https://doi.org/10.36007/4782.2023.175

PLÁGIUM ELLENŐRZŐ PROGRAM AZ AKADÉMIAI ÉRTÉKELÉS INTEGRITÁSÁNAK BIZTOSÍTÁSÁRA

Dávid PAKSI[1] - Márk CSÓKA[2]

Abstract
Evaluation and assessment are unavoidable parts of education of our time. With the continuous spread of IT, a significant amount of homework and assignments became digital. In one hand this shift towards digitalization brought the teachers tools to improve the assignments' lifecycle (creation, publication, collection). On the other hand, the same technology can be comfortably used for actions not tolerated by education and academy. Besides assessment it is the teachers' role to ensure the integrity of the educational system by confirming the authenticity of the work. The tools offered by the Internet can be used fairly to create high quality submissions, however the same devices create equal temptation for students to finish the task in a more efficient, less decent way. There are already plenty of plagiarism checking tools available for use, however in most cases they lack essential features, which makes their usage limited to specific cases. This paper deals with the challenges of plagiarism checking of the most used file types of office environment in a higher education setting.

Keywords: *plagiarism, MS Office, office tools, evaluation, file comparison*

Bevezetés

A tanulók tanulmányi előmenetelének megfigyelése és felmérése kiemelkedően fontos szerepet játszik az oktatás folyamatában. A teljesség igénye nélkül az említett csoportba tartozik a tanulók fejlődésének nyomon követese, visszajelzés biztosítása a hallgató felé az elért eredményekről, tanítási folyamat felügyelete, valamint a megfelelő oktatási módszerek megválogatása. (1) Ahhoz, hogy ezen célok megvalósulhassanak, a mérésnek a lehető legpontosabbnak kell lennie. Napjainkban a digitális megoldások már szép számmal jelen vannak az oktatási intézményekben és egyaránt támogatják a tanítási, valamint tanulási folyamatot is. (2) Számos, hagyományosan papír alapú folyamat helyeződött át idővel a digitális térbe, mint például a jelenlét nyilvántartása, tananyag megosztása, értékelés és a cikk tárgyát képező beadandók menedzselése is. (3) A digitális eszközök kétségkívül hasznosak, azonban fontos figyelembe venni, hogy ezek az eszközök sajnos a tanulás megkerülésére is éppúgy alkalmasak, mint a hagyományos oktatásban is létező „rövidítések". Korunk sajátosságai miatt ezek a mérések számos esetben erősen torzulhatnak. Ide sorolnánk a jelenleg nagy port kavaró és publikusan hozzáférhető generatív mesterséges intelligenciák (pl.: ChatGPT, Google Bard, Bing AI) által készített szövegeket. (4) (5) A másik jelentős probléma főleg beadandó jellegű feladatokat érinti, mégpedig azon diákok/hallgatók személyében, akik egymás munkáját adják le kisebb-nagyobb módosításokkal. Ez a probléma kiküszöbölhető azzal, hogy a tanár minden diák számára egyedi feladatot készít elő, azonban könnyű belátni,

[1] Mgr. Paksi Dávid, Selye János Egyetem, Informatikai Tanszék, paksid@ujs.sk
[2] PaedDr. Csóka Márk, Selye János Egyetem, Informatikai Tanszék, csokam@ujs.sk

hogy ez hatalmas többletmunkát jelenthet részéről. Más esetekben pedig a feladat jellegéből adódóan nem kivitelezhető.

Plágium ellenőrzése

A plágium meghatározása „más személy szellemi alkotásának eltulajdonítása, az általa megalkotott műnek v. részeinek felhasználása az eredetire v. a szerzőre való hivatkozás nélkül; szellemi lopás, tolvajlás". (6) Az internetnek hála a megoldás néhány kattintásnyira lehet bizonyos esetekben. Azonban a tanulási folyamat sikertelennek tekinthető, ha az értékelt személy ebben a folyamatban nem vett részt, vagy annak elkerülését tűzte ki célul.

Létező megoldások

Plágiumellenőrzésre léteznek fizetős és ingyenes, telepíthető és online megoldások, ezek általában 2 fő csoportba kategorizálhatók. Az első típus képviselői gyakorlatilag szövegösszehasonlítást végeznek el a funkcionalitástól függően 2, vagy több kijelölt fájlon, kimenetük az egyes fájlok közti egyezés mértéke, esetenként a konkrét egyezések felsorolása, vagy kiemelése. A hátránya az ide tartozó szoftvereknek, hogy csak a rendelkezésre álló fájlokat hasonlítja össze (más forrásokkal, mint például online tartalmak, nem keres egyezést). (7) (8) (9) Két fájl összehasonlítására a legtöbb szövegszerkesztő képes, de ez egy csoportnyi beadandó esetén kevés.

A második csoportba azok az eszközök tartoznak, amik a vizsgált szöveget online tartalmakkal, esetleg saját, erre a feladatra elkészített adatbázisokkal hasonlítják össze. Programozás területén az egyik legismertebb ilyen eszköz a Stanford által fejlesztett és karbantartott MOSS (*Measure of Software Similarity*), ami funkcióit tekintve rendkívül felkészített, azonban nem tökéletes. (10)

A korábban említett generatív mesterséges intelligenciák aktuális népszerűsége megköveteli egy harmadik csoport bevezetését, ami azokat az eszközöket foglalja magába, amik az adott szöveg szerzőjét (személy, vagy mesterséges intelligencia) igyekeznek behatárolni. (11)

Specifikációk

Az ilyen esetek kiszűrése egy rendkívül kényes és figyelmet igénylő folyamat. A létező megoldások felmérése alapján és a témakör érzékenységéből kifolyólag az általunk készített plágium ellenőrző programmal szemben elvárásokat határoztunk meg a tervezési fázisban, amelyek a következők:

- Alkalmas nagy mennyiségű dokumentummal (fájllal) való munkára és azok összehasonlítására,
- irodai szoftvercsomag alapvető fájltípusait képes kezelni (szövegszerkesztő, táblázatkezelő, prezentációkészítő),
- automatizálható és bővíthető,
- ítélet helyett indikátorokat keres,
- több szempontot vizsgál,
- nem autonóm és oktatói felülbírálást igényel,
- viszonylag rugalmasan használható.

Esetünkben, a tesztelés során ez a mennyiség egy évfolyamnyi hallgató munkáját jelenti. Későbbiekben azonban képesnek kell lennie az aktuális beadandók a korábbi évekkel való összehasonlítására. Legyen H az összes dokumentum hasonlóságát tartalmazó mátrix

$$H = \begin{bmatrix} h_{1,1} & \cdots & h_{1,n} \\ \vdots & \ddots & \vdots \\ h_{n,1} & \cdots & h_{n,n} \end{bmatrix},$$

ahol, $h_{i,j}$ $(1 \leq i, j \leq n)$ az i-edik és a j-edik dokumentumok hasonlóságát jelöli. A főátlón lévő elemek vizsgálata nem szükséges, mivel minden dokumentum önmagával azonos. Továbbá a főátló alatti értékek a főátló felettieknek is megfeleltethetőek, így csak az egyik vizsgálata indokolt. Az ellenőrzéshez szükséges összehasonlítások száma így megadható az alábbi képlettel:

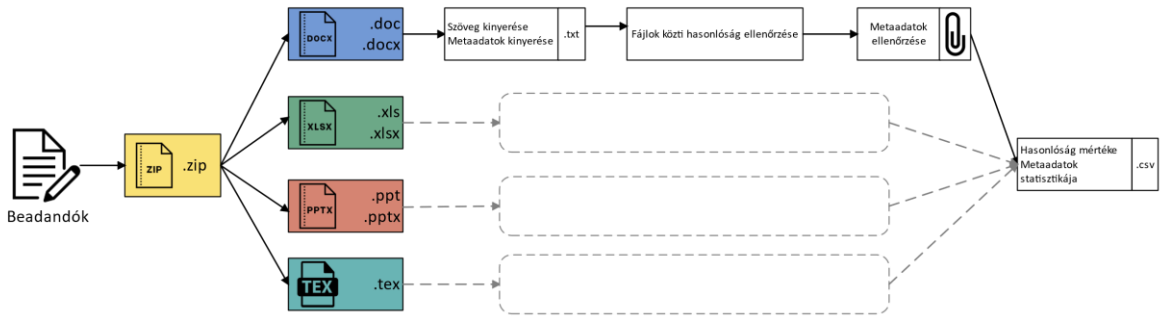
$$d_n = \frac{(n^2-n)}{2},$$

ahol n a vizsgált dokumentumok száma, d_n pedig az összehasonlítások száma n dokumentum esetén. (12)

További cél, hogy a program kimenete ne egy könnyen kezelhető adattípusban legyen, aminek a megnyitásához nincs szükség újabb alkalmazás telepítéséhez, így esett a választás a .csv-re (*Comma Separated Values*), ami egy népszerű és általános fájlkiterjesztés nagy mennyiségű szöveges adat tárolására.

Adatok kinyerése

A dokumentumok beszedését Moodle felületen végeztük, így biztosítva az adatok rendszerezett beérkezését, ami esetünkben mappákba rendezett fájlokat jelentett, ahol a mappák neve a hallgatók nevét örökli. Az állományokból python szkript segítségével kinyertük az ellenőrzésre szánt szöveges részeket, majd ezeket egy-egy *.txt* állományba mentettünk el a későbbi munkálatokhoz. Erre a lépésre azért volt szükség, mivel így több algoritmussal is lehet a jövőben összehasonlítást végezni. Ezt munkát az általunk írt *prep()* függvénnyel végeztük el, amelyben a fájlműveletekhez a *glob* modult (13), a Microsoft Office dokumentumokkal való munkához pedig a python-pptx (.ppt, .pptx állományokhoz) (14), a python-docx (.doc, .docx állományokhoz) (15) illetve a openpyxl (.xls .xlsx állományokhoz) (16) könyvtárakat használtuk. A .tex állományok kezelése a .txt állományok kezelésével megegyező volt, így nem igényelt külső könyvtárat. Fontos megemlíteni, hogy kizárólag a dokumentumok szöveges részeit nyertük ki, amelybe nem tartoznak bele például a prezentációkban szereplő diagrammok feliratai, valamint az excel cellák esetén az értékek mögött lévő képletek. Az 1. Ábra tömören összefoglalja a használt folyamatot, amely fájlkiterjesztéstől függetlenül azonos, csak más Python könyvtár szükséges hozzá. Esetünkben adott beadandóhoz konkrét fájltípus is tartozott követelményként, de vegyes fájltípusok kezelése is tervezett bővítés.



1. Ábra – Összehasonlítási folyamat beolvasástól a kiértékelésig

Hasonlóság mérése

Napjainkban digitális környezetben plagizálni sokkal kisebb erőfeszítést igényel és kézenfekvőbb megoldásnak tűnhet a diákok számára, mint megdolgozni az eredményekért. Jó hír azonban, hogy az integritás megőrzésének érdekében az szóban forgó digitális eszközöket hasonló sikerességgel lehet ellenőrzésre is használni.

Az első megközelítésünk a dokumentumok közötti hasonlóság vizsgálatára az volt, hogy összehasonlítottuk a dokumentumokból kinyerhető metaadatokat. A vizsgálat során olyan adatokat sikerült kinyerni, mint az utolsó módosítás, létrehozás dátuma, szerkesztéssel töltött idő, dokumentum szerzője, valamint a dokumentumot utoljára szerkesztő személy neve. Az így kinyert adatok összegzése hasznos információval tud szolgálni, azonban nem döntésértékű.

A következő megközelítés szerint a dokumentumok hasonlóságát mérhetjük dokumentumtávolsággal, amelyben a szavakat vektorokként kezeljük, és két adott vektor közötti szögként számítjuk ki. (17) A dokumentumvektorok a szavak előfordulási gyakoriságát mutatják egy adott dokumentumban. Jelölje A és B a két dokumentum szavait. Esetünkben az összehasonlítást a koszinuszhasonlóság matematikai és statisztikai módszerével végeztük, ami fontos eszköz számos másik területen is, mint például a gépi tanulásnál. (12) (18) A képlet felírható az alábbi alakban:

$$\cos\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

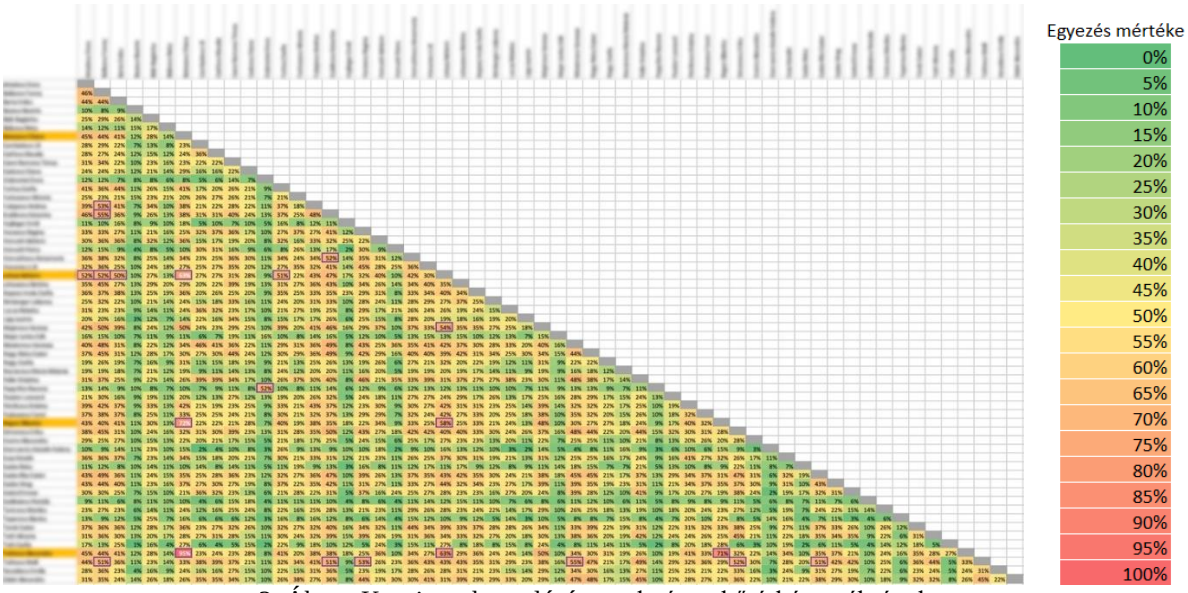
Minél közelebb van az eredmény 90°-hoz, annál hasonlóbbak a szövegek, míg 0°-hoz közelebbi eredmény azt jelenti, hogy eltérnek. A koszinuszhasonlóság eredménye radiánban van kifejezve. Legyen h az egyes fájlok közötti hasonlóság mértéke százalékban meghatározva, ekkor a hozzá tartozó képlet az alábbi

$$\mathrm{h} = \frac{2 - \theta \cdot \pi}{2}$$

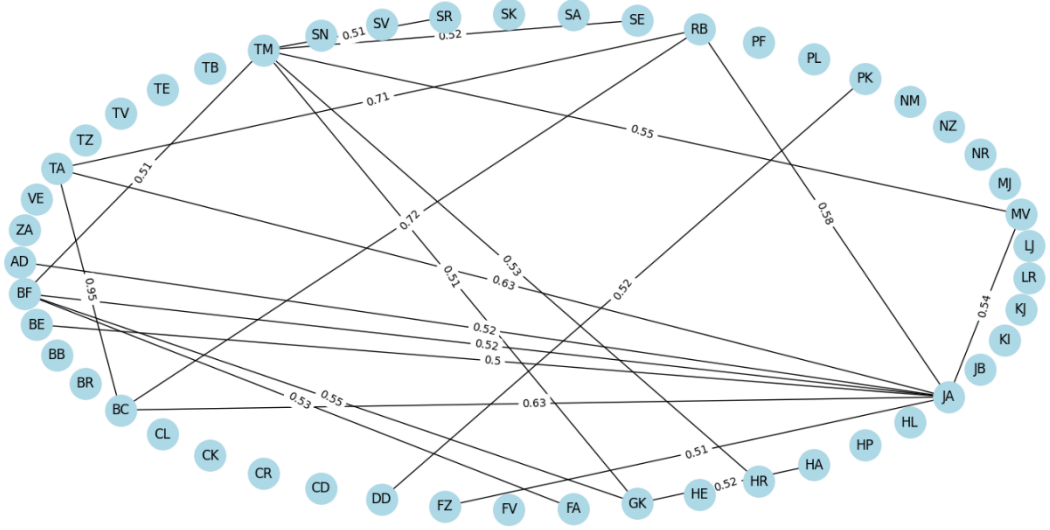
Az így kapott érték sokkal könnyebben értelmezhető, ugyanis 0% jelöli azt, hogy a két dokumentum teljesen különböző, míg 100% a teljes azonosságot.

Eredmények kiértékelése

A kapott eredményeket egy táblázatba rendeztettük (Lásd „1. Ábra”), ahol az első sorban, illetve oszlopban a hallgatók nevei kapnak helyet, többi cellában a hasonlóság százalékos meghatározása található. Mivel adott dokumentum önmagához mért hasonlósága 100%, ezért hőtérkép alapú ábrázolás esetén az átló alatti (vagy fölötti) értékek tartalmazzák az összes eredményt. Másik lehetséges vizualizáció egy kapcsolati gráf lehet.



2. Ábra - Koszinuszhasonlóság eredménye hőtérképen ábrázolva



3. Ábra -Koszinuszhasonlóság eredménye 0,5 küszöbérték esetén (50%) kapcsolati gráfon ábrázolva

Megvitatás

Hasonlóság vizsgálatára jelen esetben koszinuszhasonlóságot mérő algoritmust implementáltunk, ami az elvárásoknak megfelelt. A plágiumellenőrzésben 58 hallgató munkája került elemzésre. Amennyiben a vizsgálatot emberi erővel szerettük volna elvégezni, akkor 1635 db összehasonlítást kellett volna elvégezünk. A kiértékelés során kisebb hasonlósági értékeket is ellenőriztünk, de esetünkben a 0,5 küszöbérték (50%-nál nagyobb egyezést mutató munkák) vizsgálata hozott eredményt. A küszöbértéke helyes meghatározása nagyban függ a beadandó terjedelmétől és témájától. Esetünkben 19 emberi erőforrást igénylő összehasonlítást jelentett. A munkák áttekintése után 7 hallgatónál egyértelműen a plágium ténye került megállapításra. Ebből látszik, hogy az ellenőrzés hozott fals pozitív eredményt is, ezáltal igazolva az emberi felülbírálás szükségességét. A kapott eredményeket kielégítőnek találjuk, azonban ennek ellenére látunk lehetőséget más, hasonlóság mérésére fejlesztett algoritmusok

tesztelésében és implementálásában is. A forráskód elérhető az alábbi linken (19).
A metaadatok várakozásaink ellenére csalódást okoztak, mivel sok (reprodukálható és nem reprodukálható) esetben ezek a begyűjtés során elvesznek, módosulnak. Mindenképp hasznos funkciónak tartjuk ezek vizsgálatát, azonban eredményük jelenleg formában nem lehet döntésértékű.
A fejlesztéssel és teszteléssel töltött idő tanulságosnak bizonyult, mivel megerősítette azt az elképzelést, hogy szükség van saját, testreszabott alkalmazások fejlesztésére. A plágium és csalás kiszűrése rendkívüli kihívásokat rejtő és állandó feladat, mivel a tanulási-tanítási folyamat sikeréhez is hozzájárul.

Irodalomjegyzék

[1] *Validation of a Statewide Teacher Evaluation System: Relationship Between Scores From Evaluation and Student Academic Progress.* Xu, Xianxuan, Grant, Leslie és Ward, Thomas J. 4, hely nélk. : NASSP Bulletin, 2016., 100. kötet. 10.1177/0192636516683247.
[2] *Implementation of Digital Education Tools in the Pedagogical Community.* Kornienko, Dmitriy V és Mishina, Svetlana V. 13, hely nélk. : Journal of Higher Education Theory and Practice, 2023., 23. kötet. 10.33423/jhetp.v23i13.6370.
[3] *Digital tools in education.* Daniel, Dancsa, és mtsai. hely nélk. : International Journal of Advanced Natural Sciences and Engineering Researches, 2023., 7. kötet. 10.59287/ijanser.717.
[4] *Generative AI in education: To embrace it or not?* Samuel, Okaiyeto és Hong-Wei, Xiao. hely nélk. : International Journal of Agricultural and Biological Engineering, 2023., 16. kötet. 10.25165/j.ijabe.20231603.8486.
[5] *Generative AI and education ecologies.* Kathryn, Coleman. hely nélk. : Pacific Journal of Technology Enhanced Learning, 2023., 5. kötet. 10.24135/pjtel.v5i1.175.
[6] Arcanum. [Online] 2023. 09 01. https://www.arcanum.com/hu/online-kiadvanyok/Lexikonok-a-magyar-nyelv-ertelmezo-szotara-1BE8B/p-44572/plagium-45DB1/.
[7] GoTranscript. [Online] 2023. 09 01. https://gotranscript.com/text-compare.
[8] Quillbot. [Online] 2023. 09 01. https://quillbot.com.
[9] Text-compare. [Online] 2023. 09 01. https://text-compare.com/.
[10] MOSS. [Online] 2023. 08 15. https://theory.stanford.edu/~aiken/moss/.
[11] OpenAi. [Online] 2023. 08 18. https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text.
[12] *Measurement of Text Similarity: A Survey.* Wang, Jiapeng és Dong, Yihong. 421, hely nélk. : MDPI, 2020., 11. kötet.
[13] glob. [Online] 2023. 08 26. https://docs.python.org/3/library/glob.html.
[14] python-pptx. [Online] 2023. 09 02. https://python-pptx.readthedocs.io.
[15] python-docx. [Online] 2023. 09 02. https://python-docx.readthedocs.io.
[16] openpyxl. [Online] 2023. 09 02. https://openpyxl.readthedocs.io.
[17] agarwalkeshav8399. [Online] 2023. 09 02. https://www.geeksforgeeks.org/measuring-the-document-similarity-in-python/.
[18] *Document similarity for error prediction.* Marjai, Péter, Lehotay-Kéri, Péter és Kiss, Attila. 4, hely nélk. : Taylor & Francis, 2021., Journal of Information and Telecommunication, 5. kötet, old.: 407-420.
[19] Paksi, Dávid és Csóka, Márk. Document_smiliarity. *GitHub.* [Online] 2023. [Hivatkozva: 2023. 09 03.] https://github.com/JSelyeUniversity/Document_similarity.