh a hiperbolikus tangens aktivációs függvényt, × az elemenkénti szorzást, V, U, b a tanulható súlyokat és biasokat jelölik.

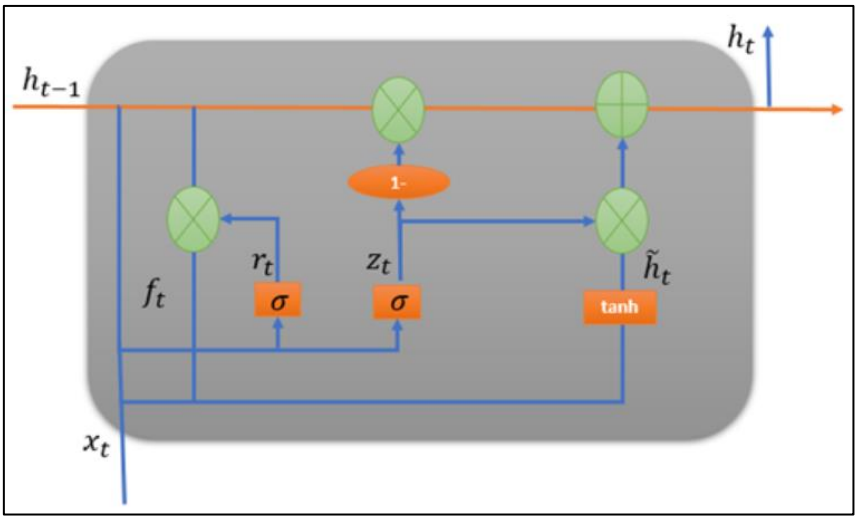
Konkrét példában az álhírek esetében, egy szónál ez a következőképp működik. Legyen a bemenet (x_t) az „elképesztő” (incredible), ami egy vektorként lép be a modellbe egy szóbeágyazás réteg révén, én esetemben a GloVe által. Ez egyszer az elfelejtési kapun (f_t) megy keresztül, ami meghatározza, hogy az előző kontextusból mennyi információ maradjon meg. Például, ha a szó előzményében objektív tartalom állt (pl. „tanulmány kimutatta”), de az „elképesztő” szenzációhajhász jelleget sugall, a hálózat csökkentheti az előző információ jelentőségét. Ezt követően a bemeneti kapu (i_t) dönt arról, hogy az új információ milyen mértékben frissítse a hosszútávú memóriát, cellaállapotot (C_t). Ebben az esetben, ha LSTM felismeri, hogy az érzelmi töltetű kifejezések indikátorai lehetnek az álhíreknek, az „elképesztő” szó erősebb súlyt kap, és nagyobb hatással lesz a memóriafrissítésre. Legvégül, a kimeneti kapu (o_t) szabályozza, hogy az „elképesztő” szó mennyire befolyásolja a következő időlépést. Ha a szó hozzájárulhat az álhírek felismeréséhez, az LSTM felerősíti a hatását a következő szóra, például „felfedezés”, amely így már egy módosított kontextusban kerül feldolgozásra. Ezzel a mechanizmussal az LSTM képes felismerni és súlyozni azokat a kulcsszavakat, amelyek jellemzőek lehetnek az álhírek nyelvezetére, ezáltal segítve a pontosabb klasszifikációt.

Az így felépülő modell biztosítja, hogy az LSTM hosszú szekvenciális adatokat is képes legyen megtanulni, amit több RNN-hez hasonló területen végzett kutatás be is bizonyított, mint a szövegfelismerésben, kép leírás generálásban és a természetes nyelvfeldolgozásban. Alnabhan és Branco áttekintésében (2024), álhírek detektálásával foglalkozó cikkeket mutatott be, ahol a tanulmány kiadásáig, az ilyen tudományos publikációk 72%-a, vagyis 106 darab alkalmazta az LSTM-et, vagy annak kétoldali fajtáját (Bi-LSTM), egyedül vagy más modell mellett. A sima LSTM eredményei 80% körül mozogtak. További kutatási eredményeket a GRU modell bemutatását követően írok le, miközben összehasonlítom a két modellt.

GRU

A kapuzott rekurzív egység, vagyis GRU (Gated Recurrent Unit) egy újabb neurális hálózati architektúra, ami az eltűnő gradiens problémáját, úgy oldja meg, hogy egyszerűsíti az LSTM alap koncepcióját (Cho és mtsai, 2014). Cho és munkatársai (2014) publikálták az első tanulmányt róla 2014-ben. Megegyezik abban az LSTM-el, hogy ahhoz hasonlóan a GRU kapus mechanizmust használ, amely megoldja az eltűnő gradiens problémáját, viszont ez alacsonyabb komplexitású és kevesebb paramétert használ (Nosouhian és mtsai, 2021). A három kapu helyett kettő szerepel, a frissítő (update) és az visszaállító (reset) kapu (Cho és mtsai, 2014). Ezt úgy érték el, hogy a korábbi elfelejtési, és bemeneti kaput kombinálták a frissítővel, miközben egybevonták a cella (LSTM memóriáját) és a rejtett állapotot (Mienye és mtsai, 2024).

3. Ábra: *GRU belső architektúrája: Kapuk és állapotfrissítések*



Forrás: Mateus és mtsai, 2021

A második ábrához hasonlóan, a harmadik ábrán, vagyis a GRU felépítésében t időlépésben, is a szigma (σ) aktivációs függvényekkel jelölt blokkok reprezentáljak a kapukat. A két kapu segítségével szabályozza az információ áramlását az időlépések között. Egy adott szó példájával a következőképp lehet levezetni, mit is csinál pontosan egy GRU modell. Legyen a „hihetetlen” szó ebben a példában. Első lépésben egy szóbeágyazási rétegen keresztül vektorizálódik (x_t), amely tartalmazza annak szemantikai és kontextuális információit. Majd a modell a visszaállító kaput (r_t) alkalmazva meghatározza, hogy az előző rejtett állapotból (h_{t-1}) mennyi információt vegyen figyelembe, az aktuális bemenetet (x_t), „hihetetlen”-t figyelembe véve. Ha R_t értéke alacsony, a hálózat inkább az aktuális bemenetre koncentrál,

míg magas érték esetén a korábbi kontextus erősebben befolyásolja az új állapotot. Ezt követően, a modell kiszámolja a jelölt rejtett állapotot ($\tilde{h}_t$), amely a „hihetetlen” szó kontextusbeli szerepét tükrözi. A frissítési kapu (z_t) határozza meg, hogy az új vagy a korábbi információ domináljon. Ha a „hihetetlen” egy feltételezett álhír kulcsszava, ez a kapu nagyobb súlyt ad neki, ha viszont semleges környezetben jelenik meg, kevésbé befolyásolja a következő időlépést. Legvégül az új rejtett állapot (h_t) a frissítési kapu által szabályozottan ötvözi a múltbeli és az aktuális információt, amelyet a modell továbbvisz a következő szavak feldolgozásához.

A GRU rejtett állapotainak és kapuinak pontos kiszámítása a következő egyenletekkel írható le (Jozefowicz és mtsai 2015; Mateus és mtsai, 2021):

11. $$r_t = \sigma(U_r x_t + V_r h_{t-1} + b_r)$$
12. $$z_t = \sigma(U_z x_t + V_z h_{t-1} + b_z)$$
13. $$\tilde{h}_t = tanh(U_h x_t + V_h(r_t \times h_{t-1}) + b_h)$$
14. $$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t$$

A jelölések megegyeznek az LSTM jelöléseivel.

Felépítésében és folyamatában észrevehető a hasonlóság a két modell közt, viszont szembetűnő a GRU egyszerűsége a LSTM-hez képest. Korábbi kutatások bizonyították (Mienye és mtsai, 2024; Mridha és mtsai, 2021; Nosouhian és mtsai, 2021) hogy nemcsak egyszerűbb felépítésű, hanem hatékonyabb is az időbeli összefüggések modellezésében, mivel egyszerűbb szerkezete révén könnyebben implementálható és gyorsabban konvergál, így képzése kevesebb időt igényel. Emellett a GRU egy újabb algoritmus, amely teljesítményben összevethető az LSTM-mel, de számítási szempontból hatékonyabb, ezáltal különösen előnyös lehet olyan alkalmazásokban, ahol az erőforráskorlátok kritikus szerepet játszanak (Nosouhian és mtsai, 2021). Ennek ellenére Nosouhian és kutatótársai azt is kiemelik, hogy bár megvan a maga előnye a GRU-nak, a két modell közti választás bizonyos feladatok és adathalmaz választástól függhetnek, mivel lehetnek olyan feladatok, amelyek jobban teljesítenek az LSTM hozzáadott komplexitásával és mechanizmusával.

Neurális hálózatok valószínűségi kimenete

A rekurens neurális hálózatok (RNN-ek), mint a bemutatott LSTM és GRU architektúrák, kimeneti rétegének kialakítása és interpretációja kulcsfontosságú, a különböző feladatok, mint a bináris (sigmoid) és multiklasszifikációs (softmax), feladatok számára. A bináris osztályozási feladatokban, ahol a célváltozó $y \in \{1,0\}$, a modell kimenete gyakran valószínűségként értelmezhető, ami azt fejezi ki, hogy az adott bemenethez tartozó kimenet milyen eséllyel tartozik az 1-es osztályba, azaz (Goodfellow és mtsai, 2016, p. 184; Murphy és mtsai, 2012, p. 5):

15. $$\hat{y} = \mathrm{P}(\mathrm{y} = 1|\mathrm{x})$$

A tanulás célja ilyenkor egy olyan függvény közelítése, amely megbízható előrejelzést tud adni ismeretlen, korábban nem látott bemenetek esetén is (Murphy, p. 3). Ezt a függvényt az $f(x) \approx \hat{f}(x)$ formában, a tanulási folyamat során a bemeneti–kimeneti párokra alapozva közelítjük. A bináris klasszifikáció során a kimeneti valószínűségi érték előállítása általában két lépésből áll. Először a hálózat egy lineáris transzformáció segítségével kiszámítja a bemenetből származtatott jellemzők súlyozott összegét:

16. $$z = w^T h + b$$

ahol h a bemeneti reprezentáció (pl. egy rejtett réteg aktivációja), w a súlyvektor, míg b a bias. Ezt követően egy sigmoid aktivációs függvény segítségével a lineáris kimenetet [0, 1] intervallumba „préseljük” (Goodfellow és mtsai, 2016, p. 183):

17. $$\sigma(z) = \frac{1}{1+e^{-z}}$$

Ez a sigmoid (más néven logisztikus vagy logit) függvény a teljes valós számhalmazt leképezi a zárt [0, 1] intervallumra, így a kimenet valószínűségként értelmezhető. Az ilyen „S” alakú függvényeket szokás „squashing function”-nek is nevezni, mivel összenyomják az értéktartományt egy adott skálára. A származtatott valószínűségi érték alapján döntési szabályt is ki lehet alakítani, például 0.5-ös küszöbérték alkalmazásával (Murphy és mtsai, 2012, p. 21):

18. $$\hat{y}(x) = f(x) = \begin{cases} 1, ha\ P(y = 1|x) > 0{,}5 \\ 0, \quad különben \end{cases}$$

A valószínűségi kimenet nemcsak a döntés meghozatalában, hanem a tanulási folyamatban is fontos szerepet játszik. A log-likelihood optimalizálása során a kimeneti valószínűség logaritmusát használjuk fel. Mivel a logaritmus csak pozitív számokra értelmezett, szükséges, hogy a modell kimenete valóban a (0,1) intervallumba essen. Ez a követelmény is indokolja a sigmoid függvény használatát, mivel az garantálja ezt a tartományt (Goodfellow és mtsai, 2016., p. 197). Ezenkívül, a sigmoid függvény nemcsak kimeneti rétegként használható, hanem rejtett rétegként is, viszont ebben az esetben a gradiensek szaturációja miatt ritkábban alkalmazzák. A rekurens neurális hálózatokban (pl. LSTM, GRU) azonban gyakori a sigmoid és a hozzá kapcsolódó aktivációs függvények (mint a tanh) használata, mivel ezek képesek megfelelő módon modellezni az időbeli dinamikát és nem-linearitást (Goodfellow és mtsai, 2016, p. 195).

Összességében az RNN-alapú modellek valószínűségi kimenete lehetővé teszi, hogy a modell ne csak merev döntéseket hozzon, hanem megbízható becsléseket is adjon az osztályba sorolás bizonyosságáról – ez különösen fontos olyan területeken, mint az álhírek osztályozása, ahol az eredmények megbízhatósága kiemelten fontos lehet.

Korábbi eredmények

A korábbi szakirodalmakban az álhírek detektálásában elsősorban az LSTM modellt alkalmazták, tekintettel annak korábbi bevezetésére és bevált teljesítményére a szekvenciaalapú feladatokban. Az utóbbi években viszont a GRU egyre szélesebb körű elterjedésével számos tanulmány kimutatta, hogy bizonyos esetekben hatékonyabb lehet, gyakran jobb eredményeket produkálva a klasszifikációs teljesítmény és számítási hatékonyság szempontjából. Az alábbi táblázatban olyan kutatásokból mutatok be eredményeket, amelyek mind álhírek klasszifikációjával foglalkoztak különböző modellekkel.

Toor és munkatársai (2025) több adatbázison is végzett el álhírekkel foglalkozó klasszifikációt. Ebben a kutatásban több előfeldolgozási lépést is alkalmaztak, amiket a korábbiakban bemutattam, ilyen volt a stopszavak eltávolítása, tokenizáció, stemming és a központozás kivétele a korpuszból. A szavakat, a GloVe és az TF-IDF (Term Frequency-Inverse Document Frequency) segítségével is vektorizálták, de mivel én csak a GloVe-ot alkalmazom, így csak az ezzel elért eredményt mutatom be. A két használt korpusz, amin az előfeldolgozást végrehajtották, az a Politifact és a Buzzfeed korpusza volt. Az így kapott eredményeken, amit

a 2. táblázat szemléltet, az látható, hogy mindkét esetben a kiértékelési szempontokat tekintve, a GRU jobb teljesítményt nyújtott, mintegy 10%-al. Az LSTM modell átlagban 76%-ot ért el, míg a másik 87%-ot. Amennyiben csak ezt az egy kutatást vizsgálnánk, egyértelműnek tűnne, hogy melyik modellt érdemesebb használni a jobb eredmény érdekében, viszont más publikációk nem mutatnak ilyen jelentős különbséget a modellek közt.

A korábbi kutatással szemben, korábban is említett Bajaj (2017) által elért eredményeiben az látható, hogy kiértékelési szempontoktól függően van, hogy az LSTM jobb a GRU-nál. Kutatásban az előfeldolgozási lépésekről nem írnak, lehet nem is volt a szövegeken alakítva, egyedül a GloVe embedding van kiemelve, mint használt módszer, amit a lefuttatás során nem frissítettek. Az így kapott eredmény azt mutatja, hogy precizitás szempontjából LSTM jobb eredményt ért el a 93%-al, szemben a 89%-el. Viszont a GRU itt is jobban teljesített recall és f1 érték szempontjából, előbbi 79% és 72% utóbbi pedig 84% és 81%.

Rachmawati és Darmawan kutatásukban (2024) szintén az álhírek detektálásával foglalkoztak, azonban vizsgálatukat nem angol, hanem indonéz nyelvű cikkeken végezték. Tanulmányukban részletesen bemutatták az alkalmazott előfeldolgozási lépéseket, amelyek közé tartozott a tokenizáció, a kisbetűsítés, a stop szavak eltávolítása és a lemmatizáció. Emellett különös figyelmet fordítottak az általuk használt modellek pontos architektúrájának ismertetésére, részletezve azok rétegszerkezetét. Az általuk alkalmazott LSTM és GRU modellek egyaránt négy rétegből épültek fel: a bemeneti réteget egy beágyazási (embedding) réteg követte, ezt követően a megfelelő rekurrens neurális hálózati réteg (LSTM vagy GRU), végül pedig egy teljesen összekapcsolt (dense) kimeneti réteg zárta a struktúrát. A kísérleti eredmények azt mutatták, hogy a két modell teljesítménye szinte teljes mértékben megegyezett, mindössze 1%-os eltérés volt tapasztalható az értékelési metrikák tekintetében.

A bemutatott kutatások sorában az utolsó vizsgálat a COVID-19-cel kapcsolatos álhírek osztályozására fókuszált. A tanulmány részletesen ismertette az alkalmazott előfeldolgozási lépéseket és a modellek rétegszerkezetét. Az előfeldolgozási fázis során a szerzők eltávolították a szövegből az URL-eket, a speciális szimbólumokat és az írásjeleket, továbbá az emojikat és a számokat szöveges megfelelőjükké alakították. A stop szavak eltávolítása is része volt a folyamatnak, azonban a kutatás nem részletezte, hogy pontosan milyen stop szólistát alkalmaztak. A szövegek egységesítése lemmatizációval zárult. A tanulmány bemutatta az LSTM és GRU modellek architektúráját. Az LSTM modell egy beágyazási rétegből, két egymásra

épített LSTM rétegből, egy 25%-os dropout rétegből és három dense rétegből (128, 16, 1 neuron) állt. A GRU modell hasonló felépítésű volt, azonban három GRU réteget, egy további dropout réteget, valamint eltérő méretű dense rétegeket (200, 8, 1 neuron) tartalmazott. A kísérletek eredményei szerint a két modell teljesítménye gyakorlatilag megegyezett: az LSTM 92,38%-os, míg a GRU 92,89%-os pontosságot ért el, a többi értékelési mutató (precizitás, visszahívás, F1-score) pedig szintén minimális eltérést mutatott.

A bemutatott kutatások eredményeit a *2. táblázat* összegzi, lehetővé téve a különböző megközelítések és modellek teljesítményének közvetlen összehasonlítását.

2. Táblázat: *LSTM és GRU teljesítmények álhírek klasszifikációjában*

Forrás	Adatbázis	Modell	Pontosság	Precizitás	Recall	F1
Toor és mtsai. 2025	Politifact	LSTM	76,23%	75,43%		75,38%
		GRU	88,39%	87,54%		88,65%
Toor és mtsai. 2025	Buzzfeed	LSTM	77,36%	76,34%		76,14%
		GRU	87,21%	86,27%		86,67%
Rachmawati & Darmawan, 2024	Indonéz hírcikkek	LSTM	89%	89%	86%	88%
		GRU	90%	90%	87%	88%
Malhotra & Mahur, 2022	COVID-19 Fake News	LSTM	92,38%	92,95%	92,9%	92,9%
		GRU	92,89%	92,92%	92,9%	92,89%
Bajaj, 2017	2 Kaggle dataset	LSTM		93%	72%	81%
		GRU		89%	79%	84%

A kapott eredményekből az látható, hogy többségben a GRU modell ad pontosabb klasszifikációt, viszont ez mind a használt adatbázisoktól, az előfeldolgozási lépésektől és a modell architektúrájától is függ. Ezen és a korábban szereplő előfeldolgozási lépések irodalmainak eredményei alapján alkalmazom a legjobbaknak bizonyult módszereket.

Modellek korlátjai

Viszont, nem mondható el a különböző modellek teljesítményéről, hogy tökéletesek lennének. A legtöbb kutatás, mint a bemutattak is, csak azon az adatbázison tesztelték saját

modellük, amin be lettek tanítva, és általában nagy kihívást jelent számukra, hogy független korpuszra is általánosíthatóak legyenek az eredmények. Többen is megfogalmazzák, hogy az álhír detektálás, mint kutatási terület és az LSTM, GRU számára ez jelenti a legnagyobb kihívást, vagyis hogy a tanító modellen való túlilleszkedés elkerülése és az általánosíthatóság elérése más ismeretlen korpuszokra (Mridha és mtsai, 2021; Alnabhan & Branco, 2024; Mienye és mtsai, 2024; Toor és munkatársai, 2025). A túlilleszkedés ellen alkalmaztak (Bajaj, 2017; Camelia és mtsai, 2024; Malhotra & Mahur, 2022; Tahat és munkatársai, 2024; Zamir és mtsai, 2024) és javasolnak véletlenszerű kapcsolatkihagyásos (dropout) regularizációt (Srivastava és munkatársai, 2014) és L2 regularizátorokat (Lewkowycz & Gur-Ari, 2020; Mienye és mtsai, 2024) a modellekben.

Srivastava és munkatársai kifejtik (2014), hogy a dropout regularizáció hatékony módszer a túlilleszkedés csökkentésére mély neurális hálózatokban. Leírtakban szerepel, hogy a módszer lényege az, hogy a tanítás során véletlenszerűen eltávolít bizonyos neuronokat és azok kapcsolatait, így minden egyes tanítási iterációban egy „megritkított” hálózatot hoz létre. Ezáltal a hálózat nem tud túlzottan alkalmazkodni az egyes bemenetekhez, csökkentve a túlilleszkedés kockázatát és javítva az általánosító képességet. Viszont a bemutatott modellek (Bajaj, 2017; Camelia és mtsai, 2024; Malhotra & Mahur, 2022; Tahat és mtsai, 2024; Zamir és mtsai, 2024), amik alkalmazzák ezt a módszert, sem egy független korpuszon voltak alkalmazva, hanem a tanítón.

L2 regularizáció a túlilleszkedés ellen pedig úgy véd, hogy bünteti a túl nagy súlyokat a hálózatban. A veszteségfüggvényhez egy extra büntető tényezőt ad hozzá, amely korlátozza a súlyok nagyságát, ezáltal csökkentve a modell komplexitását. Ennek eredményeként a hálózat simább döntési határt alakít ki, kevésbé érzékeny a bemenetek apró változásaira, és jobban teljesít ismeretlen adatokon (Lewkowycz & Gur-Ari, 2020; Liu és mtsai, 2019).

Nagy kihívást jelent még továbbá megfelelő adatbázisok elérése és létrehozása (Alnabhan & Branco, 2024; Deepak & Chitturi, 2020). Ez azért van, mert nagyon idő és erőforrás igényes egy alapos, részletes leírással rendelkező nagy korpusz kialakítása pontos címkékkel ál és igaz hírek szempontjából.

A következőkben a kutatás során használt korpuszt ismertetem és az azon végrehajtott alakításokat, majd bemutatom a módszertant, beleértve az alkalmazott előfeldolgozási

lépéseket és a modellarchitektúrákat rétegszintű részletezéssel. Végül az elért eredmények kerülnek bemutatásra és értékelésre.

Adatbázisok

Az álhírek detektálására irányuló kutatások egyik meghatározó kihívása az alkalmazott adathalmazok minősége és elérhetősége. Korábbi kutatásokban leírják (Mridha és mtsai, 2021; Çetiner, 2024), hogy a neurális háló alapú modellek teljesítménye nagymértékben függ attól, hogy milyen adatokat használnak a tanítás és értékelés során. A manuálisan címkézett hamis híreket tartalmazó adatbázisok korlátozott hozzáférhetősége jelentős akadályt jelent a területen végzett kutatások számára, ami nemcsak az eredmények általánosíthatóságát nehezíti meg, hanem a modellek valódi hatékonyságának objektív értékelését is (Deepak & Chitturi, 2020). Emellett az adathalmazok potenciális torzítása is komoly problémát jelent, hiszen, ha az adatok nem megfelelően reprezentálják az általánosan előforduló híreket, akkor a modellek hajlamosak lesznek az adott torzítások megerősítésére, és nem biztosítanak megbízható detektálási, klasszifikációs teljesítményt (Alnabhan & Branco, 2024). Ennek ellenére csak kevés tanulmány fordít kiemelt figyelmet az adathalmazok minőségének biztosítására, ami hozzájárul ahhoz, hogy a hamis hírek felismerésére fejlesztett modellek gyakran alulteljesítenek (Mridha és mtsai, 2021). Ezekre a problémákra figyelve, én négy különböző korpusszal dolgoztam, kettő amit a tanítás során használtam és kettőt a tesztelésre.

Tanító korpusz

A tanításhoz használt ’ISOT Fake News Dataset[2]’ és a ’Misinformation & Fake News text dataset[3]’ két publikus korpusz ami a kaggle oldalán szabadon elérthető. Azért választottam és vontam egybe ezt a két nagy adatbázist ehhez a lépéshez, mert bizonyos szempontók szerint jól kiegészítik egymást, melyet a későbbiekben kifejtek. Továbbá, róluk találtam a legjobb leírást, ami bemutatja honnan vannak a megfigyelések és hogy mi alapján van besorolva valós vagy hamis hírnek.

[2] https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset
[3] https://www.kaggle.com/datasets/stevenpeutz/misinformation-fake-news-text-dataset-79k

ISOT Fake News Dataset

Ez a korpusz két külön fájlból adódik össze, a ’True’ és ’Fake’, melyekben a cikkek címe, tartalma, témája és dátuma figyelhető meg.

Az elsőben csak valós hírek szerepelnek, melyek mind a Reuters oldaláról származnak. Éppen ezért, nem csak ezt az egy korpuszt használom tanításra és egészítem ki, mivel lehet, hogy a modell csak a Reuters stílusát vagy rájuk egyedi jellemzőt tanulna be. Összesen 21 ezer cikket tartalmaz, melyek világi és politikai hírekkel foglalkoznak. A fájl cikkhossz-eloszlása a következő statisztikai jellemzőkkel írható le:

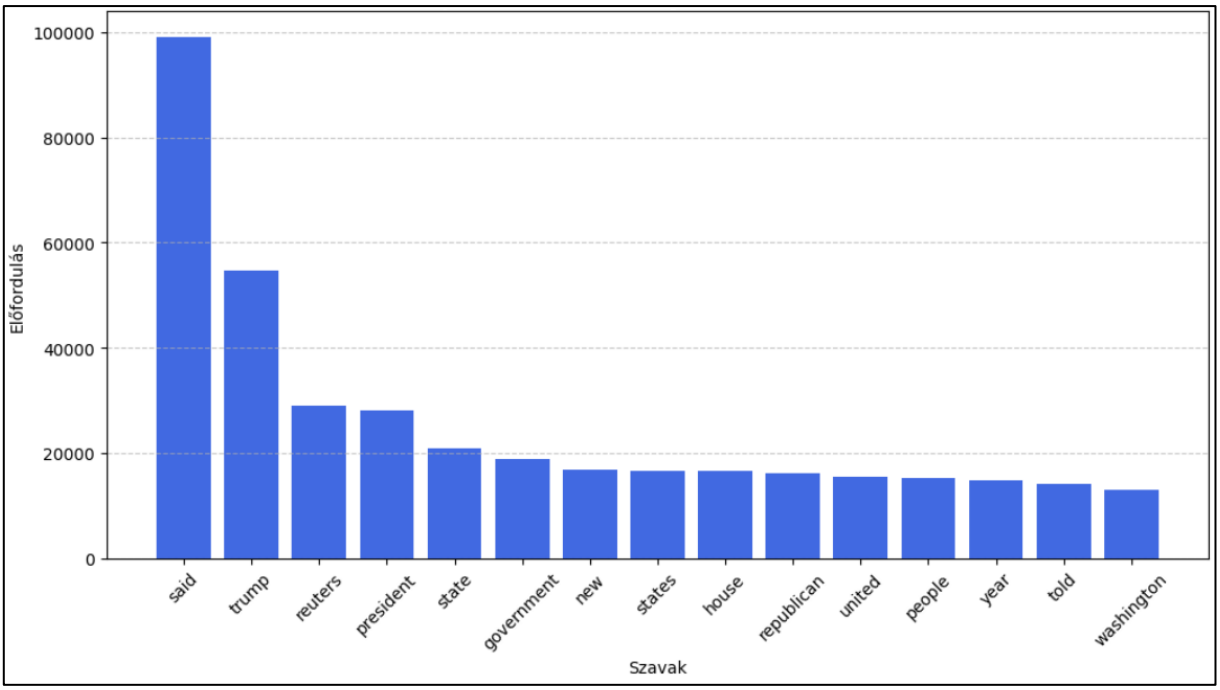
- Minimális hossz: 1 szó
- Maximális hossz: 29 781 szó
- Átlagos hossz: 2 383,28 szó
- Medián hossz: 2 222 szó
- Szórás: 1 684,84 szó

Az adatok alapján a cikkhosszok jelentős szórást mutatnak, amely arra utal, hogy az adatbázis tartalmaz mind rövid, mind rendkívül hosszú cikkeket.

A szóeloszlást vizsgálva stopszavak nélkül, az látható, hogy főleg az amerikai politikai hírek dominálnak, ami azért van mert a 2016-17 éveiben gyűjtötték az adatokat, ami az amerikai választással esett egy időbe.

A szógyakoriság elemzése során még kiemelendő, hogy a 15 leggyakoribb szó között szerepel a "reuters" és "washington". Ennek oka az, hogy szinte minden cikk a "reuters" megjelöléssel vagy az adott város nevével kezdődik és ellenkezőleg a másik szó a második. Korábbi kutatások, amelyek ezt az adatbázist vizsgálták (Camelia és mtsai, 2024; Pimpalkar és mtsai, 2021;), nem hangsúlyozzák ennek a jelenségnek a jelentőségét. Pedig ez lényeges szempont, mivel a korpusz alapján minden valódi hír tartalmazza ezeket az elemeket, míg az álhírek, mint később megfigyelhető, egyike sem. Ennek következményeként fennáll annak a veszélye, hogy a tanított modell túlzottan nagy jelentőséget tulajdonít ezeknek a szavaknak, és ez torzíthatja a klasszifikáció eredményét. Ezért az előfeldolgozás során különös figyelmet kell ennek szentelni.

4. Ábra: ISOT adatbázis igaz híreinek szóeloszlása stopszavak nélkül

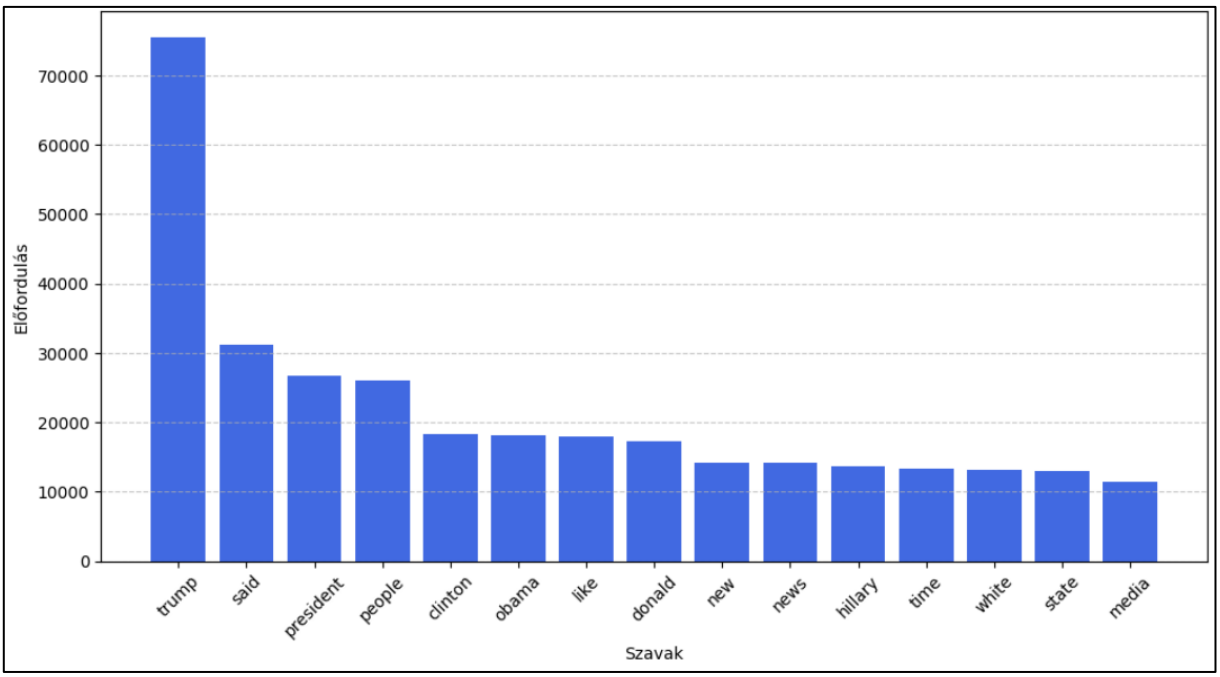


Az álhíreket tartalmazó fájl hat különböző témát tartalmaz: kormányzati, közel-keleti, amerikai, bal oldali, politikai, és általános hírek. Összesen 23 ezer cikk tevődik különböző forrásokból össze. Ezek a források megbízhatatlan oldalakról származnak, melyeket a Politifact nevű tényellenőrző amerikai szervezet sorolt be annak. Az álhírek statisztikai mutatói a következők:

- Minimális hossz: 1 szó
- Maximális hossz: 51 794 szó
- Átlagos hossz: 2 547,40 szó
- Medián hossz: 2 166 szó
- Szórás: 2 532,88 szó

Az álhírek esetében a cikkhosszok átlaga (2 547,40 szó) kissé magasabb, mint a valódi hírek esetében (2 383,28 szó), azonban a medián érték kisebb (2 166 vs. 2 222 szó). Ez arra utal, hogy bár az álhírcikkek között előfordulnak hosszabb tartalmak, a cikkek többsége valamivel rövidebb, mint a valódi híreké. A szórás lényegesen nagyobb az álhírek esetében (2 532,88 szó vs. 1 684,84 szó), ami a hosszúság tekintetében nagyobb változékonyságot jelez. Ez különösen a maximális hosszértékeknél figyelhető meg.

5. Ábra: *ISOT adatbázis álhíreinek szóeloszlása stopszavak nélkül*



A szavak gyakoriságánál, ugyanúgy az amerikai választásra utaló szavakat lehet leginkább észrevenni. Kiemelendő még az álhíreknél, hogy bizonyos kifejezések többségben csak itt jelennek meg, mint a ’reuters’ a valós híreknél. Ezek a szavak ebben a korpuszban a következők: 'featured image', 'photo by', 'getty images[4]'. Megjelenésük az álhírekben azzal magyarázható, hogy a cikk scrapel-és során a képek forrása is megmaradt, amelyek gyakran stock fotókhoz vezettek vissza.

Misinformation & Fake News

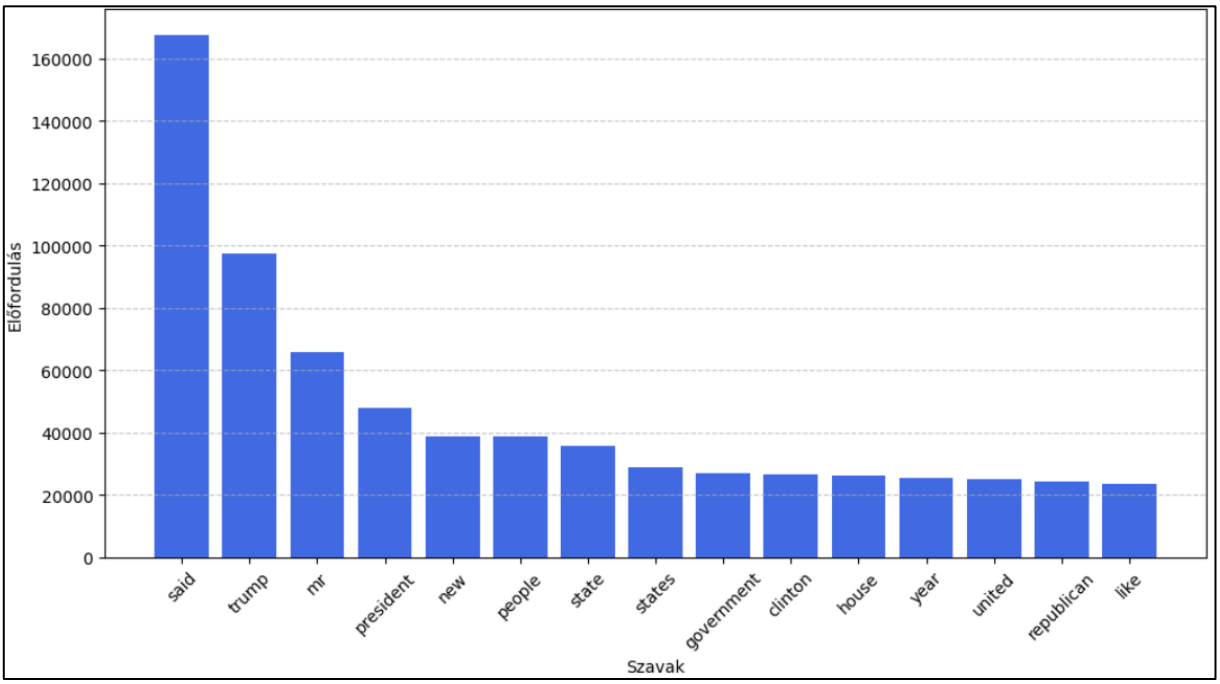
A második korpusz ami kombinálva lesz az ISOT-al, az Misinformation & Fake News text dataset. Ez egy átfogó adathalmaz, összesen 79 000 cikket tartalmaz félrevezető, álhír vagy propagandatartalomként kategorizált forrásokból. Az adathalmaz korábbihoz hasonlóan két

[4] Nagy képi könyvtárral rendelkező stock fotó oldal: https://www.gettyimages.com/

fájlra oszlik, a valódi hírek, 35 ezer, és a félrevezető, álhír tartalmú cikkek, 43 ezer. Az adatbázis összeállítása során minden egyéb metaadatot eltávolítottak, így az egyes cikkek kizárólag a szövegtartalmukat tartalmazzák.

A valódi hírek különböző megbízhatóként ismert forrásokból származnak, mint The New York Times, The Washington Post illetve a Reuters. Utóbbi miatt átfedés lesz az ISOT korpusszal, melyre az egybevonásnál figyelni kell.

6. Ábra: *Misinformation & Fake News adatbázis igaz híreinek szóeloszlása stopszavak nélkül*



A 15 leggyakoribb szónál, hasonlóan az amerikai 2016 választási hírek dominanciája vehető észre a Trump, Clinton és goverment szavak gyakorisága miatt.

A cikkek statisztikai jellemzői a következők:

- Sorok száma: 34 975
- Minimális hossz: 3 szó
- Maximális hossz: 85 948 szó
- Átlagos hossz: 3 221,17 szó
- Medián hossz: 2 394 szó
- Szórás: 3 337,73 szó

A valódi hírekből álló korpusz statisztikai elemzése alapján megállapítható, hogy a szövegek jelentős eltéréseket mutatnak hosszúság tekintetében. A maximális hossz (85 948 szó) kiugró

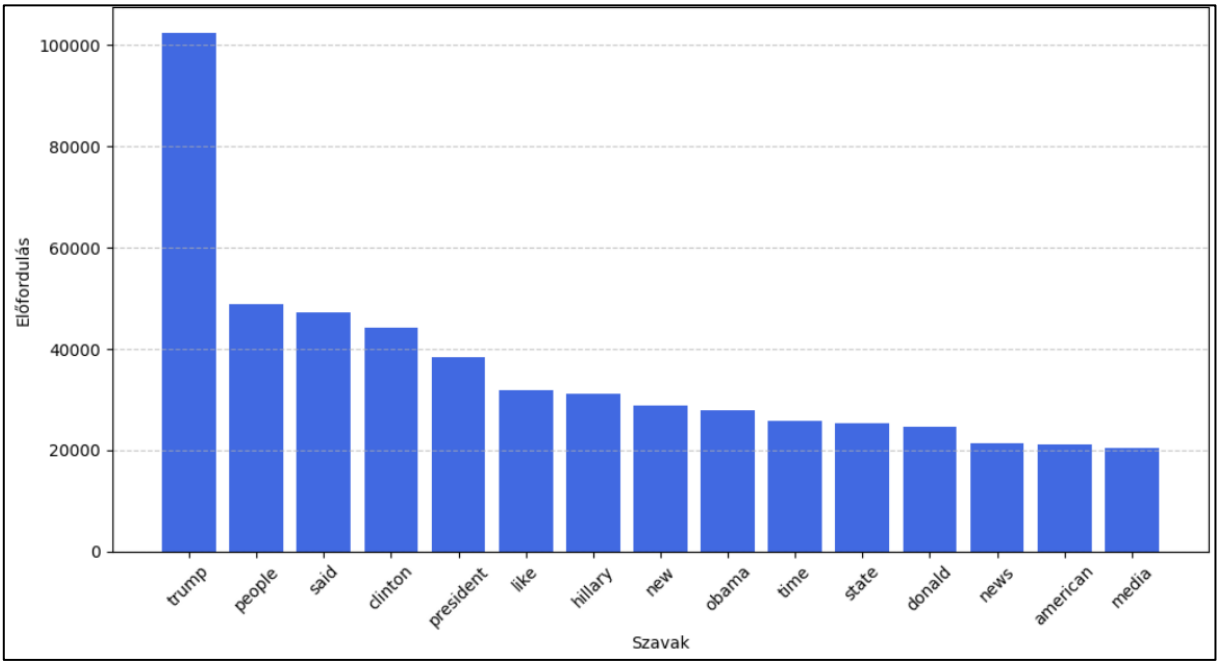
értékként kezelhető, amely arra utal, hogy az adathalmaz tartalmazhat rendkívül hosszú elemzéseket vagy több részből álló cikkeket. Az átlagos hossz (3 221,17 szó) és a medián hossz (2 394 szó) közötti különbség arra utal, hogy az eloszlás aszimmetrikus, és a hosszabb szövegek felfelé torzíthatják az átlagértéket.

Az álhírek és félretájékoztatást tartalmazó hírek hasonlóan több forrásból kerültek összeállításra. Míg az ISOT-nál az álhírek közt bal oldali hírek szerepeltek, itt szélsőjobboldali amerikai weboldalakról (pl. Redflag Newsdesk, Breitbart, Truth Broadcast Network) származó hírek jelennek meg. Illetve még egy korábban publikált adathalmaz, amelyet Ahmed, Traore és Saad (2017) dolgoztak fel és ismertettek az ISDDC 2017 konferencián, amelyben ismét a Politifact által azonosított álhírek szerepelnek. Végül, az adathalmaz tartalmazza még az EUvsDisinfo projekt által azonosított dezinformációs és propagandatartalmakat is. Ez a projekt 2015 óta figyeli és ellenőrzi azokat az álhíreket, amelyek a Kreml-barát médiából származnak és az Európai Unió területén terjednek. Az így összeadódott 43 ezer cikk 15 leggyakrabb szavai a *7. ábrán* látható.

Az álhírek statisztikái az alábbiak:

- Sorok száma: 43 642
- Minimális hossz: 1 szó
- Maximális hossz: 142 961 szó
- Átlagos hossz: 2 637,24 szó
- Medián hossz: 1 949 szó
- Szórás: 3 880,36 szó

7. Ábra: *Misinformation & Fake News adatbázis álhíreinek szóeloszlása stopszavak nélkül*



Az álhírek és félretájékoztató tartalmak általában rövidebbek, mint a valódi hírek. Az átlagos hossz (2 637,24 szó) alacsonyabb, mint a valódi hírek esetében (3 221,17 szó), és a medián érték is kisebb (1 949 vs. 2 394 szó). Ez arra utal, hogy az álhírek általában tömörebbek, és kevesebb terjedelemben közvetítik az információkat. Kiemelendő még az, hogy a Misinformation & Fake News dataset olyan európai híreket is tartalmaz, amelyek révén az adatbázis nemzetközibbé válik, ezáltal csökkentve az angolszász médiára korlátozódó torzításokat. Mindezek az intézkedések hozzájárulnak a modell bias-csökkentéséhez és általánosíthatóságának növeléséhez, amely kulcsfontosságú a félretájékoztatás széles körű felismerésében.

Kombinált tanító korpusz

Az ISOT és a Misinformation & Fake News text dataset együttes használata azért fontos a kutatás során, mivel így egy kiegyensúlyozottabb és általánosíthatóbb korpusz áll rendelkezésre a modell betanításához. Az ISOT adatbázis valódi hírei kizárólag a Reuters hírügynökségtől származnak, ami önmagában korlátozhatja a modell teljesítményét, hiszen a nyelvi sajátosságok és a cikkstruktúra homogén mintázatokat eredményezhetnek. Ennek ellensúlyozására a Misinformation & Fake News datasetben megtalálható valódi hírek is beépítésre kerültek, amelyek olyan forrásokból származnak, mint a The New York Times és a The Washington Post, ezáltal növelve a hírforrások diverzitását. Emellett, az adathalmazok

egyesítésének további előnye, hogy az álhírek politikai spektruma szélesebb skálán mozog. Míg az ISOT adatbázis elsősorban baloldali álhíreket tartalmaz, addig a Misinformation & Fake News dataset jobboldali amerikai és európai dezinformációs tartalmakat is magában foglal. Ez biztosítja, hogy a modell ne csupán egy adott politikai narratívát tükröző álhírmintázatokat tanuljon meg, hanem képes legyen szélesebb körben azonosítani a félretájékoztatás különböző formáit.

A két adatbázis együttes alkalmazása nemcsak az álhírek tartalmi változatosságát növeli, hanem jelentősen kibővíti az adathalmaz méretét is. A korpuszok egyesítésével összesen 123 ezer cikk állt rendelkezésre, azonban ebből 30 ezer duplikált tétel eltávolításra került, így a végső adathalmaz 55,7 ezer valódi hírt és 37,8 ezer álhírt tartalmaz. Ezáltal a modell tanításához egy jelentős méretű, torzításoktól mentesebb és politikai szempontból kiegyensúlyozottabb adathalmaz jött létre. Ezt az együttes korpuszt az előfeldolgozás nulladik lépéseként megtisztítottam, hogy eltávolítottam minden üres és nem angol megfigyelést, majd kiegyensúlyoztam a valós-álhír arányát, legvégül 80-20%-ba szétosztottam tanító és teszt adathalmazra.

Független adathalmazok

Ahhoz, hogy a modellem általánosító képességét tudjam tesztelni, egy olyan korpusz próbáltam összeállítani, ami tartalmilag eltér a tanító adatbázis szövegeitől. Sajnos korlátozott számban elérhetőek, olyan jól definiált, előre klasszifikált korpuszok, melyek nincsenek szinte teljes átfedésben a korábban már használt adatokkal, vagyis az amerikai politikával. Részben sikerült találni ez alapján két megfelelő korpuszt, amelyeket kombinálok, először is egy fülöp-szigeteki angol nyelvű híreket tartalmazó fájlt, ami igaz és álhírekből tevődik össze, illetve egy, csak hamisokból állót, ami a BS_detector nevű program sorolt be azoknak.

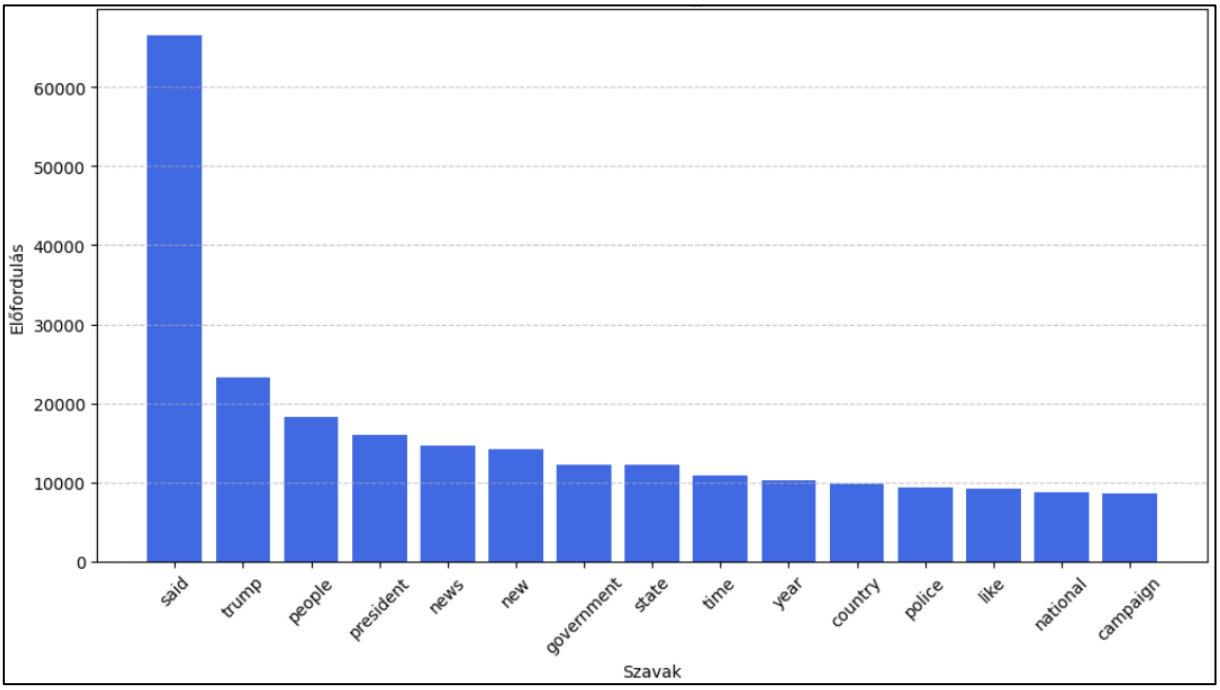
Fülöp-szigeteki angol hírek

A fülöp-szigeteki angol hírek[5] összegyűjtésének módjáról sajnos a GitHub oldalán nincsen pontos leírás, hogy milyen időszakot ölel fel, mi alapján lettek besorolva igaz, vagy álhírnek. Viszont manuális átnézés alapján azt lehet elmondani, hogy a 14 ezer igaz hír, többségben Reuters által írt Fülöp-szigeteken történtekről szólnak, míg a 3 ezer álhír főként a 2016-os

[5] https://github.com/francheska-vicente/data102-fake-news

választás körülményeiről íródtak. A korpuszban a metaadatok nem szerepelnek, csak a címke, hogy milyen besorolású a hír, illetve a nyers szöveg. A részletesebb átnézése során több nem angol nyelvű cikket találtam, erre a későbbi előfeldolgozásnál, figyelni kell, hogy megfelelően ki legyenek szűrve. Összesen 17521 adat szerepel, melyek 15 leggyakoribb szava a következőek, miután a stop- és a nem angol szavakat kiszűrtem:

8. Ábra: *Fülöp-szigeteki angol hírek adatbázis szóeloszlása stopszavak nélkül*



A leggyakrabb szavak közt itt is megjelenik a Trump és pár különböző választásokra utaló jel, viszont csak kisebb mennyiségben. A korpusz részletesebb átnézése során több nem angol nyelvű cikket találtam, ezért a későbbi előfeldolgozásnál, erre is figyelni kell.

BS Detector korpusz

Mivel az eloszlása túlságosan dől az igaz hírek fele, 82%-a az, ezért egy olyan publikusan elérhető adatbázissal egészítettem ki[6], ami csak álhíreket tartalmaz, 12999 darabot. Ez a gyűjtemény 244 különböző weboldalról származó álhíreket tartalmaz. A weboldalak kategorizálása, ahonnan a cikkek származnak a BS Detector[7] jelölései alapján történt, amely egy olyan eszköz, amely azonosítja a félrevezető és manipulált híroldalakat. Azok az oldalak, melyek cikkjei ebben az adatbázisban szerepelnek a „bs” (bullshit) címkével voltak ellátva. A

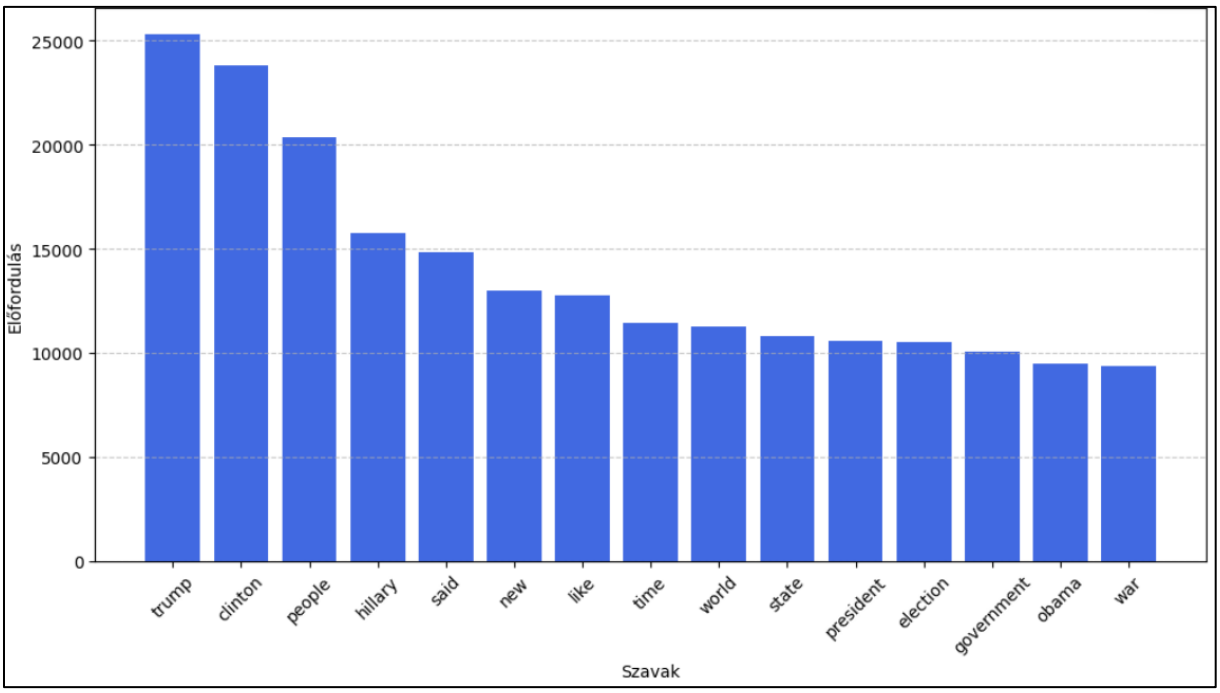
[6] https://www.kaggle.com/datasets/mrisdal/fake-news
[7] https://github.com/selfagency/bs-detector

szöveg mellett még különböző adatok vannak, mint a szerző, a publikálás dátuma, cím és még több, de számomra ezek nem relevánsak. A dátum még azért lehet fontos, mert ebből derül csak ki, hogy az adatok itt is 2016-ban lettek gyűjtve, ami a szavak eloszlásán is látni.

A korpusz statisztikai jellemzői alapján megfigyelhető, hogy a cikkhosszok jelentős eltéréseket mutatnak:

- Minimális hossz: 1 szó
- Maximális hossz: 142 961 szó
- Átlagos hossz: 3 870,96 szó
- Medián hossz: 2 373 szó
- Szórás: 5 661,38 szó

9. Ábra: *FAKE adatbázis szóeloszlása stopszavak nélkül*



Kombinált független korpusz

A két adatbázis kombinálását követően, az előfeldolgozási lépések előtt összesen 30 520 cikk szerepelt a korpuszban. Az angol nyelvű Fülöp-szigeteki hírekkel történő egyesítés után eltávolítottam az átfedéseket a két fájl között, majd ugyanezt az ellenőrzést elvégeztem a két tanító adatbázis összevont fájlára. Az ismétlődések kiszűrése és tisztítás után a végső korpusz

19 478 cikket tartalmaz, amelyek közül 12 326 valódi hír, míg 7 152 álhír, amit szintén kiegyensúlyoztam az előfeldolgozás előtt.

2025 márciusi korpusz

A modellek hasznosíthatósága és általánosítása miatt, úgy vélem, hogy nem elég, hogy csak korábbi, több éves korpuszokkal való teljesítmény ellenőrzést végzek. Emiatt, manuálisan gyűjtöttem 20 igaz és 20 álhír cikkeket, mind 2025 március hónapból, hogy a lehető legfrissebb cikkekkel is ellenőrizni tudjam, így a modellek időbeliségének ellenőrzését is el tudom végezni.

A valós híreket széles körben megbízható adatforrásnak jelölt oldalakról gyűjtöttem, mint a DeutcheWelle, APNews, és a korábbiakban is használt Reuters. Ezek a cikkek lefednek politikai hírek mellett, pénzügyi, közéleti és sport híreket is. Az álhíreket a Politifact[8] által hamisnak ítélt cikkei közül válogattam ki. Ezek főként amerikai szélsőséges hírügynökségek írásai, illetve orosz háborús propaganda, mint például, hogy „Németország afgán bevándorlókat toboroz zsoldosoknak Ukrajna számára”[9].

A következő fejezetben bemutatom a kutatás során alkalmazott módszertant, részletezve az előfeldolgozási lépéseket, amelyek biztosítják a szövegek megfelelő reprezentációját a mélytanulási modellek számára. Ezt követően ismertetem az LSTM és GRU modellek felépítését és működését, amelyek segítségével a valódi és álhírek közötti osztályozást végzem.

Módszertan

Ebben a fejezetben először ismertetem az adat előfeldolgozás öt eltérő folyamatát: az első lépést részletesen, lépésről lépésre, majd bemutatom azokat az 1-1 módosításokat, amelyeket a későbbi fázisok során végrehajtok. Ezt követően bemutatom a kutatás során alkalmazott mélytanulási modelleket, az LSTM és GRU architektúrák felépítését és a modellparaméterek megválasztását. Végezetül meghatározom azokat az összehasonlítási szempontokat, amelyek alapján az alkalmazott modellek teljesítményét értékelem, részletezve ezek jelentőségét és

[8] https://www.politifact.com/
[9] https://www.rt.com/news/614351-source-tells-rt-germany-afghan-migrants-ukraine/

relevanciáját a kutatás szempontjából. Az elemzés során használt teljes kód elérhető a GitHub oldalamon[10].

Előfeldolgozási alapok

Az előfeldolgozás során, a korábbi fejezetekben ismertetett lépéseket hajtottam végre minimális kiegészítéssel. A szöveges korpusz előfeldolgozása során a következő Python könyvtárak alkalmazása volt a legfontosabb: az NLTK[11] (Natural Language Toolkit) (Bird és mtsai, 2009) a szótövezés, a szóalapú tokenizáció és a morfológiai elemzés (POS tagging) feladataiban; a spaCy[12] (Montani & Honnibal, 2018) a stop-szavak kezelésére; valamint a TensorFlow[13] (Abadi és mtsai, 2016) a szekvenciahosszok egységesítésére (padding és truncation) és a numerikus tokenizációra majd később a modellek kialakítására. Mindegyik előfeldolgozás során, először egyenlő számra hoztam az igaz és álhírek számát azáltal, hogy véletlenszerűen eltávolítok a nagyobb csoportból, vagyis az igaz cikkekből annyit, hogy számuk megegyezzen az álhír cikkekkel, majd kiszűrtem az üres és nem angol cikkeket, továbbá kisbetűssé alakítottam a teljes korpuszt. Ezt követően a regular expressions segítségével átalakítottam a számokat *numtoken*-né, majd eltávolítottam a különböző zajokat, mint a dátumok és telefonszámok, html és url-ek illetve a nem alfanumerikus elemek. Az alfanumerikus elemekre szűrés előtt az angol nyelvi összevonásokat, mint a például *don't* és az *I'm* kifejezéseket szétbontottam, *do not* és *I am* szavakra, illetve a cikkeket mondatokra szegmentáltam, majd szavakként tokenizáltam, így a cikkek mondatok listája lett, melyeken belül szavak listája szerepel. Ezeket a lépéseket mindegyik előfelkészítés során elvégeztem, illetve egy 0. módszer esetében, ezenkívül egyéb alakítást nem végeztem, hogy így biztosítani tudjak egy alapvető viszonyítási pontot a többi, komplexebb előfeldolgozási eljárás hatékonyságának értékeléséhez. Tovább folytatva az 1. módszer bemutatását, a következőkben a *spaCy* stopszavak és a POS tag segítségével eltávolítottam a felesleges szavakat, míg a szakirodalom szerint, álhírek szóhasználatára jellemző szófajokat: névmásokat, határozószavakat és főneveket, a szövegben hagytam. Ezen felül kivettem a korpuszokból a számokból és betűkből álló vegyes, a kettő vagy annál ritkábban előforduló, illetve a *reuters* és *washington* szavakat. Lemmatizálás volt a következő lépés, ami a POS tag

[10] https://github.com/Konye07/Konye-MscCode
[11] https://www.nltk.org/
[12] https://spacy.io/
[13] https://www.tensorflow.org/

segítségével ment végbe. Majd utolsó lépésekben, lekértem az eddig kialakult korpusz statisztikáit, és ez alapján megadtam a maximális hosszt cikkekhez, ami alapján ki lettek párnázva vagy levágva. Majd végül a szavak numerikus tokenizáláson mentek keresztül és vektorizálva lettek a GloVe segítségével.

Ezt a leírt folyamatot módosítottam négy különböző módon.

2. Bent tartottam minden stopszót
3. Kivettem minden stopszót
4. Stemming lemmatizálás helyett + legjobban teljesítő stopszó módszer
5. Számok átalakítása szöveggé (*5* → *five*) + legjobban teljesítő stopszó módszer

Így végül hat különböző előfeldolgozáson keresztülment korpuszom lett, amelyen a következőben leírt modelleket tanítottam és teszteltem.

Modell architektúrák

Az LSTM és GRU modelljeim, ugyanolyan paraméterekkel rendelkeznek, hogy az összehasonlítás a lehető legpontosabb legyen, ezért a felépítést az LSTM struktúráján keresztül mutatom be. Tensorflow Keras[14] csomag segítségével készültek.

Először a modell bemeneteként a tokenizált szöveg kerül feldolgozásra, amely az előfeldolgozási lépések eredményeként jött létre. Ezt követi egy 300 dimenziós GloVe szóbeágyazási réteg, amely a szavakat sűrű vektortérbe ágyazza, megőrizve azok szemantikai kapcsolatait. A szóbeágyazási réteg után a bemenet egy három rétegű rekurrens neurális hálózaton halad végig. Az első LSTM-réteg 128 neuront tartalmaz, és visszaadja a teljes szekvencia kimenetét, amelyet egy dropout réteg követ. A második LSTM-réteg 64 neuronnal működik, szintén megtartva az időlépésenkénti kimeneteket, és ezt is külön dropout réteg követi. A harmadik LSTM-réteg már csak 32 neuront tartalmaz, és csak a végső rejtett állapotot adja tovább a következő rétegnek. Minden LSTM-rétegben belső dropout mechanizmus is alkalmazásra kerül (0,3 arányban), amely az LSTM cellák bemeneti és visszacsatolási kapcsolataira vonatkozik. Ez a rétegen belüli és rétegek közötti dropout kombináció segíti a hálózat regularizációját, és jelentősen csökkenti a túlilleszkedés (overfitting) kockázatát (Srivastava és mtsai, 2014). A kimeneti réteg egyetlen neuront tartalmazó, teljesen

[14] https://www.tensorflow.org/guide/keras

csatlakoztatott (dense) réteg, amely sigmoid aktivációs függvényt alkalmaz a bináris osztályozási feladat megoldására, azaz az álhír felismerésére.

Összességében tehát a modell egy előzetesen betanított szóbeágyazási réteggel indul, amely lehetővé teszi a szavak jelentésének megfelelő vektortérbeli ábrázolását. Ezt követi egy többrétegű rekurrens hálózat (LSTM/GRU), amely képes felismerni a szöveg hosszú távú összefüggéseit, majd egy teljesen csatlakoztatott kimeneti réteg, amely bináris osztályozást végez az álhírek detektálására.

A modell tanítása során az Adam (Adaptive Moment Estimation) optimalizációs algoritmust alkalmaztam, ami az irodalomban széles körben használt alapértelmezett tanulási rátákkal (Pimpalkar és mtsai, 2021; Airlangga, 2024; Çetiner, 2024 ; Bajaj, 2017; Tahat és mtsai, 2024; Rachmawati & Darmawan, 2024). Számos klasszifikációs kutatás igazolja, hogy az Adam az egyik leghatékonyabb optimalizálási módszer az osztályozási feladatok esetén (Anjana és mtsai, 2019; Hossain és mtsai, 2020). Hatékonyságát annak köszönheti, hogy ötvözi a momentum-alapú és az adaptív tanulási rátát alkalmazó megközelítések előnyeit. Az algoritmus a gradiens első és második momentumát is figyelembe veszi, így gyorsabb és stabilabb konvergenciát biztosít a hagyományos gradiens-alapú eljárásokhoz képest (Kingma & Ba, 2014).

A veszteségfüggvényként bináris keresztentrópiát (binary crossentropy) alkalmaztam, amely a valódi címkék és a modell által előrejelzett valószínűségek közötti eltérést méri, és különösen alkalmas bináris kimenetű osztályozási problémák esetén.

Az Adam optimalizáló és a bináris keresztentrópia veszteségfüggvény együttesét a szakirodalom rendszerint összehangoltan alkalmazza, mivel számos empirikus vizsgálat igazolta, hogy ez a kombináció stabil és hatékony tanulást eredményez (Airlangga, 2024; Rachmawati & Darmawan, 2024; Das és mtsai, 2020, Jose és mtsai, 2021). Ezt a bevett szakmai gyakorlatot követve döntöttem a fentiek alkalmazása mellett.

A tanítás során 64-es batch size-ot alkalmaztam korábbi álhírekkel foglalkozó kutatásokhoz hasonlóan (Pimpalkar és mtsai, 2021; Deepak & Chitturi, 2020; Tahat és mtsai, 2024). Továbbá a modelltanítás során a tanulási folyamatot 10 epizódra maximalizáltam, amely tapasztalataim alapján elegendőnek bizonyult ahhoz, hogy a hálózat hatékonyan megtanulja az adatokban rejlő mintázatokat, ugyanakkor elkerülje a túlilleszkedést. A megfelelő epizódszám

megválasztása érdekében előzetes kísérletként egy 100 epizódos tanítást is lefuttattam mind az LSTM, mind a GRU modell esetében. Ezek az eredmények egyértelműen azt mutatták, hogy a validációs veszteség már viszonylag korán, néhány epizód után, növekedni kezd, ami arra utal, hogy a modellek rövid tanítási idő alatt elérik optimális teljesítményüket, hosszabb tanítás esetén viszont hajlamosak a túltanulásra. Ezért a végleges modellképzés során a validációs veszteség alakulását folyamatosan nyomon követtem, és a túltanulás elkerülése érdekében early stopping technikát alkalmaztam, ami lehetővé tette, hogy a tanulási folyamat automatikusan leálljon, amikor a modell már nem mutat további javulást a validációs adatokon, így biztosítva az általánosítási képesség maximalizálását.

Mindezt, vagyis modell tanítását és értékelését, Google Colab környezetben végeztem, ahol az ingyenesen elérhető GPU-erőforrásokat használtam a számítási teljesítmény optimalizálása érdekében. A Colab lehetőséget biztosít a NVIDIA Tesla K80, T4 vagy P100 GPU-k igénybevételére, amelyek jelentősen felgyorsítják a mély tanulási modellek, különösen az RNN-alapú architektúrák tanítását.

Összehasonlítási metrikák

A modellek teljesítményének kiértékelését és összehasonlítását a korábbi táblázatokban bemutatott kiértékelési szempontok alapján végzem. Mindegyik modellt **öt alkalommal** futtatom le, majd a különböző metrikák (*pontosság, precizitás, recall, F1-score*) **átlagát** és **szórását** mutatom be. Ez a megközelítés azért indokolt, mert a mély tanulási modellek a tanulási folyamat sztochasztikus természete miatt jelentős teljesítményingadozás figyelhető meg. Az egyszeri értékelés így nem adna megbízható képet a modell általánosíthatóságáról. Az átlagos teljesítmény a modell várható viselkedését, míg a szórás annak robusztusságát jellemzi, ezáltal a többszöri futtatás statisztikailag megalapozottabb képet nyújt.

Továbbá, abból az okból kifolyólag használok több mérőszámot, mert a klasszifikációs feladatok esetén az önmagában vett pontosság nem mindig tükrözi pontosan a modell teljesítményét, különösen kiegyensúlyozatlan adathalmazok esetén. Ennek megfelelően az értékelés során bemutatom a *pontosság*, *precizitás*, *recall* és *F1-score* értékeit, melyeket a hasonló célú álhírelemzésekben is hangsúlyosan alkalmaznak (Çetiner, 2024; Mridha és mtsai, 2021). Kiegészítésként a konfúziós mátrix és a félreklasszifikált példák tartalmi jellemzői is ismertetésre kerülnek, amellett hogy hisztogram segítségével bemutatom a modellek egyes

cikkeinek klasszifikációra vonatkozó valószínűségi eloszlását. A konfúziós mátrixok esetében is az átlagot és a szórást, a hisztogramok esetében pedig az 5 futás aggregált eredményeit fogom majd bemutatni.

Az első ilyen értékelési szempont a **pontosság** (*accuracy*), amely százalékos formában mutatja meg, hogy a modell az összes predikcióhoz viszonyítva milyen arányban hozott helyes döntést. Kiszámolásának módja:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

ahol TP a helyesen azonosított pozitív esetek (*true positives*), TN a helyesen azonosított negatív esetek (*true negatives*), FP a tévesen pozitívként azonosított esetek (*false positives*), és FN a tévesen negatívként azonosított esetek (*false negatives*).

A **precizitás** (*precision*) azt méri, hogy a modell által pozitívnak jelzett esetek közül valójában hány volt ténylegesen pozitív.

$$Precision = \frac{TP}{TP + FP}$$

A **recall**, vagy szenzitivitás azt jelzi, hogy a ténylegesen pozitív esetek mekkora részét sikerült helyesen azonosítani:

$$Recall = \frac{TP}{TP + FN}$$

Az **F1-mutató** (F1-score) a precizitás és a recall harmonikus átlaga, amely különösen hasznos kiegyensúlyozatlan osztályeloszlású adathalmazok esetén. Az F1-score figyelembe veszi mind a hamis pozitív, mind a hamis negatív eseteket, ezáltal kiegyensúlyozottabb értékelést nyújt a modell teljesítményéről:

$$F1 = 2\frac{Precison \times Recall}{Precison + Recall}$$

Az F1-score különösen elterjedt metrika az álhírek detekciójához hasonló feladatokban (Mridha és mtsai, 2021), ahol az osztályok közötti egyensúlyhiány gyakori, és mindkét típusú hiba (FP és FN) fontos tényező.

Legvégül az eredmények **konfúziós mátrixai** részletes képet adnak arról, hogyan teljesít a modell az egyes osztályok szintjén. A mátrix minden sora a tényleges osztályokat, míg az oszlopok a modell által becsült osztályokat jelölik. Ezáltal jól láthatóvá válik, hogy mely osztályokat tévesztette össze a modell, illetve mely osztályok esetén érte el a legjobb teljesítményt.

Eredmények

Az elért eredményeket a teszthalmazok sorrendjét követve mutatom be. Elsőként a tanító adathalmazból leválasztott tesztkészletre vonatkozó eredményeket ismertetem, ezt követően a kombinált, független teszthalmaz, végül pedig a manuálisan összeállított, 2025 márciusában gyűjtött teszthalmaz elemzése következik. A részletes eredmények rendre a *3.*, *4.* és *5.* táblázatban találhatók, ahol az egyes modellkonfigurációk ötszöri futtatása során kapott átlagos teljesítménymutatók kerültek feltüntetésre, melyek mellett zárójelben az adott eredmények szórásai láthatóak. Továbbá a táblázatokban a legjobb eredmények zöld színnel és félkövér betűstílussal kerültek kiemelésre, míg a leggyengébb teljesítményeket piros szín jelöli, elősegítve ezzel az eredmények közötti gyors vizuális eligazodást a könnyebb összehasonlítás érdekében.

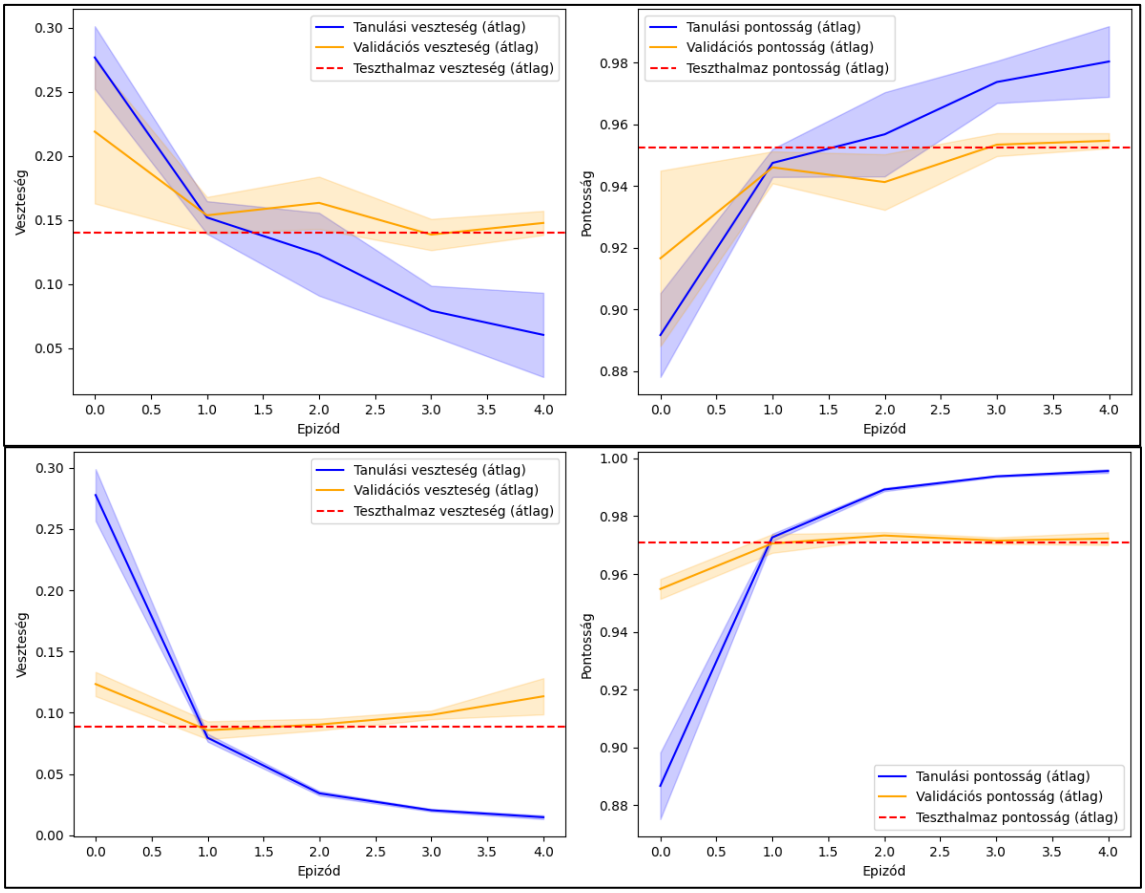
Eredmények a tanítóból leválasztott teszthalmazon

Első eredményként a tanító adathalmazból leválasztott teszthalmaz eredményeit mutatom be, a *3.* táblázat segítségével. A tanulási folyamatok jellemzőinek szemléltetésére a *9. ábrán*, a későbbiekben részletesebben bemutatott legrosszabb (LSTM) és legjobb (GRU) modelleken keresztül mutatom be a tanulási és validációs veszteség, valamint pontosság epizódonkénti alakulását. Mint korábban bemutattam, a modellek tanítását egységes beállítások mellett végeztem, vagyis a maximális epizódszámot 10-ben határoztam meg, és 3 epizódos early stopping (korai leállítás) mechanizmust alkalmaztam. A görbék minden esetben öt különálló futás átlagát jelenítik meg, miközben az árnyékolt terület a szórást mutatja, jelezve az egyes tanítási próbálkozások közti variabilitást.

A *9. ábra* felső részén a leggyengébben teljesítő LSTM modell eredményei láthatók. Bár a tanulási veszteség folyamatosan csökkent, a validációs veszteség jelentős ingadozást mutatott, és nem alakult ki stabil konvergencia. Hasonló instabilitás figyelhető meg a

pontossági görbén is: a tanulási pontosság javult, de a validációs értékek hullámzóak maradtak, ami túlilleszkedésre utal. A modell tehát nem tudott megbízhatóan általánosítani, és az early stopping az ötödik epizód környékén aktiválódott. Ezzel szemben az alsó ábrákon látható GRU modell egyenletesebb és robusztusabb tanulást mutatott. A veszteséggörbék szorosan követték egymást, a szórás minimális volt, és a pontossági mutatók is stabil emelkedést jeleztek. A GRU modell már néhány epizód után elérte legjobb teljesítményét, a korai leállítás pedig időben és hatékonyan lépett életbe.

9. Ábra: *Tanulási és validációs veszteség és pontosság, legrosszabb LSTM felül, legjobb GRU alul*



Mindezek alapján, a GRU architektúra gyorsabb és megbízhatóbb konvergenciát biztosított, míg az LSTM hajlamosabb volt a túltanulásra. Mivel a későbbiekben bemutatott teszthalmazokon is hasonló mintázatok jelentkeztek, a modellek korán elérték az optimális teljesítményt, és a tanulás rövid időn belül lezárult, a továbbiakban az ilyen jellegű ábrák bemutatását mellőzöm, és csupán szöveges hivatkozásokat alkalmazok.

Ezt követően, az előfeldolgozási módszerek összehasonlító elemzése során megfigyelhető a *3. táblázat* mutatói alapján, hogy azon előfeldolgozási módszerek közül, amelyek a

stopszavak kezelésében különböztek, a **legjobb klasszifikációs teljesítményt** azok a változatok nyújtották, amelyben valamennyi **stopszó megtartásra** került a korpuszban. Ezek közül is a nulladik, csak normalizálásnak alávetett korpusz modell eredményei teljesítettek legjobban. A **legalacsonyabb eredmények az összes stopszó eltávolításával** járó előfeldolgozási beállítás esetén jelentkeztek. Ennek megfelelően a negyedik és ötödik előfeldolgozási módszer esetében, mindhárom teszthalmazon, a stopszavak megtartása mellett végzem a további kísérleteket. Ez alapján, az is megállapítható, hogy az a módosítás, amelyben a lemmatizálást stemminggel helyettesítettem, mindkét modell esetében kedvezőbb eredményeket hozott. Ugyanakkor a numerikus kifejezések szöveges megfelelőkkel való helyettesítése nem befolyásolta érdemben a klasszifikációs teljesítményt.

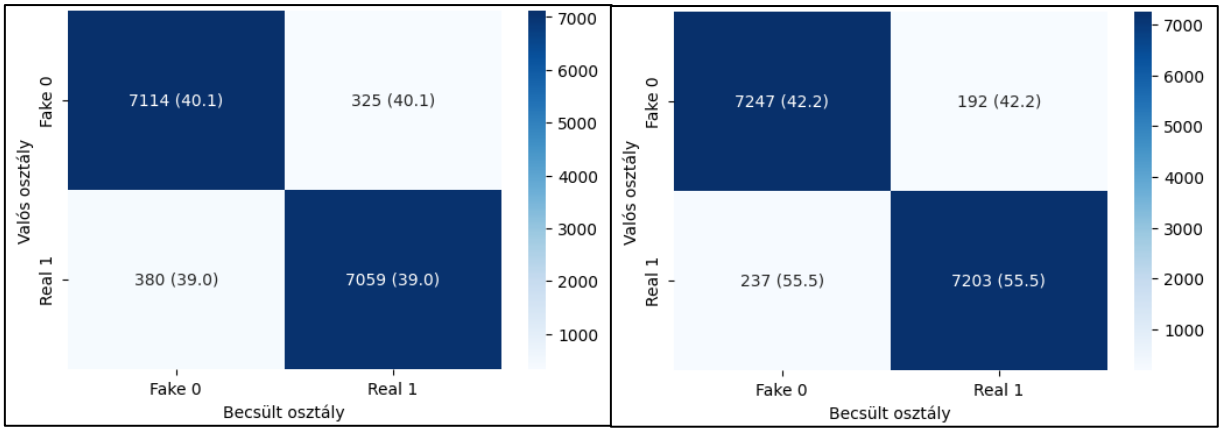
3. Táblázat: Tanítóból leválasztott teszt *adatbázis eredményei*

Előfeldolgozás	Modell	Pontosság	Precizitás	Recall	F1-score
0. Normálás *(Minden stop maradt)*	LSTM	95,99% (0,6%)	96% (0,8%)	95,96% (0,5%)	95,99% (0,6%)
	GRU	**97,11% (0,1%)**	97,4% (0,6%)	96,8% (0,8%)	**97,1% (0,2%)**
1.*Álhír stopszavak maradtak*	LSTM	95,66% (0,4%)	95,71% (0,2%)	95.59% (1%)	95.65% (0,5%)
	GRU	96.70% (0,1%)	96.66% (0,4%)	96.75% (0,1%)	96.7% (0,1%)
2.*Minden stopszó maradt cikkekben*	LSTM	96,43% (0,5%)	96,83% (0,5%)	96% (0,7%)	96,41% (0,5%)
	GRU	96,98 (0,1%)	**97,32% (0,2%)**	96,62% (0,1%)	96,97% (0,1%)
3.*Semmi stopszó nem maradt*	LSTM	95,25% (0,5%)	95,59% (0,8%)	94,88% (0,7%)	95,23% (0,5%)
	GRU	96,43% (0,3%)	95,6% (0,6%)	**97,33% (0,4%)**	96,46% (0,3%)
4.*Stemming* *(Minden stop maradt)*	LSTM	96,7% (0,2%)	96,47% (0,5%)	96,96% (0,9%)	96,71% (0,2%)
	GRU	97,08% (0,1%)	97,16% (0,2%)	97% (0,5%)	97,08% (0,2%)
5.*Szám szöveggé* *(Minden stop maradt)*	LSTM	96,41% (0,1%)	96,62% (0,1%)	96,18% (0,4%)	96,4% (0,2%)
	GRU	96,96% (0,2%)	97,1% (0,3%)	96,8% (0,4%)	96,95% (0,2%)

Az LSTM és GRU típusú rekurrens neurális hálók teljesítményének összehasonlítása során megállapítható, hogy bár mindkét modell közel azonos, átlagban 96% körüli értékeket ért el fél százalékpont vagy kisebb szórással, valamennyi kiértékelési metrikában, az alkalmazott öt előfeldolgozási módszer mindegyikében a GRU architektúra következetesen felülmúlta az LSTM teljesítményét. A **leggyengébben** szereplő modell, ami minden szempontban a legrosszabb eredmény nyújtotta, az **LSTM** azon változata, ahol a korpuszból el lett távolítva minden stopszó. Ugyanakkor fontos kiemelni, hogy ez a **„legrosszabb" eredmény is 95%** körüli

teljesítményt mutatott mind pontosságban, precizitásban, recall-ban és F1-score-ban, amely továbbra is erős modellre utal. A **legjobb** modell a **GRU** hálózat volt, amely a csak normálás alapú, 0. előfeldolgozással előkészített korpuszon 96,8%-os recall és 97,1%-os F1-score értékeket ért el minimális szórással. A **legjobb és a leggyengébb eredmény között** csupán **1,5 százalékpont körüli eltérések** mutatkozott, ami a két modell általános teljesítményében jelentéktelen különbséget jelez. Ez a kis mértékű eltérés a *10. ábrán* az ötfutásos, átlagolt és zárójelben szórást mutató konfúziós mátrix alapján is csupán korlátozottan érzékelhető.

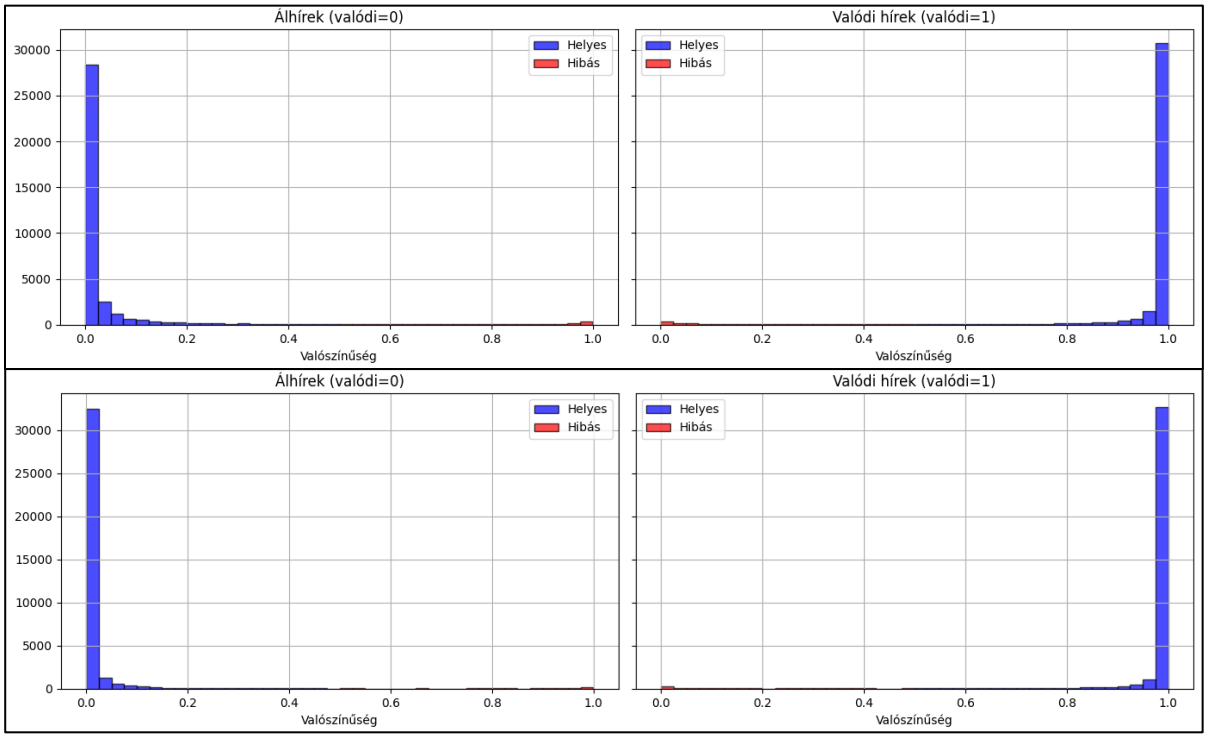
10. Ábra: *Konfúziós Mátrixok Leválasztott teszt korpusz, Legrosszabb LSTM bal és Legjobb GRU jobb oldalt*



A cikkekhez tartozó valószínűségi eloszlások, azt mutatják, a korábbi két példánál maradva, hogy a modellek a cikkek besorolást nagy magabiztossággal sorolják a megfelelő osztályba. Ezt az eloszlást jól szemlélteti a *11. ábrán* szereplő hisztogramok. A legjobb és legrosszabb eredmények közti különbség minimálisan látható csak, abban, hogy a felső ábrában jobban látható a, hogy azokat a hírek melyeket rosszul sorol be, nagy „magabiztossággal" teszi.

Összességében az eredmények azt mutatják, hogy a modellek magas pontossággal, magabiztossággal és osztályozási valószínűséggel képesek a cikkeket a két kategóriába sorolni, előfeldolgozási módszerek közti minimális különbségekkel, miközben a GRU szisztematikusan felülmúlja az LSTM teljesítményét. Mindazonáltal, fontos megjegyezni, hogy ezek az eredmények a tanítókorpuszból leválasztott 20%-os tesztkészleten alapulnak, így elsősorban azt tükrözik, mennyire hatékonyan tudtak a modellek tanulni az adott adathalmazból, az általánosíthatósági képességek a következő teszthalmazokon nyilvánul meg.

11. Ábra: *Valószínűségi eloszlás, Leválasztott teszt korpusz, legrosszabb LSTM felül, legjobb GRU alul*



Eredmények a kombinált független adathalmazon

Következőként a *4.* táblázat segítségével a külső, kombinált-független adathalmazon elért eredményeket ismertetem. A validációs tanulási és veszteség mutatók a korábbiakban bemutatott módon alakultak. Még a részletes elemzést megelőzően, a *3.* és a *4.* táblázatban szereplő teljesítménymutatókat összevetve, az adatokból jól látható, hogy az eredmények jelentős mértékben eltérnek a tanító adathalmazból leválasztott tesztkészleten tapasztalt értékektől. Ez az eltérés arra utalhat, hogy a modellek nagy mértékben támaszkodnak a tanító adathalmaz statisztikai mintázataira, és ezekhez jól adaptálódnak, azonban általánosítási képességük korlátozottabb, amikor korábban nem látott, heterogénebb vagy eltérő eloszlású adatokkal találkoznak. Ez különösen a Recall és F1-score mutatók drasztikus csökkenésében mutatkozik meg, ahol az látható, hogy míg a tanítóból leválasztott teszten ezek a mutatók 95–97% körül mozognak, addig a kombinált-független teszthalmazon több esetben akár 10–20 százalékpontos csökkenéssel, 55–75% közötti értékek figyelhetők meg.

***4. Táblázat: Kombinált független** adatbázis eredményei*

Előfeldolgozás	Modell	Pontosság	Precizitás	Recall	F1-score
0. Normálás *(Minden stop maradt)*	LSTM	77,19% (4,7)%	92,3% (2%)	59,55% (10%)	71,93% (7,4%)
	GRU	79,73% (2%)	**95,99% (1,2%)**	62,24% (3,7)	75,48% (3%)
1.*Álhír stopszavak maradtak*	LSTM	74,51% (2,5%)	93,02% (1,1%)	52,99% (5,1%)	67,41% (4%)
	GRU	75,17% (2,9%)	93,33% (1,1%)	54,16% (5,9%)	68,41% (5%)
2.*Minden stopszó maradt cikkekben*	LSTM	80,66% (6,3%)	91,06% (2%)	67,78% (12%)	77,31% (8,7%)
	GRU	**85,41% (4,3%)**	**95,58% (1,1%)**	**74,28% (9,3%)**	**83,33% (5,9%)**
3.*Semmi stopszó nem maradt*	LSTM	75,1% (1,8%)	93,26% (1,8%)	54,08% (5,8%)	68,33% (5%)
	GRU	76,25% (1,2%)	93,21% (2,2%)	56,65% (1,6%)	70,45% (1,6%)
4.*Stemming* *(Minden stop maradt)*	LSTM	78,74% (3,7%)	92,06% (2,5%)	62,87% (6,9%)	74,57% (5,2%)
	GRU	81,56% (4,3%)	95,25% (1,1%)	66,37% (8,4%)	78,02% (6,1%)
5.*Szám szöveggé* *(Minden stop maradt)*	LSTM	84,10% (4,2%)	94,1% (1,4%)	72,75% (8,7%)	81,83% (5,8%)
	GRU	85,41% (4,9%)	95,59% (1,5%)	74,22% (9,8%)	83,29% (6,5%)

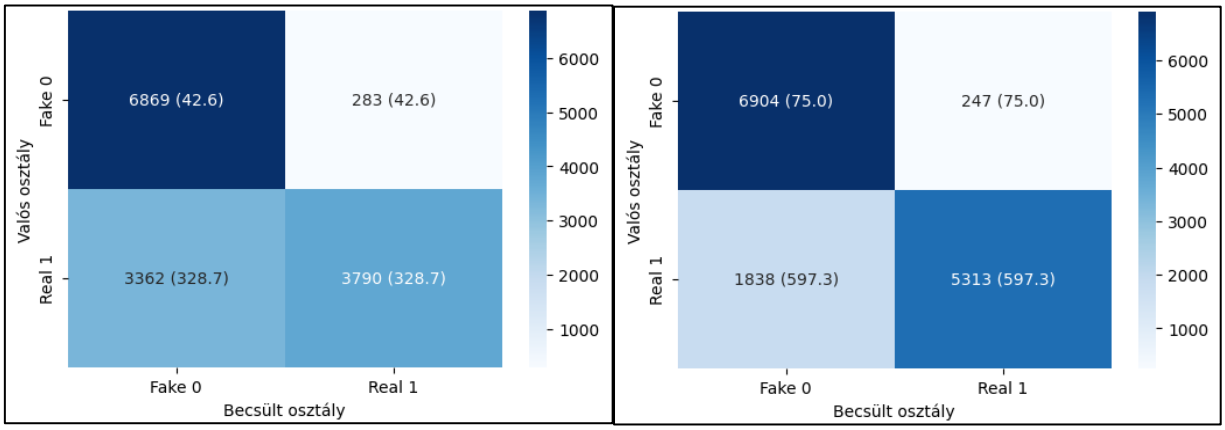
Az itt elért adatok megerősítik a korábbi eredmények alapján tett megállapításokat, miszerint a stopszavakat megtartó előfeldolgozási eljárások rendszeresen jobb teljesítményt nyújtanak, mint azok a módszerek, amelyekben a stopszavakat részben vagy teljes mértékben eltávolítottam. A precizitás minden vizsgált esetben meghaladta a 91%-ot, viszont **a recall és az F1-score** értékek sokkal pontosabb képet nyújtanak a modellekről. Ezeknél a mutatóknál jóval nagyobb szórás és jelentősebb eltérések figyelhetők meg az egyes előfeldolgozási stratégiák között. A szórásértékek különösen az utóbbi két metrikáknál emelkednek ki, ahol a legtöbb modellnél meghaladják az 5,8 százalékpontot, egyes esetekben, pedig elérik a 12 százalékpontot. Ez a viszonylag nagy variancia arra utal, hogy a modellek teljesítménye ezekben a dimenziókban érzékenyebb a teszthalmaz változatosságára és az előfeldolgozási beállításokra, tehát ezen metrikák alapján mért teljesítmény kevésbé tekinthető stabilnak. Mindazonáltal, a **legeredményesebb előfeldolgozási módszer** mind az LSTM, mind a GRU modell esetében az volt, amelyben a **stopszavakat meghagytam** a szövegkorpuszon belül, továbbá a számokat szöveges megfelelőikkel helyettesítettem. Ez az 2. és 5. előfeldolgozási stratégia GRU modell esetén 85,41%-os pontosságot, 95,59-95,58%-os precizitást, valamint 83,33-83,29%-os F1-score-t eredményezett, amelyek mind a vizsgált konfigurációk legjobb értékei közé tartoznak. Ugyanez az előfeldolgozások LSTM esetén is kiemelkedően

teljesítettek, megerősítve a két módszer általánosan pozitív hatását. Viszont, bár szinte megegyeztek az eredmények, a **legjobban teljesítő** neurális hálózati **modell a GRU volt, a 2 előfeldolgozás** során. Azért emeltem ezt ki, mint legjobb modell, mert bár átlagok az 5.mószerrel szinte megegyeznek, viszont a szórás 1 százalékponttal alacsonyabb a recall és F1-score tekintetében, így ez az eredmény **megbízhatóbb**. Kiemelendő még, hogy a 0. módszer esetében, ahol csak normalizálást végeztem, míg a korábbi teszthalmazon elért eredmények a legjobbak lettek az összehasonlítás során, ebben az esetben a stopszavakat tartalmazó módszerek közül, a leggyengébb eredményt érte el.

Mindemellett a modellek összehasonlításán az látható, hogy a GRU modellek továbbra is konzisztensen jobban teljesítenek, 2-5% százalékponttal recall és F1-score esetében, mint az LSTM-ek. A **legjobb eredményt** a korábban említett **GRU** hálózat nyújtotta az **2. lemmatizáláson** és **5. szám-szöveggé** előfeldolgozási módszer esetén, míg a **legrosszabb** ismét egy **LSTM** hálózat, **az első**, csak válogatott stopszavakat tartalmazó előfeldolgozási technika esetében.

A pontosság és precizitás értékeket, azért kerültek kiemelésre, mert mint a *12. ábrán* is látható, a kombinált-független teszthalmaz esetében, a legjobb és legrosszabb modell egyaránt a cikkek nagyrészét az álhírek közé sorolja. A többi modell esetében is hasonló tendencia figyelhető meg az álhíreket tekintve. Ez a torz osztályozás azt jelenti, hogy a modell inkább az álhír kategória irányába hajlik, ami felhúzza a precizitás értékét. A precizitás definíciója szerint ugyanis csak az számít, hogy az pozitívnak (valódi hírnek) becsült elemekből mennyi volt valóban az, de nem veszi figyelembe, hány valódi hír maradt rejtve a negatív osztályban. Ez a jelenség jól mutatja, hogy a precizitás önmagában nem alkalmas a modell általános teljesítményének teljes körű jellemzésére, különösen olyan esetekben, amikor az osztályozó hajlamos a pozitív osztály alulprediktálására. Éppen ezért a precizitás értelmezése során célszerű figyelembe venni más mutatókat is, mint a recall-t és az F1-scoret, amelyek kiegyensúlyozottabb képet adnak az osztályozási teljesítményről.

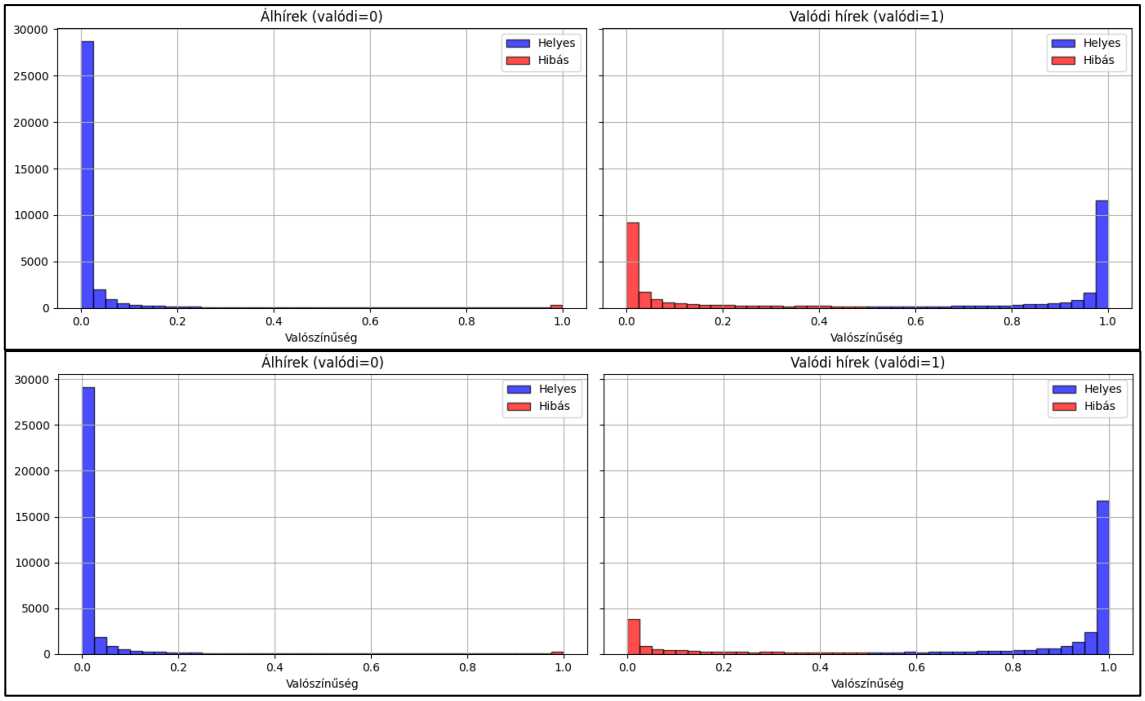
12. Ábra: *Konfúziós Mátrixok, Független korpuszon, legrosszabb LSTM bal, legjobb GRU jobb oldalt*



A predikciós valószínűségek eloszlása a *13. ábrán* ezt a jelenséget tovább erősíti. A bal oldali hisztogramokon (álhírek osztályozása LSTM és GRU esetében) látható, hogy az álhír példák többségét a modellek rendkívül magabiztosan (0–0,1 közötti valószínűséggel) osztályozzák, szinte teljes bizonyossággal. A valódi hírek esetében (jobb oldali ábrák) jól megfigyelhető egyfajta osztályozási torzítás: a helyesen osztályozott példák esetén a modell által adott valószínűségi értékek többsége közel esik az 1-hez, vagyis a modell magabiztosan és helyesen sorolja őket a valódi hírek közé. Ugyanakkor a hibásan osztályozott példák esetében markáns koncentráció figyelhető meg az alacsony, jellemzően 0,0 és 0,4 közé eső valószínűségi tartományban. Ez arra utal, hogy a modell nem csupán tévesen sorolja ezeket az eseteket az álhírek közé, hanem ezt nagy fokú bizonyossággal teszi. A fentiek alapján tehát megállapítható, hogy a modell nemcsak hibázik, hanem esetenként nagy magabiztossággal követ el klasszifikációs hibát, ami különösen aggályos lehet érzékeny alkalmazási területeken.

A korábban legjobb modell félrebesorolt cikkeinek saját megvizsgálását követően, az figyelhető meg, hogy a **hamis negatívnak**, vagyis az álhírnek ítélt igaz hírek, tematikusan többnyire olyan területeket érintettek, amelyek közéleti vagy geopolitikai feszültséget hordoznak, mint például politikai botrányok, migráció, terrorizmus, vagy nemzetközi konfliktusok. A leggyakoribb szavak között szerepelnek, ha a teljes teszthalmaz 30 leggyakoribb szavát eltávolítjuk, a következőek: *word, philippine, national, case, report, police, force, law, ukraine*. A szavak alapján látható az a tendencia, hogy a modell hajlamos volt álhírként besorolni azokat az igaz híreket, amelyek közéleti vagy geopolitikai feszültséget hordozó témákat, mint például rendvédelemet, jogi ügyeket vagy nemzetközi konfliktusokat, érintettek. Az olyan gyakori szavak, mint *law, police, national*, erre utalnak. A *philippine* szó

13. Ábra: *Valószínűségi eloszlás, Független korpuszon, legrosszabb LSTM felül, legjobb GRU alul*



megjelenése pedig arra utal, hogy a modell nehezen kezelte a Fülöp-szigeteki eseményekről szóló híreket, amelyek gyakran kapcsolódtak politikai válságokhoz vagy biztonsági intézkedésekhez.

Az igaznak sorolt álhírek, vagyis a **hamis pozitív** cikkek esetében, jellemzően olyan álhíreket tartalmaztak, amelyek formailag erősen hasonlítanak a hiteles hírekre. A korábbihoz hasonló módon végzett szógyakoriság alapjpán, itt, többek közt, a következő szavak voltak gyakoriak: *trump, clinton, rpesident, new, donald government, american, hillary, israel, obama*. A szógyakoriság alapján az látszik, hogy ezek a cikkek gyakran amerikai bel- és külpolitikai témákhoz kapcsolódtak, különösen ismert politikai szereplőkhöz, mint *trump, clinton* vagy *obama*. Ezek az álhírek gyakran tartalmazhattak hamis állításokat róluk, miközben nyelvezetük és szerkezetük, például az olyan szavak használata, mint *government, policy, american, healthcare*, erősen hasonlított a hiteles hírekre. A hivatalos hangnem és a tényszerűnek ható kulcsszavak megtéveszthették a modellt, amely így nehezen tudta elkülöníteni ezeket a jól megfogalmazott, de valótlan tartalmakat az igaz hírektől.

Mindkét félrebesorolási esethez, töltöttem fel 15-15 példát a kódok mellé[15], ahol a valószínűségek vagy 1-hez vagy 0-hoz nagyon közel vannak, vagyis ahogy a modell „nagy magabiztossággal” tévedett.

Összegezve, a kombinált független teszthalmazon végzett vizsgálatok rávilágítottak a modellek korlátozott általánosítási képességére és a tanítóhalmazból származó mintázatok iránti érzékenységére. Az eredmények megerősítették, hogy a stopszavak megtartása és a számok szöveges reprezentációja pozitív hatással volt a modellek teljesítményére, különösen a GRU hálózat esetében, ami a korábbi vizsgálatokhoz hasonlóan, minden előfeldolgozási módszerben kedvezőbb mutatókkal rendelkezett az LSTM-hez képest. Emellett kiemelendő volt a precizitás mutató stabilan magas értéke, miközben a recall és F1-score nagy varianciája a modellek instabilitására és érzékenységére utal. Problémaként merült fel az álhírek irányába történő torz osztályozás, valamint a hamis pozitív és negatív hibák tematikus sajátosságai.

2025 márciusi cikkek eredményei

Az *5. táblázatban* bemutatott eredmények alapján megállapítható, hogy a modell teljesítménye jelentősen elmarad a tanítókorpuszból leválasztott belső teszthalmazon mért, korábban kiemelkedő eredményektől. Ez az eltérés várható volt, tekintettel arra, hogy a belső teszthalmaz szerkezetileg és nyelvileg is szorosabban kapcsolódik a tanítóadatokhoz, így a modell számára könnyebben értelmezhető és feldolgozható mintázatokat tartalmaz. Azonban, érdekes módon az itt vizsgált harmadik teszthalmaz, amely manuálisan gyűjtött, 2025 márciusában megjelent cikkekből áll, meglepően hasonló teljesítményt eredményezett, mint a korábban tesztelt, Kaggle- és GitHub-forrásból származó, kombinált független adathalmaz, sőt, bizonyos metrikák, mint a recall és az F1-score esetében enyhe javulás is tapasztalható a korábbi, szintén régebbi (2016 körüli) cikkeket tartalmazó adatokhoz képest.

Ez a megfigyelés különösen figyelemre méltó abból a szempontból, hogy a tanulási fázis során a modell szinte kizárólag 2016 körüli forrásokból származó szövegeken keresztül tanulta meg a klasszifikációhoz szükséges nyelvi mintázatokat, míg a jelenlegi teszthalmaz közel egy évtizeddel későbbi aktuális szövegeket tartalmaz. Az, hogy a modell ilyen jelentős időbeli eltérés mellett is képes volt értelmezhető és bizonyos mutatókban még javuló teljesítményt

[15] https://github.com/Konye07/Konye-MscCode/tree/main/preproc_and_models/02second_method/felrebesoroltak

nyújtani, arra utal, hogy a tanult reprezentációk bizonyos fokú **időbeli általánosíthatóságot** is magukban hordoznak. Mindez azt jelzi, hogy a modell nem kizárólag a tanítóadatokhoz igazodva működik jól, hanem képes felismerni azokat az alapvető nyelvi és tartalmi mintázatokat, amelyek a médiadiskurzus változásai ellenére is fennmaradnak.

***5. Táblázat: 2025 márciusi** adatbázis eredményei*

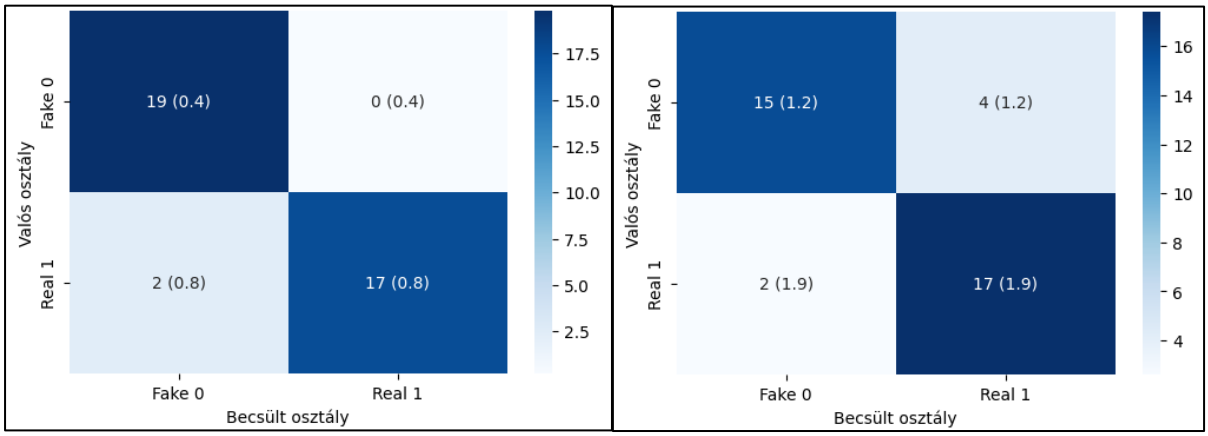
Előfeldolgozás	Modell	Pontosság	Precizitás	Recall	F1-score
0. Normálás (Minden stop maradt)	LSTM	**93,2% (1,3%)**	**99% (2,2%)**	88% (4,4%)	**93,08% (1,6%)**
	GRU	92% (2,5%)	95,72% (2,4%)	89% (4%)	92,19% (2,6%)
1.*Álhír stopszavak maradtak*	LSTM	89,5% (2%)	87,99% (5,2%)	92% (2,7%)	89,81% (1,6%)
	GRU	86% (2,8%)	81,33% (4,4%)	**94% (4,1%)**	87,07% (2,2%)
2.*Minden stopszó maradt cikkekben*	LSTM	84,5% (4,4%)	85,44% (6,3%)	84% (9,6%)	84,27% (4,9%)
	GRU	85,5% (5,4%)	82,89% (5,4%)	90% (7%)	86,3% (4,1%)
3.*Semmi stopszó nem maradt*	LSTM	88% (2,7%)	92% (6,7%)	84% (6,5%)	87,48% (2,8%)
	GRU	86% (4,8%)	85,09% (7,2%)	88% (4,4%)	86,36% (4,5%)
4.*Stemming (Minden stop maradt)*	LSTM	85,5% (2,7%)	88,4% (3,4%)	82% (7,5%)	84,85% (3,4%)
	GRU	83% (7,3%)	80,49% (5,9%)	87% (10%)	83,5% (7,5%)
5.*Szám szöveggé (Minden stop maradt)*	LSTM	88% (2,7%)	85,9% (2,9%)	91% (4,1%)	88,33% (2,7%)
	GRU	87% (2%)	86,18% (5,9%)	89% (6,5%)	87,24% (1,9%)

Az előfeldolgozási módszerek eredményeit összehasonlítva, a **legjobb eredményt a 0**., vagyis az a módszer nyújtotta, ahol a **normalizáláson kívül mást előfeldolgozás nem volt alkalmazva**. Ez a módszer rendkívül magas értékeket ért el mindkét modell esetében mind a négy kiértékelési metrikában, de a precizitás és pontosság értékei emelkedtek ki leginkább, ahol a két modell eredményeinek átlaga 93,2% és 99% lett. Ez abból az okból kifolyólag lehet, mert a tanulható GloVe alapú, 300 dimenziós szóbeágyazás képes volt kihasználni az eredeti szóalakokban rejlő morfológiai és szemantikai információkat, amelyeket a lemmatizálás vagy stemming során elveszített volna a modell. Így az eredeti szóalakok és stopszavak megtartása lehetővé tette, hogy a modell finomabb nyelvi mintázatokat is felismerjen és megtanuljon, ezzel hatékonyabbá téve az általánosítást. Ezzel szemben a **leggyengébben teljesítő előfeldolgozási eljárás a második és negyedik** módszer lett, amelyekben a szövegek lemmatizáláson, illetve stemmingelésen estek át, miközben a stopszavak is megtartásra kerültek. Ezen módszerek gyengébb teljesítménye valószínűsíthetően a korábban kifejtett

okok miatt vannak. Egyrészt, a lemmatizálás és a stemming eljárások az eredeti szóalakokat redukált formájukra egyszerűsítik, amely folyamat során számos nyelvtani, szemantikai és stilisztikai információ elvész, amik jelentős szerepet játszhatnak a GloVe alapú szóbeágyazások hatékony működésében, mivel a tanulható vektorterek az eredeti nyelvi kontextus megőrzésére és finom szemantikai különbségek reprezentálására épülnek. Másrészt, a stopszavak megtartása a redukált szóalakokkal együtt zavaró hatást is gyakorolhatott a modellre. Ennek eredményeként a tanulási folyamat során a modell kevésbé releváns vagy torzított mintázatokat sajátíthatott el, ami végső soron a kiértékelési metrikákban is alacsonyabb pontszámokat eredményezett. Az így kapott legjobb és legrosszabb modellek

A modellek esetében is megfordultak az itt kapott eredmények. A kapott eredmények alapján az LSTM modellek jobban teljesítenek szinte az összes mutatóban, mindegyik előfeldolgozási metódusban a másodiktól eltekintve. Bár nagyobbak az elért százalékok, de ezzel együtt a szórások is 1-2 százalékponttal is nagyobbak, mint a GRU-nál. Érdekes eredmény még az, hogy messze a **leggyengébb modell GRU** lett, a **4. előfeldolgozási** módszer során. Az itt szereplő, **stemming alapú GRU** és a csak normalizált esetben lévő LSTM öt lefutású átlagolt eredményeinek konfúziós mátrixai láthatóak a *14. ábrán*, mint a legjobb és legrosszabb modellek összehasonlítása.

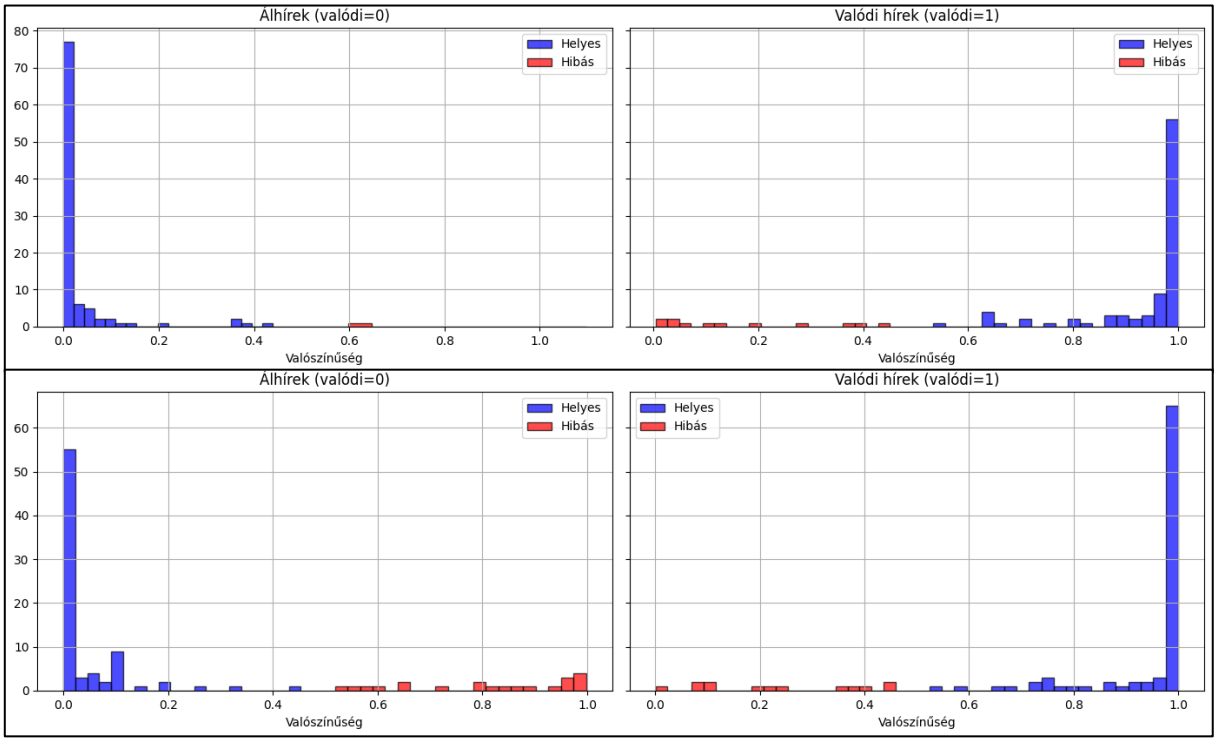
14.Ábra: *Konfúziós Mátrixok, 2025 márciusi korpuszon, legjobb LSTM bal, legrosszabb GRU jobb oldalt*



A konfúziós mátrixok alapján jól látható, hogy az LSTM modell, a korábbi független teszthalmaz eredményeihez hasonlóan, az osztályozás során az „álhír” kategória irányába torzít, vagyis a hamis negatív esetek száma magas. Ezzel szemben a GRU inkább, az igaz hírek irányába torzít.

Az ugyanezen modell eredményeinek aggregált valószínűségi eloszlásain, a *15. ábrán*, az figyelhető meg, hogy az az LSTM modell (felső sor) esetében az álhírként jelölt cikkek esetén a helyes predikciók túlnyomórészt a 0-0,1 valószínűségi tartományban koncentrálódnak, ami arra utal, hogy a modell nagy magabiztossággal képes detektálni az álhíreket. Ezzel szemben a valódi hírek esetében szélesebb eloszlást mutatnak, mivel több cikk is található a közepes (0,2–0,8) valószínűségi tartományban. Emellett, az alsó sorban szereplő GRU modell esetében szintén megfigyelhető a magabiztos, balra torzult eloszlás, viszont az álhírek klasszifikációjában is megjelennek a nagyobb bizonytalanságot mutató értékek. Mindez alapján, arra lehet következtetni, hogy az LSTM modell nemcsak pontosabb, hanem következetesebben is osztályoz a 2025 márciusi cikkek álhíreinek esetében.

15. Ábra: *Valószínűségi eloszlás, 2025 márciusi korpuszon, legjobb LSTM felül, legrosszabb GRU alul*



A modellek esetében a félrebesorolt, **hamis negatív,** hamisnak sorolt igaz cikkek közt, a korábbiakhoz hasonlóan az ötszöri futtatás miatt, mindegyik cikk ötször szerepel a *15. ábrán*, melyek közül a félresoroltak, szinte mindig ugyanazokat tartalmazták, csak más valószínűségi értékkel. Ezeknek a témái a sportot, háborút (Ukrajna, Jemen), illetve tudományt (sötét energia, űrrakéta) ölelték magukba. A **hamis pozitív**, vagyis igaznak becsült álhíreket több kisebb csoportra lehet bontani. Ilyen témák a különböző összeesküvés elméletek, amelyekbe

beletartoznak a politikai és tudományos hírek. Ezek amiatt tűnhetnek valós híreknek, mert a tartalom valós elemekkel van keverve, és a szakmai, tudományos nyelvezetet utánozzák.

Összegezve, az itt kapott eredményeket, a modellek teljesítménye meglepően stabil maradt az újabb, időben teljesen eltérő teszthalmazon, méghozzá a recall és F1-score tekintetében enyhe javulás is megfigyelhető, a kombinált független teszthalmazhoz képest, ami az alacsony mintaszámra utalhat, de a százalékos eredmények alapján a tanult nyelvi reprezentációk képesek időbeli általánosíthatóságra. Az előfeldolgozási technikák összehasonlítása ebben az esetben azt mutatta, hogy a **legjobb eredményeket** a minimális beavatkozást alkalmazó (**csak normalizáló, 0.**) módszer nyújtotta, ami a szóbeágyazás révén lehetett ilyen magas, mivel a GloVe alapú szóvektorok hatékonyabbak eredeti szóalakokkal, mivel így megőrizhetők a morfológiai és szemantikai finomságok. Ezzel szemben a lemmatizálást és stemminget alkalmazó eljárások gyengébb teljesítményt hoztak, vélhetően az információveszteség és zajos mintázatok miatt.

Modellszinten az **LSTM** architektúra konzisztensen **jobban szerepelt**, bár nagyobb szórásokkal. Érdekesség, hogy mind a legjobb, mind a legrosszabb modell LSTM volt, előbbi a nyers, utóbbi a túlzottan redukált bemeneten futott. A konfúziós mátrixok és valószínűségi eloszlások alapján az LSTM nemcsak pontosabb, hanem következetesebb is volt az álhírek felismerésében, míg a **GRU nagyobb bizonytalanságot** mutatott.

Konklúzió

A szakdolgozatom célja az volt, hogy különböző előfeldolgozási eljárások mellett kiértékeljem és összehasonlítsam az LSTM és GRU típusú rekurrens neurális hálózatok teljesítményét az álhírek klasszifikációjában. Az elemzés középpontjában az állt, hogy meghatározható-e, melyik modell teljesít jobban az angol nyelvű hírcikkeken, illetve milyen hatással vannak az előfeldolgozási lépések a klasszifikációs teljesítményre.

Az eredmények alapján **egyértelmű fölényről nem lehet beszélni** a két modell között. A tanítóból leválasztott **teszt- és a kombinált független korpuszon** is a **GRU konzisztensen jobb** teljesítményt mutatott, viszont a 2025 márciusi, időben későbbi, frissebb cikkeket tartalmazó **harmadik teszthalmazon** az **LSTM bizonyult stabilabbnak**, különösen a pontosság és F1-score szempontjából. Ez arra utalhat, hogy az LSTM bizonyos jobb időbeli általánosító-képességgel

rendelkezik, viszont mivel ez az adatbázis csak 40 cikket tartalmazott, így statisztikai szempontból ez nem tekinthető elég reprezentatívnak. A megbízhatóbb következtetésekhez szükséges lenne a 2025-ös korpusz bővítése több ezer cikkre. Ebből fakadóan, ki lehet jelenteni, hogy a GRU neurális hálózat teljesített jobban, vagyis amellett, hogy egyszerűbb és kevesebb erőforrást használ fel, a kutatásom alátámasztja azokat a korábbi szakirodalmakat, amik hasonló eredményt értek el, vagyis teljesítményileg is előnyösebb az LSTM-el szemben.

A különböző előfeldolgozási technikák összehasonlítása alapján elmondható, hogy az **összes stopszó megtartása hozta a legjobb eredményeket** az első két teszthalmazon, mindkét modell esetében. Hangsúlyos, hogy még a 0. módszer is, ahol a normalizáláson (és GloVe alkalmazásán) kívül más lépésen nem esett át a korpusz, is jobb mutatókkal rendelkezett. Ez némileg meglepő, mivel a klasszikus NLP-gyakorlat a szakirodalmakat tekintve a stopword-ök eltávolítását preferálja, mivel zajként hivatkoznak rájuk. Az eredmények alapján viszont úgy tűnik, hogy a nyelvi mintázatok megőrzése fontosabb lehet, mint azok egyszerűsítése, különösen neurális hálózatok esetén. Bár megpróbáltam „álhír stopszó" listát létrehozni a nyelvi források alapján, érdemesebb az összes szót megtartani. A „stemming" típusú előfeldolgozások nem vezettek szignifikáns javuláshoz, sőt néhol csökkentette a recall értéket, így ennek alkalmazását érdemes körültekintően mérlegelni. A számok szöveggé alakítása viszont nagyban növelte a modellek pontosságát a független-kombinált korpuszon, ami alátámasztja azt az állítást, hogy a számok megfelelő kezelése jelentősen javíthatja a modellteljesítményt a hírek tekintetében.

Továbbá fontos kiemelni, hogy a tanulási és kiértékelési folyamatot mindössze ötször futtattam le, ami bár hasznos képet adott az átlagos teljesítményről, nem szolgáltat elég megbízható szórásbecslést. A jövőbeli munkákban indokolt lenne a modelleket nagyobb számú (pl. 50–100) független futtatásnak alávetni, hogy a teljesítménymutatók statisztikailag megalapozottabb képet adjanak.

Mindemellett, a dolgozatban alkalmazott megközelítések és modellek gyakorlati alkalmazási lehetőségei is kifejezetten relevánsak. A klasszifikációs rendszer beépíthető lenne webes platformok vagy közösségimédia-felügyelő rendszerek backendjébe, ahol első szűrőként működhetne az álhírek azonosításában. Ugyanígy alkalmazható lenne újságírói munkát segítő eszközként, amely cikkek automatikus előértékelését végzi, valamint oktatási vagy kutatási célból, a tartalmak megbízhatóságának vizsgálatához. Végezetül fontosnak

tartom kiemelni, hogy bár a kutatás angol nyelvű adatbázisokon zajlott, nagy szükség lenne egy magyar nyelvű, általános híreken alapuló álhír korpusz létrehozására, valamint az arra épülő, szélesebb tematikájú álhírdetektor fejlesztésére. Bár már létezik a Magyar Tudományos Akadémia által fejlesztett HuBERT-alapú álhírdetektor[16], annak fókusza elsősorban az egészségügyi témájú álhírekre korlátozódik. A hazai médiatér sokszínűsége és a dezinformáció tematikai változatossága miatt azonban indokolt lenne egy olyan rendszer kialakítása, amely képes a közéleti, geopolitikai vagy más társadalmi témákban terjedő álhírek felismerésére is. Egy ilyen eszköz nemcsak technikai szempontból lenne jelentős, hanem érdemben hozzájárulhatna a nyilvános diskurzus tisztaságához és a dezinformáció elleni hazai fellépéshez.

[16] https://www.alhirdetektor.hu/

Irodalomjegyzék

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation* (OSDI 2016) (pp. 265–283). USENIX Association.

Abualigah, L., Al-Ajlouni, Y. Y., Daoud, M. S., Altalhi, M., & Migdady, H. (2024). Fake news detection using recurrent neural network based on bidirectional LSTM and GloVe. *Social Network Analysis and Mining*, 14(1), 40.

Airlangga, G. (2024). Advancing fake news detection: A comparative study of RNN, LSTM, and bidirectional LSTM architectures. *Jurnal Teknik Informatika CIT Medicom, 16*(1), 13–23.

Al Sharou, K., Li, Z., & Specia, L. (2021, szeptember). Towards a better understanding of noise in natural language processing. *In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)* (pp. 53–62).

Alnabhan, M. Q., & Branco, P. (2024). Fake news detection using deep learning: A systematic literature review. *IEEE Access, 12*, 114435–114459. https://doi.org/10.1109/ACCESS.2024.3435497

Anjana, S., Saruladha, K., & Sathyabama, K. (2019, március). Bidirectional and stacked LSTM for sleep disorders prediction. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 912–916). IEEE.

Arora, M., & Kansal, V. (2019). Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis. *Social Network Analysis and Mining, 9*(1), Article 12.

Bajaj, S. (2017). *The pope has a new baby! Fake news detection using deep learning* [Kézirat]. Stanford University, CS 224N.

Bender, E. M. (2019). The #BenderRule: On naming the languages we study and why it matters. *The Gradient.* Elérhető: https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/ (Letöltve: 2024.03.12.)

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python.* O'Reilly Media.

Camelia, T. S., Fahim, F. R., & Anwar, M. M. (2024). A regularized LSTM method for detecting fake news articles. *arXiv preprint*, arXiv:2411.10713. https://arxiv.org/abs/2411.10713 .

Çetiner, H. (2024). Fake news detection and classification with recurrent neural network-based deep learning approaches. *Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 7*(3), 973–993.

Chai, C. P. (2023). Comparison of text preprocessing methods. *Natural Language Engineering*, 29(3), 509–553. https://doi.org/10.1017/S1351324922000213 .

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint*, arXiv:1406.1078. https://arxiv.org/abs/1406.1078.

Chomsky, N. (2002). *Syntactic structures* (2. kiad.). Mouton de Gruyter.

Das, A., Das, A., Datta, A., Si, S., & Barman, S. (2020, július). Deep approaches on malicious URL classification. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1–6). IEEE.

de Oliveira, N. R., Pisa, P. S., Lopez, M. A., de Medeiros, D. S. V., & Mattos, D. M. F. (2021). Identifying fake news on social networks based on natural language processing: Trends and challenges. *Information, 12*(1), 38. https://doi.org/10.3390/info12010038.

Deepak, S., & Chitturi, B. (2020). Deep neural approach to fake-news identification. *Procedia Computer Science*, 167, 2236–2243.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Dogra, V., Verma, S., Kavita, Chatterjee, P., Shafi, J., Choi, J., & Ijaz, M. F. (2022). A complete process of text classification system using state-of-the-art NLP models. *Computational Intelligence and Neuroscience, 2022*, 1883698. https://doi.org/10.1155/2022/1883698.

Du, S., Lee, J., Li, H., Wang, L., & Zhai, X. (2019, május). Gradient descent finds global minima of deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)* (pp. 1675–1685). PMLR.

Dwarampudi, M., & Reddy, N. V. (2019). Effects of padding on LSTMs and CNNs. *arXiv preprint*, arXiv:1903.07288. https://doi.org/10.48550/arXiv.1903.07288.

Eisenstein, J. (2018). *Natural language processing.* MIT Press.

Elov, B. B., Khamroeva, S. M., & Xusainova, Z. Y. (2023). The pipeline processing of NLP. *E3S Web of Conferences, 413*, 03011.

Firoozi, T., Bulut, O., Epp, C. D., Naeimabadi, A., & Barbosa, D. (2022). The effect of fine-tuned word embedding techniques on the accuracy of automated essay scoring systems using neural networks. *Journal of Applied Testing Technology*, 21, 21–29.

Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16), E3635–E3644.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation, 12*(10), 2451–2471.

Goldberg, Y. (2017). *Neural network methods in natural language processing.* Morgan & Claypool Publishers.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Haviana, S. F. C., Mulyono, S., & Badie'Ah. (2023, szeptember). The effects of stopwords, stemming, and lemmatization on pre-trained language models for text classification: A technical study. In *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)* (pp. 521–527). IEEE. https://doi.org/10.1109/EECSI59885.2023.10295797

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Horne, B., & Adali, S. (2017, május). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 11, No. 1, pp. 759–766). https://doi.org/10.1609/icwsm.v11i1.14976.

Hossain, M. D., Inoue, H., Ochiai, H., Fall, D., & Kadobayashi, Y. (2020, július). Long short-term memory-based intrusion detection system for in-vehicle controller area network bus. *In 2020 IEEE 44th Annual Computers, Software, and Applications Conference* (COMPSAC) (pp. 10–17). IEEE.

Hu, H., Liao, M., Zhang, C., & Jing, Y. (2020, június). Text classification based recurrent neural network. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)* (pp. 652–655). IEEE. https://doi.org/10.1109/ITOEC49072.2020.9141747.

Jose, X., Kumar, S. M., & Chandran, P. (2021, október). Characterization, classification and detection of fake news in online social media networks. In *2021 IEEE Mysore Sub Section International Conference (MysuruCon)* (pp. 759–765). IEEE.

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015, június). An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (pp. 2342–2350). PMLR.

Jurafsky, D., & Martin, J. H. (2025). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models* (3. kiad.). Online kézirat. Elérhető: https://web.stanford.edu/~jurafsky/slp3 (Letöltve: 2025. január 10.)

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980. https://arxiv.org/abs/1412.6980.

Kishwar, A., & Zafar, A. (2021, szeptember). Predicting fake news using GloVe and BERT embeddings. In *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)* (pp. 1–6). IEEE.

Lewkowycz, A., & Gur-Ari, G. (2020). On the training dynamics of deep networks with L2 regularization. In *Advances in Neural Information Processing Systems, 33*, 4790–4799.

Liu, Z., Li, X., Kang, B., & Darrell, T. (2019). Regularization matters in policy optimization. *arXiv preprint*, arXiv:1910.09191. https://arxiv.org/abs/1910.09191.

Lopez-del Rio, A., Martin, M., Perera-Lluna, A., & Saidi, R. (2020). Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction. *Scientific Reports, 10*, 14634. https://doi.org/10.1038/s41598-020-71450-8.

Malhotra, R., & Mahur, A. (2022, január). COVID-19 fake news detection system. In *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 428–433). IEEE.

Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing.* MIT Press.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics, 19*(2), 313–330.

Mateus, B. C., Mendes, M., Farinha, J. T., Assis, R., & Cardoso, A. M. (2021). Comparing LSTM and GRU models to predict the condition of a pulp paper press. *Energies, 14*(21), Article 6958. https://doi.org/10.3390/en14216958

Michel, P., & Neubig, G. (2018). MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 543–553). Association for Computational Linguistics.

Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15(9), 517.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint*, arXiv:1301.3781. https://arxiv.org/abs/1301.3781 .

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in pre-training distributed word representations. *arXiv preprint*, arXiv:1712.09405. https://arxiv.org/abs/1712.09405 .

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* (Vol. 26).

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM, 38*(11), 39–41. https://doi.org/10.1145/219717.219748.

Montani, I., & Honnibal, M. (2018). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks, and incremental parsing. *Explosion AI*. Elérhető: https://spacy.io (Letöltve: 2024. március 12.).

Mridha, M. F., Keya, A. J., Hamid, M. A., Monowar, M. M., & Rahman, M. S. (2021). A comprehensive review on fake news detection with deep learning. *IEEE Access, 9*, 156151–156170. https://doi.org/10.1109/ACCESS.2021.3129803.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Noguer i Alonso, M. (2024, október 27). The mathematics of recurrent neural networks. *SSRN*. https://doi.org/10.2139/ssrn.5001243.

Nosouhian, S., Nosouhian, F., & Khoshouei, A. K. (2021). A review of recurrent neural network architecture for sequence learning: Comparison between LSTM and GRU [Preprint]. *Preprints*. https://doi.org/10.20944/preprints202107.0252.v1.

Orebi, S. M., & Naser, A. M. (2025). Opinion mining in text short by using word embedding and deep learning. *Journal of Applied Data Sciences, 6*(1), 526–636.

Oshikawa, R., Qian, J., & Wang, W. Y. (2018). A survey on natural language processing for fake news detection. *arXiv preprint*, arXiv:1811.00770. https://doi.org/10.48550/arXiv.1811.00770.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *In Proceedings of the 30th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research*, 28(3), 1310–1318. https://proceedings.mlr.press/v28/pascanu13.html .

Patwardhan, N., Marrone, S., & Sansone, C. (2023). Transformers in the real world: A survey on NLP applications. *Information, 14*(4), 242.

Pennington, J., Socher, R., & Manning, C. D. (2014, október). GloVe: Global vectors for word representation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543).

Petridis, C. (2024). Text classification: Neural networks vs machine learning models vs pre-trained models. *arXiv preprint*, arXiv:2412.21022. https://doi.org/10.48550/arXiv.2412.21022.

Pimpalkar, A., Singh, M., Sheikh, S., Gedam, K., & Khadgi, A. (2021). Fake news classification using bi-directional LSTM-recurrent neural network. *Journal of Huazhong University of Science and Technology, 50*(6), 1–9. https://doi.org/10.1007/s11516-021-0000-0.

Polat, S. B., & Cankurt, S. (2023, június). Fake news classification using BLSTM with GloVe embedding. In *2023 17th International Conference on Electronics Computer and Computation (ICEC CO)* (pp. 1–5). IEEE.

Poluru, E., & Syed, H. (2023). A hybrid deep learning GRU-based approach for text classification using word embedding. *EAI Endorsed Transactions on Internet of Things, 10*(1), https://doi.org/10.4108/eetiot.4590 .

Prasad, O. J., Nandi, S., Dogra, V., & Diwakar, D. S. (2023, július). A systematic review of NLP methods for sentiment classification of online news articles. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1–9). IEEE.

Rachmawati, O. C. R., & Darmawan, Z. M. E. (2024). The comparison of deep learning models for Indonesian political hoax news detection. *CommIT (Communication and Information Technology) Journal*, 18(2), 123–135.

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017, szeptember). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 2931–2937). Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1317.

Sahala, A., & Lindén, K. (2023, szeptember). A neural pipeline for POS-tagging and lemmatizing cuneiform languages. In *Proceedings of the Ancient Language Processing Workshop* (pp. 203–212). Association for Computational Linguistics.

Saleh, H., Alharbi, A., & Alsamhi, S. H. (2021). OPCNN-FAKE: Optimized convolutional neural network for fake news detection. *IEEE Access, 9*, 129471–129489. https://doi.org/10.1109/ACCESS.2021.3112806 .

Shah, A., Ayushi, S., Akshata, G., & Satish, P. (2022). A survey on text preprocessing, data vectorization and recent NLP models. *International Journal of Emerging Technologies and Innovative Research, 9*(7), f176–f183. https://www.jetir.org/papers/JETIR2207524.pdf.

Sharma, D. K., Hota, H. S., & Rababaah, A. R. (Szerk.). (2024). Machine learning for real world applications*. Springer Nature Singapore*. https://doi.org/10.1007/978-981-97-1900-6.

Shrivastava, P., & Sharma, D. K. (2021, október). Fake content identification using pre-trained GloVe-embedding. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 1–6). IEEE.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, és Ruslan Salakhutdinov. „Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” *Journal of Machine Learning Research* 15, 56. szám (2014): 1929-1958.

Straka, M., Straková, J., & Hajič, J. (2019). Czech text processing with contextual embeddings: POS tagging, lemmatization, parsing and NER. In *Text, Speech, and Dialogue: 22nd International Conference, TSD 2019, Ljubljana, Slovenia, September 11–13, 2019, Proceedings* (Vol. 22, pp. 137–150). Springer International Publishing.

Su, Q., Wan, M., Liu, X., & Huang, C. R. (2020). Motivations, methods and metrics of misinformation detection: An NLP perspective. *Natural Language Processing Research, 1*(1–2), 1–13. https://doi.org/10.2991/nlpr.d.200522.001.

Tahat, D., Alfaisal, R., Tahat, K., & Salloum, S. (2024, december). Navigating the newscape: Long short-term memory models for misinformation detection. In *2024 11th International Conference on Software Defined Systems (SDS)* (pp. 112–114). IEEE.

Thawani, A., Pujara, J., Ilievski, F., & Szekely, P. (2021). Representing numbers in NLP: A survey and a vision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 644–656). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.53.

Toor, M. S., Shahbaz, H., Yasin, M., Ali, A., Fitriyani, N. L., Kim, C., & Syafrudin, M. (2025). An optimized weighted-voting-based ensemble learning approach for fake news classification. *Mathematics, 13*(3), Article 449. https://doi.org/10.3390/math13030449

Traylor, T., Straub, J., Chaudhary, S., & Snell, N. (2019, január). Classifying fake news articles using natural language processing to identify in-article attribution as a supervised learning estimator. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)* (pp. 445–449). IEEE. https://doi.org/10.1109/ICSC.2019.00086.

Turian, J., Ratinov, L., & Bengio, Y. (2010, július). Word representations: A simple and general method for semi-supervised learning. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 384–394).

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, s2-42*(1), 230–265.

Wang, W. Y. (2017). "Liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint*, arXiv:1705.00648. https://arxiv.org/abs/1705.00648 .

Webster, J. J., & Kit, C. (1992). Tokenization as the initial phase in NLP. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992)* (Vol. 4).

Werbos, P. J. (1990, október). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10), 1550–1560. https://doi.org/10.1109/5.58337 .

Yadav, J., Kumar, D., & Chauhan, D. (2020, július). Cyberbullying detection using pre-trained BERT model. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 1096–1100). IEEE. https://doi.org/10.1109/ICESC48915.2020.9155700.

Yolchuyeva, S., Németh, G., & Gyires-Tóth, B. (2018). Text normalization with convolutional neural networks. *International Journal of Speech Technology, 21*, 589–600.

Zamir, M. T., Ullah, F., Tariq, R., Bangyal, W. H., Arif, M., & Gelbukh, A. (2024). Machine and deep learning algorithms for sentiment analysis during COVID-19: A vision to create a fake news resistant society. *PLOS ONE,* 19(12), e0315407.

Zhou, B., Ning, Q., Khashabi, D., & Roth, D. (2020). Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7579–7589). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.678.