2023. 04:

- Formai követelményeket újra átnézni
- ✓ Hivatkozás számok elé szóközöket tenni
- ✓ Ahol százalékos értékek vannak konkrét számokat megemlíteni

2023. 01:

- ✓ Könyvek linkjeit irodalomjegyzékbe
- ✓ Szavak darabszáma könyvenként
- ✓ Kódpéldákat frissíteni a szavak száma dolog miatt
- ✓ Kezdő oldalakon a képeknek nincs cím adva
- ✓ 7. oldalon nincs forrás megjelölve
- ✓ Online konvertert konkrétan megemlíteni az elején
- ✓ Régi módszer -> online konverter, új konverter -> kézi módszer (szóhasználat rossz)
- ✓ Megemlíteni, hogy lehetett volna fizetőssel is
- ✓ Tesseractot irodalomjegyzékhez felvenni
- ✓ Miért van az, hogy a nagyszó és kicsivel írt szó külön van (MNSZ-re hivatkozva)
- X Ugyan azon évfolyam tk vs szgy rangkorreláció vagy akár MNSZ-el szemben (10-10 szó pl. a leggyakoribb nem töltelékszó)
- ✓ Irodalomjegyzék számozásokat átnézni és sorrendbe rakni amikor minden fix már

2022. 09:

- Szófelhőknél mondjuk az első 15 szó gyakoriságát megemlíteni, több szám adat szerepeljen
- Az eredmények rész után egy összegzés részt csinálni
- Szakdolgozat elején hiányzó részeket kitölteni
- Átolvasni az egészet, szócserék, fura részek esetleges átfogalmazása

2022. 05. 04:

- MNSZ adathalmaz kérdés: Leírásba csatolni a manuális megoldás leírását, ne legyen csatolva a szakdolgozat fájlai közé a teljes adathalmaz
- Grafikon generálással érdemes foglalkozni, hátha több érdekes paraméter lesz látható vizuálisan.
- Szófelhős résznél szöveget csökkenteni. Konkrét számokra kitérni az MNSZ felhő javítás után

2022. 04. 13:

- Végeredmények rendberakása, érdekességek gyűjtése, a nem érdekes szavak rész minimalizálása
- Elektronikus melléklet rész befejezése
- Általánosságban keresni még szógyakorisággal kapcsolatos dolgokat

2022. 04. 06:

- Irodalomjegyzéket linkelni a szakdolgozat szövegein belül
- Ahol lehet irodalomjegyzéket konkretizálni
- Gyakoriságok utáni kutatás (nyelvtanulásnál szógyakoriság pl)

2022. 03. 23:

- Mások elemzéseiről írni a szakdolgozat elején (bevezetés)
- Végeredményeket elkezdeni megfogalmazni
- Kérdés: Végeredményeket milyen formában lenne érdemes prezentálni? (Mivel .json fájlokban vannak az adatok kimentve, esetleg wordben generálni hozzá grafikonokat? Szófelhők már vannak generálva könyvenként, könyvek összesítve + az MNSZ adataihoz is)
- Kérdés: Elektronikus melléklet részt kell írni? Ha jól értelmezem igen, mivel majd a kódot meg a generált dolgokat .zip-ben kell majd leadni

2022. 03. 02:

- Más elemzésekkel összehasonlításhoz kutatás (nyelvtudományi intézet)
- Szókincs: 2008-as kutatás szerint átlagosan a kiadott tankönyvcsalád kb. 40 ezer egyedi szót tartalmaznak 3-8. osztályig
  TODO: Ide egyedi szavak számáról lehetne pl. számokat gyűjteni a tesztadatból
  Ide érdekes táblázat a https://anyanyelv-pedagogia.hu/cikkek.php?id=771 linken látható táblázat a 'A szókincs becsült nagyságának alakulása az életkor előrehaladtával' táblázat
- http://www.nefmi.gov.hu/letolt/kozokt/tkvrendelet.ppt linken érdekes lehet a 16. oldalon szereplő tulajdonneves táblázat, bár mi nem számoltatunk ilyen adatokat. Itt még szó esik a pl. 100 betűnél hosszabb mondatok számáról, ami szintén érdekes lehet. Ezt akár mi is tudnánk mérni
- http://www.nytud.mta.hu/adatb/index.html Linken csomó adatbázis elérhető, köztük pl. az eddig már említett MNSZ-es adatbázis is
- http://www.nytud.hu/hhc/ Itt lehet könyvekre szerzőkre szavakra keresni
- Kérdés: Szófelhőkből mennyit tegyek a szakgodába?
- TODO: Szófelhőkről írni a vázlatba hogy mire jók mivel lesznek használva vizualizációként

2022. 02. 09:

- Tankönyvek szógyakorisági elemzések keresése, akár angolul akár magyarul
  - https://anyanyelv-pedagogia.hu/cikkek.php?id=771
  - http://www.nefmi.gov.hu/letolt/kozokt/tkvrendelet.ppt
    (Ez letölthető .ppt fájl)
  - https://efolyoirat.oszk.hu/00000/00011/00104/pdf/iskolakultura_EPA00011_2006_05_079-088.pdf
  - http://pedagogus.edia.hu/sites/default/files/public/2_4/Vidakovich_Vigh_Somi_Thekes_2013.pdf
- Kódokat dokumentálni

2022. 01. 27:

- Vázlatot bővíteni az új dolgokkal, átnézni az eddigieket
- Szövegkinyerési hiba statisztika: Kicsiben kezdeni, minél többféle hibát összeszedni

2022. 01. 13:

- Szövegkinyerési statisztikák a hibákról amiket a Tesseract okoz keresni & jegyzetelni
    - ‘nahát’ szó rendellenesség
    - Kérdés: Hogy lenne érdemes ezt csinálni pontosan?
      Én olyasmire gondoltam, hogy a régi szövegkinyerési módszerekkel venni 1-2 oldalnyi szöveget és azt összehasonlítani hiba típus és mennyiség alapján.
- Esetleg Stanford CoreNLP-t kipróbálni a szövegre, mi az amit ez tud és pl a magyarlánc nem:
    - dokumentáció & online futtatható verzió
    - A szófajok elemzése jól működik, tokenizálási különbségek előfordulnak.
    - A lemma annotáció valamiért nem működik helyesen: nem adja vissza a lemmákat. pl:
      A ‘szerződésről’ szóra a magyarlánc azt mondja, hogy a lemma az ‘szerződés’, míg a coreNLP azt mondja, hogy ‘szerződésről’, ami nem helyes. Természetesen a szófaj az mind a kettő programnál helyes.
    - Named Entity Recognition működik. pl:
      ‘Budapesten nagy a köd.’
      Ebben felismeri hogy a ‘Budapesten’ egy helyszínt jelöl
    - TODO: Idézet felismerő kipróbálása
    - Feltűnt 1 érdekesség a mondattagolással kapcsolatban: a coreNLP nem ugyanúgy darabolja mondatokra a bemeneti szöveget mint a magyarlánc.
      Pl: a címszavakat, amik végén nincs írásjel a coreNLP egybeveszi a következő mondat szövegével.
    - Kimentettem a kimenetét ugyan olyan formátumban, mint amit a magyarlánc ad -> amik a magyarláncra lettek írva elemző kódok akár erre is működhetnek (szófajonkénti statisztikák, szófelhők stb)
- Szófelhőknek utánanézni:
    - Szófelhőkről leírás, ebből lehetne jegyzetelni ha kellenek majd
    - A szófelhők generálását programkódból végzem, ugyan abban a lépésben amikor a statisztika .json fájlokat generálom.
    - Az eddig kinyert statisztikákból generálódik könyvenként & összesítve az 50 leggyakoribb szóra 1-1 szófelhő.
      Minta a 9.-es tankönyv top50 szavára
      Megfigyelés: Érdemes lenne kiszűrni pl. a névelőket, eddig csak az írásjeleket szűröm ki.

2021. 12. 16:

- Szógyakorisági elemzések, leggyakoribb igék, főnevek, melléknevek
- Töltelékszavak
- MNSZ szövegtár, ottani statisztikákkal összehasonlítani
- Eddigi kinyert statisztikák akár könyvenként vagy összesítve (jelenleg json fájlokban tárolva, előre kiszámolva):
    - Szófajonkénti darabszámok
    - Leggyakoribb ‘n’ szó hozzátartozó darabszámokkal és szófajokkal
    - Leggyakoribb ‘n’ szó szófajonként hozzátartozó szavankénti darabszámokkal
- Kérdés: a kinyert statisztikai adatokat milyen formában fogjuk használni?

2021. 12. 02:

- Adobe Acrobat Pro utánanézés szövegkinyeréshez:
    - Nem tudtam letölteni, ugyanis olyan adatokat kért amiket nem szerettem volna megadni egy ingyenes próbaverzióhoz... (kártyaszámok, lakcím, stb)
    - Találtam egy ehhez hasonló Adobe-s dolgot: ‘Adobe PDF Extract API’
      Ez json fájlba ment ki adatokat a pdf-ekről, viszont az ocr-nél többet tud: gépi tanulást használnak arra, hogy megértsék a struktúráját a pdf-nek és úgy nyernek ki belőle szöveget és egyéb adatokat (ez a módszer gondolom kiküszöböli pl. a hirtelen struktúra változásokban a pdf elrendezéseit illetve)
      Ezt sem tudtam sajnos kipróbálni, ugyanis ez is fizetős + regisztrációs oldal mögé van zárva.
- Új jegyzet
- Ezt jobban átolvasnom majd
- Lefuttattam az eddig általam írt programmal kinyert szöveges fájlokra a magyarláncot, 1 kivételével az összesre lefuttot. A 12.-es szöveggyűjteményből kinyert szövegben van valami amin a magyarlánc fennakad..

2021. 11. 11:

- Stanford NLP, mint alternatívum a magyarlánchoz más nyelvekhez
- Könyv készítőitől/kiadójától szöveges forrást kérni az elemezendő könyvekhez
- Új doksit létrehozni, amiben a kutatások részletesebben vannak írva?
- Nyers kimenet (Github gist link)
- Paragrafusonkénti kimenet (Github gist link, ezzel dolgozna a magyarlánc)

2021. 10. 28:

- Kutatni: stanford NLP és CoreNLP
- Kapott szakdolgozatok tanulmányozása
- Szövegkinyerés:
    - Mivel elég lassú volt a szövegkinyerés (pdf-ek képpé alakítása, majd azokra ocr), parallelizálás után a program egyszerre több pdf oldalt képes feldolgozni. Ez nagyon sokat segít a szövegkinyerés folyamat kimenetének a validálásán.
    - Új testreszabhatóságok: Bemeneti pdf-enként állítható margó, így kézzel állítható az a margó amin kívül nem történik szövegkinyerés
    - Ezekkel a módszerekkel már olyan kimenetet lehet kialakítani, ami egyszeri átfutással gyorsan elemezhető
    - A címek eltávolítása után egyesítem az összes szöveget 1 sorra (2 eset van: kötőjel + új-sor karakter, illetve új-sor karakter magában)
      Így már jól elemez a magyarlánc is

2021. 10. 14:

- Felhasznált dokumentumok:
    - https://aclanthology.org/R13-1099.pdf
    - https://rgai.inf.u-szeged.hu/magyarlanc
    - http://www.lrec-conf.org/proceedings/lrec2014/pdf/262_Paper.pdf

- Magyarlánc moduljai:
    - Sentence splitter (mondatokra bontás) and tokenizer
    - Morphological analyzer
    - POS tagger
    - Dependency parser

- Sentence splitter & tokenizer:
  MorphAdorner-be épített sentence splitter, bővítve. (pl. rövidítéseknél ahol pont van, de nem jelzi a mondat végét, pl: ‘kft.’)
  Emellett a tokenizáló sorszámozott tokenekre bontja a mondatokat, szegmentálás történik.
- Morphological analysis:
  Lemmatizáló szótári alakra alakítja az adott tokeneket, itt történik pl. a toldalék eltávolítás.
- POS tagging:
  Hasznos információkat tárol 1-1 szó pozíciója a mondatokban: pl. elárul arról dolgokat, hogy az adott szó környezetében milyen szavak találhatóak.
  Ehhez természetesen kell pl. az adott szó szófaja, ami a POS tagging része.

  Kérdések:
    - OCR-ről megéri-e kutatni? Itt lehetne beszélni magáról a technológiáról, digitalizálás előnyeiről/hátrányairól, illetve a szövegkinyerés folyamán talált hibákról/észrevételekről
    - Adathalmaz fixálása:
        - 9-12. osztályos adat elég e?
        - Mivel a szövegkinyerős problémák közül már nincs megoldhatatlan, érdemes-e már véglegesíteni az adatokat?

2021. 09. 23:

- Magyarlánc módszereinek kutatása
- MSZNY magyarlánc szakirodalom
- dblp-n keresni nevek alapján
- Magyar nyelvű magyarlánc leírást keresni, mások mit írtak erről a hibáról
- Elsősorban szógyakoriság elemzés lesz majd a végkifejlet
- Következő megbeszélés: 2021.10.14 08:45

2021. 09. 13:

- Szövegkinyerés folyamatán kell dolgozni, próbálgatni, tesztelgetni, hogy hogy mennyi kézimunka marad
- Következő megbeszélés: 2021. 09. 23 08:30

Szövegkinyerés:

- Szövegkinyeréshez átálltam 1 új módszerre: PDF-ből kép képzés, majd Tesseract-al való szövegkinyerés. (A régi megoldás a PDFBox nevű könyvtárba épített szövegfelismerő volt)
- Az előző megoldás kimenetével több probléma merült fel. Ezek között volt olyan ami automatikusan, volt olyan ami csak kézzel megoldható lett volna:
  - Speciális szimbólumok, amik zavarták volna az elemzés eredményeit. Ezeket szimpla szöveg lecseréléssel kilehetett volna szedni.
  - Nagyon gyakran fordult elő, hogy kihagyott betűket a kinyerő: pl. a dupla ‘t’-k helyére 2 db szóközt tett. Ezeket automatizálva javítani szinte lehetetlen lett volna. Főleg ez volt a fő ok amiért új irányokba kezdtem el keresgélni.
- Az új kinyerési módszerekkel ezek a hibák automatikusan kijavultak + maga a kimenet formátuma is jobban tagolt.

Magyarlánc:

- Körülbelül 5 oldalnyi szöveggel tesztelgettem dolgokat, hogy mik történnek az eddigi kimenettel.
- Észrevettem, hogy a több sorra tagolt mondatokat a magyarlánc nem elemzi teljesen helyesen.
  Erre a problémára példa:

  Rosszul formázott bemenet, 1 mondat 3 sorra tagolva:
  https://gist.github.com/Degubi/2128c546adeacd57448f63c4a358da65
  Kimenet:
  https://gist.github.com/Degubi/925fc4cdc4d053e57910cc62b22c7140

  Helyes bemenet, 1 mondat 1 soron:
  https://gist.github.com/Degubi/64981558ce50a96803624f5877be1e04
  Kimenet:
  https://gist.github.com/Degubi/05bc9d77dafab756c0636a30abe92279

  Látszik, hogy a 2 kimenet különbözik: pl. az ‘idegen’ szó elemzése teljesen eltér a 2 kimenetben, illetve a ‘,’-t teljesen egybevonta az ‘idegen’ szóval amikor a mondat több sorra volt tagolva.
- Erre megoldásnak azt találtam, hogy mivel az új szövegkinyerő paragrafusonként széttaglalja a kimenetet, 1-1 adott paragrafusban a szöveget 1 sorra tenni. (Ugyanis 1 soron lehet több mondat, azok jól elemződnek)
  Ezzel a módszerrel nem lesz gond a tagolásokkal + az eddigi 1-1 soros címek is helyesek maradnak.

- A ‘I. BEVEZETÉS AZ IRODALOMBA – MŰVÉSZET, IRODALOM‘ cím elemzésénél vettem észre, hogy a program külön mondatnak veszi az ‘I.’ és az azutáni részeket. Ennek az oka az lehet, hogy a római 1-es szám után van egy pont és azt hiszi a magyarlánc, hogy új mondat kezdődik. Ezt nem tudom, hogy ez gond lehet-e.
  Erre megoldás lehetne talán regexekkel a sorszámok utáni pontok eltávolítása.

Elemzés:

- Elkezdtem kiépíteni a kimenet modelljét. (Úgy gondolom, hogy a szövegkinyerési folyamaton még bőven lehet finomítani program ügyileg mielőtt a végleges kézzel történő átnézést megteszem, de szerintem érdemes elkezdeni gondolkodni a további lépéseken is)
  Jelenleg a magyarlánc ‘morphparse’ módjának kimenetével foglalkozok, nem tudom, hogy ő fog-e kelleni majd a továbbiakban.
- Jelenleg annyi történik, hogy a kimenet első oszlopait (az eredeti szó formát) egyesítem ismét mondatokká. Ez segít abban, hogy észrevegyem ha a mondatok nem jól darabolódnak, vagy esetleg olyan szavak kerülnek vele, amik hibásan lettek kinyerve. (A mondat több soron problémára is így akadtam rá)

2021. 09. 01:

- irodalom szöveggyűjtemény és a tankönyv különbségeinek elemzés
- 'magyarlánc' program elemzéshez, utánanézni
- eddig kinyert szöveget kéne tisztítani a programnak
- esetleg más konvertereket keresni jobb kimenettel

Adatforrások:
Általános iskolai irodalom könyvekhez nem találtam szöveggyűjteményeket, ezért a gimnázium 9-12-et gyűjtöttem ki:

- Irodalom 12. osztály:
  https://www.tankonyvkatalogus.hu/pdf/FI-501021202_1__teljes.pdf
  https://www.tankonyvkatalogus.hu/pdf/FI-501021201_1__teljes.pdf
- Irodalom 11. osztály:
  https://www.tankonyvkatalogus.hu/pdf/FI-501021102_1__teljes.pdf
  https://www.tankonyvkatalogus.hu/pdf/FI-501021101_1__teljes.pdf
- Irodalom 10. osztály:
  https://www.tankonyvkatalogus.hu/pdf/FI-501021001_1__teljes.pdf
  https://www.tankonyvkatalogus.hu/pdf/FI-501021002_1__teljes.pdf
- Irodalom 9. osztály:
  https://www.tankonyvkatalogus.hu/pdf/OH-MIR09SZ__teljes.pdf
  https://www.tankonyvkatalogus.hu/pdf/OH-MIR09TA__teljes.pdf

Szövegkinyeréshez:

- Szövegkinyeréshez szeretném elhagyni az online oldalak használatát és programkódbeli automatizálható megoldást keresni.
  (A korábban használt oldalnak nem találtam API-t + nem igazán konfigurálható a szöveg kinyerése)
- Ezzel a témával már vannak korábbi ismereteim, java-ban a PDFBox nevű könyvtárral már foglalkoztam ilyesmivel. Szerintem hasonlóval egy alapból jobban szűrt adatforráshoz lehet jutni.
  (Ugyanis sokkal jobb minőségű és testreszabhatóbb szövegkinyerési módszerei vannak az ilyen könyvtáraknak.)

Szöveg tisztítása:

- 2 úton lehet elindulni: a programkóddal való tisztítás, illetve a kézzel való tisztítás.
- Legalább olyan szintre akarok eljutni az automatizálással, hogy ne kelljen órákon át a kimenetet kézzel tisztítani és formázni.
- Automatizálásra első gondolat az volt, hogy regexek-el és egyéb szűrésekkel szedem ki azokat a szövegrészeket, amik nem kellenek.
  (pl.ha egy sor csak számból áll akkor az valószínűleg oldalszám stb...)
- Viszont mi lenne ha nem (csak) szűrnénk a beolvasott szöveget, hanem be se olvasnánk azt ami nem kell?
- A nem kelendő szöveg nagy része a margón található -> hagyjuk ki a margón lévő szövegeket, sose olvassuk be őket.
- Ez a szűrés után már csak a könyv elején és végén található oldalak okoznak gondot, de mivel azok csak 1x-1x szerepelnek akár már kézzel is megoldható a tisztításuk.
- Ezután még lehet szó azokról, hogy pl. a címeket, alcímeket is kiszűrjük-e.

Magyarlánc:

- https://rgai.inf.u-szeged.hu/magyarlanc
- A program parancssoros meghívásakor a bemeneti fájl az -input paraméter, a kimeneti fájlé pedig a -output
- A program a bemeneti szöveget először mondatokra bontja szét, majd az adott mondatot kifejezésekre bontja szét. (mivel nem csak szavakat elemez, hanem pl. vesszőket és kötőjeleket, ezért lényegében tokenizálásról van szó)
- Kimeneti formátum (parse mode esetén):
  A kimenet egy CSV-hez hasonló szöveges kimenet. 1-1 sorban 1-1 kifejezés adattagjai vannak, az adattagok tabulátor karakterekkel vannak elválasztva. Az egyes mondatokat 1-1 plusz újsor elválasztó karakter választja el.
- 1-1 elemzett kifejezés adattagjai:
  - Az adott kifejezés sorszáma a mondaton belül
  - Maga az elemzett kifejezés
  - Az adott kifejezés szótövesített alakja
  - Az adott kifejezés elemzése (pl. ige esetén az idő, hanyadik szám hanyadik személy stb…)

2021. 08. 26:

- Adatforrás: https://www.tankonyvkatalogus.hu/site/kiadvany
- Eredeti szövegkinyerő oldal: https://tools.pdfforge.org/extract-text