# Szövegelemző- és annotálórendszer fejlesztése

Önálló laboratórium

Írásbeli beszámoló

2020/21 II. félév

Bozó Bálint Vid

Konzulens:

Mészáros Tamás

(Méréstechnika és Információs Rendszerek Tanszék)

# Feladatkiírás:

## Célkitűzés

A feladat természetes nyelvű dokumentumok gépi elemzésére koncentrál, jellemzően tudáskinyerési és klasszifikációs céllal.

## A feladat részletei

Az alkalmazható módszerek és algoritmusok köre igen széles a nyelvi elemzéstől kezdve, a különféle tudásalapú és tanuló rendszereken keresztül a szótár-, vagy ontológia-alapú megoldásokig. A feladat testre szabása során a jelentkező kiválaszthatja azokat az eszközöket és módszereket, amelyekkel szívesen megismerkedne, illetve dolgozna, és közösen határozzuk meg az elkészítendő alkalmazás célját, funkcióit.

A potenciális alkalmazási területek tárháza is széles az üzleti intelligencia rendszerektől kezdve a történeti és irodalmi szövegek elemzésén keresztül az információbeszerző rendszerekig.

**Az érdeklődők részt vehetnek konkrét ipari projektekben is banki, üzleti intelligencia, orvosi és bölcsészeti területeken.**

## Lehetséges részfeladatok

A feladatkiírás sokféleképpen szabható személyre, például:

- információkinyerés webes adatforrásokból (weblapok, Facebook stb.),
- szöveggyűjtemény tudáselemeinek felismerése, annotálása, kinyerése és felhasználása (pl. útleírások térképre rendezése)
- szöveg összekapcsolása és bővítése tudásbázisokkal (pl. háttérinformációk megjelenítése a Wikidata segítségével)
- hangulatelemzés szövegekből (pl. termékértékelések osztályozása)
- szerzőség azonosítása
- és további remek hallgatói ötletek

## Szükséges kompetenciák - megszerezhető képességek

A feladat sikeres megoldásához alapvető programozói tudás (jellemzően Python, esetleg R, Java) szükséges, speciális (pl. NLP) algoritmusok és eszközök előzetes előismerete nem.

A feladat megoldása során gyakorlati ismeretek szerezhetők természetes nyelvű elemzők működéséről, mélytanuló (deep learning) rendszerek alkalmazásáról, megismerhetők fogalomtárak, ontológiák és különféle tudásreprezentációs eszközök (pl. RDF), és más MI eszközök és módszerek.

## A jelentkezés menete, mi várható a konzulenstől...

Ezen a lapon összefoglaltam, mire számítok a jelentkezőktől, illetve mi várható tőlem.

Konzulens: Mészáros Tamás (meszaros@mit.bme.hu)

# Bevezetés

## A feladat értelmezése

A feladatkiírásba sok nagyon érdekes és izgalmas feladat bele tudtam képzelni. Szerettem volna az önálló laboratóriumon viszont egy egyszerűbb, inkább az eszközök használatára fókuszáló feladatot megcsinálni, amiben kipróbálhatom magam, illetve megnézhetem tetszik-e ez a témakör.

Saját feladatnak a könyv és novella műfajmeghatározást választottam. A cél az, hogy legyen egy programom, mely bemenetként kap egy könyvet vagy novellát, kimenetként pedig visszaadja annak a műfaját.

Ezt a célt kétféleképpen lehet megvalósítani:

Heurisztikus megközelítéssel, mely szerint egy szabályrendszert hozok létre, amelyben a saját ismereteim alapján próbálom meghatározni, hogy miből találja ki, hogy melyik szöveg milyen műfajú.

Tanuló rendszer létrehozásával, amely egy sok könyvet tartalmazó tanító halmaz alapján megtanulja, hogy az adott műfajnak milyen sajátosságai vannak, és aszerint találja ki a beérkező könyv műfaját.

Az utóbbit választottam, mivel számomra érdekesebbnek tűnt.

## Ismerkedés:

A természetes nyelvű szövegek elemzéséhez számos nagyon hasznos és jól kidolgozott könyvtár létezik, így lehet miből válogatni. Én az egyik legnépszerűbb, használatra egyszerűbb, pythonban használható NLTK-t (Natural Language Toolkit) választottam, mely tartalmazza az összes olyan függvényt, mely egy ilyen feladathoz szükséges lehet, emellett nagyon sok segítség található hozzá az interneten.

Az NLTK támogatja több különböző nyelvnek a feldolgozását is, viszont értelemszerűen az angol nyelvhez van igazítva, magyar nyelvű szavakból nem tudja mindet megfelelően kezelni, így inkább angol nyelvű szövegekkel dolgoztam tovább.

Alapvető kihívás a természetes nyelvek feldolgozásánál, hogy hogyan változtassuk át olyan formátumúra a szöveget, ami a számítógépnek értelmezhető, és lehetővé teszi, hogy matematikai formában felírhassuk az adott problémánkat.

Ehhez az NLTK több lehetséges megoldást is kínál.

## Feldolgozás módjai :

**Tokenizálás[1]**
A nagy és egybefüggő szövegünket kisebb, könnyebben használható részekre, tokenekre kell darabolni. Feladattól függően lehet mondatokra, vagy különálló egybefüggő karaktersorozatokra, lehetőleg szavakra bontani azt. Mondatokat olyan esetekben érdemes választani, amikor számít valamilyen szinten a szövegkörnyezet, szavaknál inkább statisztika alapú feladatokat lehet végezni.

**Stop szavak és speciális karakterek**

Szavakra tokenizálás esetén több zavaró tényezővel is találkozunk, melyek megnehezítik a statisztika alapú szövegfeldolgozást. Sok olyan szó szerepel a szövegekben, melyeknek nincs lényeges jelentéstartalma a vizsgálatunk elősegítéséhez, mivel az majdnem minden szövegben általánosan megtalálhatóak, extra jelentést nem hordoznak. Ezek legtöbbször a névelők (a, the), segédigék (have, must), kötőszavak (and, then) és hasonlók. Ezeket stop szavaknak nevezik, az NLTK segítségével ezeket is ki lehet szűrni. Másik zavaró tényező a speciális karakterek, mint a pont és a vessző, ezek értelemszerűen nem segítik a szövegelemzést. A stop szavakhoz hasonlóan ezeket is ki lehet szűrni.

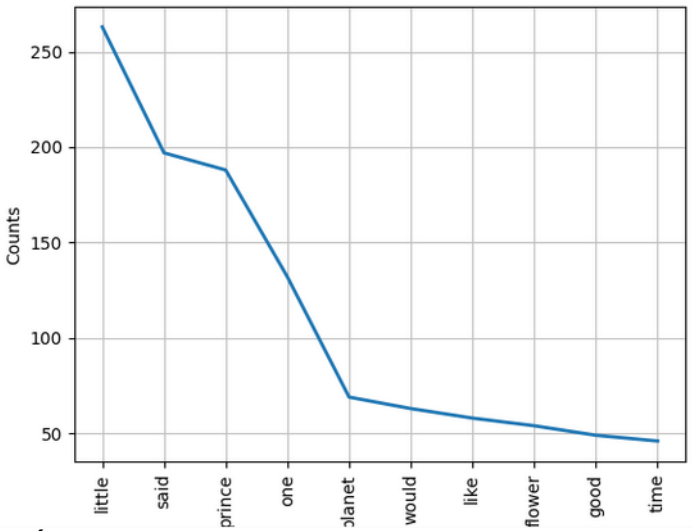
**Stemming – szavak normalizálása**

Egy adott szó több formában is előfordulhat a szövegben a ragozás különbözősége miatt, mely csak a szövegkörnyezettel együtt értelmezhető, viszont külön szavakra bontásnál csak bezavar. Ezért létezik a stemming, mely az adott szónak a ragozott formáit ugyanarra a szóalakra csonkítja le Például a „studies” és a „studying” szóból is „studi” szót csinál.

**Lemmatizálás**

A Stemminggel olyan formában megegyezik, hogy a szavakat leegyszerűsíti a könnyebb vizsgálat céljából, viszont azzal ellentétben a lemmatizálás szótári alakjára bontja vissza a szavunkat. Például a „studies” szóból „study” lesz, de a „studying” szóból „studying” marad, mivel szófajtól függően az is lehet különböző jelentésű szó.

Ezeknek az alkalmazásával már lehetőségünk is van kigyűjteni akár egy adott könyv leggyakoribb szavait is, amit az NLTK könyvtár segítségével vizualizálhatunk is.




*1. Ábra: A kis herceg című könyv leggyakoribb szavai*

**Part of Speech tagging**

Egy fentebb már említett probléma, a szavak szövegkörnyezettől való függősége. Sok olyan szó létezik, melynek több különböző jelentése van, de az alakja ugyanaz. Jó példa erre a magyar „vár”

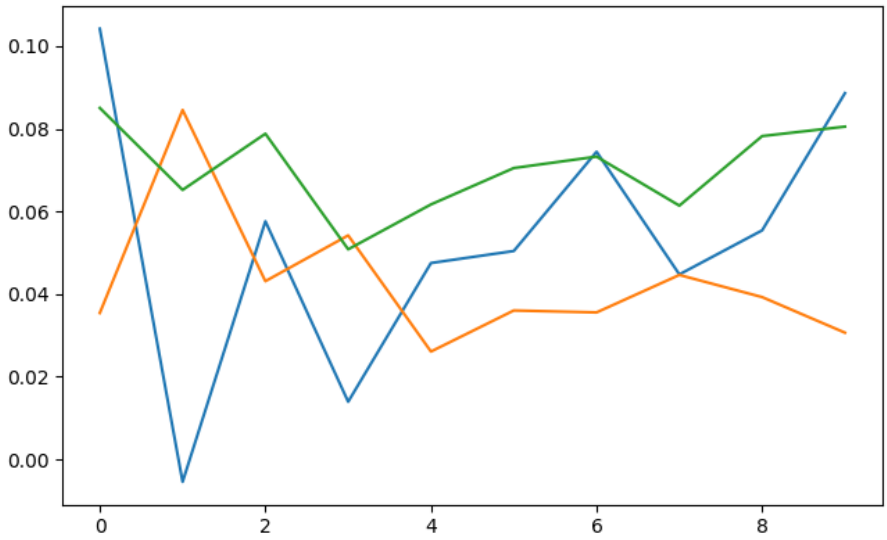
szó, mely egyszerre jelentheti az erődítményt, illetve a várakozás igét is. Erre nyújt bizonyos szintű megoldást a PoS, amely ismeri, hogy az angol nyelv szabályai szerint a mondatokon belül szórendet, illetve rengeteg szónak a szófaját is, és a kettő alapján eltudja dönteni a szónak a mondatban meghatározható szófaját.

A fentiekben megismert eszközökkel már egy működő programot el lehet készíteni, viszont még ismerkedés céljával kipróbáltam egy-két egyéb érdekes részét is az NLTK-nak.

**Vader**

A Vader egy hangulatelemző könyvtár. Twitter posztok alapján betanították neki, hogy mi számít pozitívnak hangulatúnak, ami a jóindulatú és boldog mondatokat jelenti, illetve negatívnak, ami ezeknek az ellenkezője. Mivel alapvetően ez pár mondatos szövegek alapján készítették, hosszú szövegek elemzésre nem alkalmas.

Ettől függetlenül, egy könyv mondatait ezzel is meg lehet vizsgálni annyi hiányossága van, hogy nem kezeli a mondatok közötti kapcsolatot. Készítettem egy példaprogramot, amellyel egy könyv hangulatgörbéjét meg lehet nézni Több mondat hangulatértékét kiátlagoltam, és abból készítettem vonaldiagramot. Ez alapján akár meg lehet vizsgálni azt, hogy egy adott műfajra létezik-e jellemző hangulatgörbe, esetleg írók specifikusan is lehetne így könyveket vizsgálni.[2]




*2. Ábra: Szintén A kis herceg könyvről. Fejezetenként látszódik, hogy hogyan változik a hangulata. A kék a negatív, sárga a pozitív, a zöld a kettő összefésülése*

**Word Cloud**

Legutolsó sorban egy inkább látványos, mint hasznos függvénnyel ismerkedtem meg. A Word Cloud megjeleníti az adott szövegben leggyakoribb szavakat, minél nagyobb egy szó, annál gyakoribb annak az előfordulása.

*3. Ábra: Szófelhő, ismét A kis hercegről*

# Önálló munka bemutatása

**Korpusz**

Miután megismerkedtem a lehetséges eszközökkel és lépésekkel, a feladatom elvégzéséhez első fontos lépés összegyűjteni egy szöveggyűjteményt, azaz egy korpuszt, amiben elég könyv szerepel ahhoz, hogy az alapján megtaníthassak egy programot a műfajfelismerésre.

Az alapvető probléma az, hogy a könyvek szellemi tulajdonnak számítanak, és az esetek döntő többségében nem ingyenes, így az interneten egészen kevés jelentős méretű könyvekből álló korpusz található.

A legnagyobb elérhető korpusz a Project Gutenberg [3], melybe közel 50 éve gyűjtenek szabadon elérhető könyveket, 2021-ben több, mint 60 ezer könyv található benne. Ezeknek a többsége lejárt szerzői jogú könyvek, melyek ingyenesen elérhetővé és terjeszthetővé váltak mindenki számára. Ennek a hátránya viszont az, hogy a könyvek nagy része, ami itt szerepel, nagyon régen íródott, egészen az első kiadott könyvektől a legkorábbi a XX. elejei könyvekig, mivel a szerzői jog lejárata a szerző halála utáni 70. évtől történik meg. Ez abban fog a későbbiekben nehézséget okozni, hogy mind a nyelv, mind a könyvek műfajainak a szókincse, illetve sokszor témája is megváltozott az évek alatt, így a jelenkori könyveket kisebb eséllyel fogja tudni megjósolni, egy ilyen könyvek szövegek alapján készült program.

A korpusz letöltéséhez egy külön könyvtárat használtam[4], mellyel letöltöttem a Project Gutenberg oldalán található mind a 60 ezer könyvet .txt formátumban. A weboldalon nem műfajonként vannak csoportosítva ezek a szövegek, hanem „könyvespolcokként”, amely nagyjából téma szerint csoportosítja őket, például gyerek mesekönyvek, ország szerint szétválasztott könyvek, madárhatározók, mikrobiológia könyvek stb. Átírtam egy már létező programot, amely az oldalak html kódjából kiszedi, hogy melyik könyv melyik polchoz tartozik, viszont ezeknek a többségében kevesebb, mint 100 könyv szerepelt, ami kevés ahhoz, hogy megbízhatóan megtanítsunk egy programot, illetve azt tudjuk tesztelni.

A könyvekhez szerencsére tartoztak témamegjelölő szavak a weboldalon, végül ezek alapján bontottam szét műfajokra az innen letöltött könyveket.

Itt választási lehetőség előtt álltam, hogy milyen műfajokat fogok megvizsgálni, vagy egyáltalán mit tekintek műfajnak. Az interneten található legtöbb könyv klasszifikációs példa az amerikai Library of Congress[5] katalogizáló rendszere szerint készítették, amelyben az irodalmi könyvek külön klasszifikációs egységnek számítanak a történelem könyvekkel, jogi könyvekkel és egyéb más szakmai és tudományos egységekkel szemben.

Én ezzel szemben az irodalmi műfajokra szerettem volna koncentrálni, így a legnépszerűbb jelenlegi irodalmi műfajok alapján válogattam szét, mellettük írva a jellemző szavakkal és könyvek számával:

| Műfaj | Darabszám | Jellemző szókapcsolat |
|---|---|---|
| Science-fiction | 2367 | „Science Fiction” |
| Detective | 990 | „Detective” „Mistery” |
| Fantasy: | 851 | „Fantasy” „Fairy tales” |
| Western | 543 | „Cowboy”, „Western Stories”, „Indians of North America -- Fiction” |
| Romantic | 831 | „Romance” „Love stories” |

Ezzel tehát meg is van a szükséges korpuszunk. Ezeket a halmazokat felosztottam tanító és teszthalmazra 4-1 arányban. A tanító halmazból tanulja meg majd a program, hogy melyik szó számít műfajspecifikusnak, a teszthalmazban pedig leellenőrizzük, hogy hogyan sikerült a betanítás. Ebből a szétosztásból 14 különböző verziót hoztam létre, hogy az esetleges kiugró értéket produkáló elosztások kiszűrhetőek legyenek.

**Modell**

A feladat megoldásához az egyik legegyszerűbb modellt választottam a Szózsák[5] (Bag of Words) modellt. Tulajdonképpen egy szótárról van van szó, amelyben minden szóhoz társítja annak az adott szövegben lévő előfordulási számát. Egyszerű létrehozni és dolgozni vele, viszont hátránya, hogy nem veszi figyelembe sem a szórendet, sem a nyelvtant.

Első nekifutásra, minden egyes könyvből egy szózsákot csináltam amelyben csökkenő sorrendbe tettem a szavakat gyakoriság szerint. A könyvek leggyakoribb szavainak egy részét összesítettem, és így készítettem el a műfajok szózsákját. Ez a megközelítés nem megfelelő eredményeket hoz később.

Ezután minden műfaj szózsákjában a szavakhoz súlyt rendeltem a gyakoriság alapján, olyan formában, hogy az a szó, ami az egész műfajban a leggyakoribb volt, az a legjellemzőbb az adott műfajra, és így tovább csökkenő sorrendben. Minden vizsgálandó, könyvnél összeszoroztam az adott könyv szavainak előfordulási számát a műfajonkénti szavak súlyával, és a legmagasabb műfajhoz tartozó érték alapján tippelte meg, hogy milyen műfajú az adott könyv.

Ez viszont értelemszerűen nem megbízható, és nem megfelelő eredményeket ad vissza.

Első probléma, hogy rengeteg olyan szó szerepel ezekben, mely egyáltalán nem műfajspecifikus. Például a „said”, a „man” és a „one” szó mind az öt műfajban a három leggyakoribb szavak között ott van, így teljesen felesleges vizsgálni ezeknek a szavaknak az előfordulását a könyvekben, nincsen releváns információs értékük.

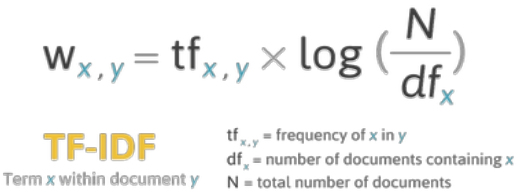
Második probléma a súlyozással van., melyben a szavak fontossága nem arányos a hozzátartozó súlyértékkel, így az félrevezető eredményt tud adni annak a használata.

Harmadik probléma, hogy a könyvben szereplő összes szavából vett leggyakoribb szavakat kezeltem. Ennek kétféle hátránya is van. Az egyik, hogy a leggyakoribb szavak egészen alacsony eséllyel hoznak ki műfajspecifikus szavakat. Előfordulhat az, hogy olyan szavak szerepelnek benne, amelyek az adott műfajban gyakrabban fordulnak elő, viszont azok a legtöbb esetben a többi könyvben is ott vannak, csak kisebb számban. Ez abban az esetben nem is okoz problémát, ha egyszerre több könyvet összevonunk, és együtt határozzuk meg a műfaját, viszont mi egyesével ellenőrizzük a könyveket, amiknél torzul az eredmény. A másik hátránya, hogy könyvek különböző hosszúak, így a hosszabb terjedelmű könyvek jobban befolyásolják a tulajdonságértékeket, mint a rövidek, ami szinté hibás irányba viszi el a megoldást.

**Műfajspecifikus szavak**

Következő fontos célom a műfajspecifikus szavak meghatározása volt, amelyek jellemzik az adott műfajt, illetve ezeknek egy releváns súlyozást adni. Erre kínál megoldást a széles körben használt tf-idf[7] (term frequency–inverse document frequency) algoritmus, amit pont erre találtak ki. Megadja, hogy egy dokumentumnak mik a legjellemzőbb szavai/kifejezései a korpuszhoz viszonyítva.

Ennek logikája egészen egyszerű.

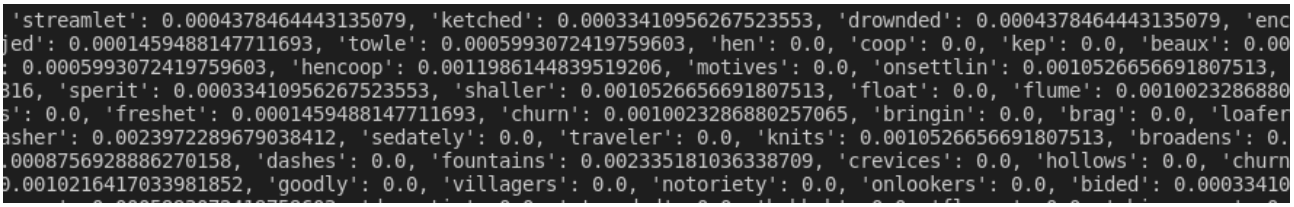



*1. Rajz: tf-idf képlete*

A term frequency része megadja, hogy egy adott kifejezés milyen fontos, azaz milyen gyakran fordul elő az adott dokumentumban. Ezt úgy éri el, hogy a vizsgálandó kifejezés előfordulási számát elosztja a dokumentumban lévő összes kifejezés számával.

Az inverse document frequency azt fejezi ki, hogy mennyi információt hordoz, avagy mennyire egyedi az adott kifejezés az egész korpuszhoz képest. Kiszámításának menete, hogy elosztjuk a dokumentumok számát azoknak a dokumentumok számával, ahol szerepel a szó, és ennek vesszük a tízes alapú logaritmusát. Az osztás értelemszerűen nagyobb értéket ad annak, ha egy szó kevés dokumentumban fordul elő, a logaritmus pedig segít abban, hogy ne szálljon el a szám mérete, könnyebben kezelhető legyen az eredmény.

Miután megvan a kettő, csak össze kell őket szorozni, és megkaptuk a megfelelő súlyozásunkat.




*4. Ábra: Rövid részlet az egyik műfaj tf-idf mátrixából*

**Tulajdonságvektor**

Következő lépésként minden műfajnál csökkenő sorrendbe rendeztem a szavakat a hozzátartozó tf-idf értékek alapján, és egy adott mennyiséget kiválasztottam belőlük, melyekből egy tulajdonságvektort alkothatunk. Ezeket összefésültem, hogy legyen egy nagy közös tulajdonságvektorom, amivel majd összevethetek az adott könyvem szavaival, illetve a könyvekben célzottan kereshetem ezeket a szavakat, nem kell az egész szöveget feldolgoznom, így a lefutási időn is gyorsíthatok.

Ezeket a lépéseket implementáltam is, viszont továbbra is követtem el hibát, melyek következtében nem megfelelően működött a program.

Több idegen nyelvű könyv belekerült a korpuszokba, például a romantikus műveken belül több francia könyv is szerepelt, aminek a szavait az algoritmus fontosnak ítélt meg, mivel azok csak annak a műfajnak a szavai között szerepeltek. Ennek a megoldása, hogy újra össze kellett állítani a korpuszt, kizárólag angol könyvekből. A fentebbi táblázatban leírt számok így álltak elő, azokban már nem szerepelnek a nem angol nyelvű könyvek.

A műfajok szavai, amikre alkalmaztam a tf-idf algoritmust, továbbra is az egy műfajba tartozó könyvek fix számú leggyakoribb szavai közül kerültek ki, ami a fentebb már leírt okok miatt itt is torzított. Ez kijött a teszteknél is, a műfajok meghatározásának sikerességének a szórása egyes műfajoknál 30%-ot is elért.

Ezt a problémát végül úgy lehetett áthidalni, hogy elengedtem a leggyakoribb szavas konstrukciót. A vizsgálandó könyvekből ezután egy fix méretű részt vizsgáltam, az 1000. szótól az 5000. szóig, amiből kigyűjtöttem az összes ott szereplő tokent. A tf-idf algoritmusnak a term frequency részével azt vizsgáltam, hogy műfajon belül mennyire gyakori az adott szó (hány dokumentumban fordul elő), az idf részben pedig azt, hogy az 5 műfaj közül mennyiben fordul elő az adott szó.

Ezek alapján az előző nekifutáshoz hasonlóan tulajdonságvektort hoztam létre, és abban szereplő szavakat kerestem a vizsgálandó könyvekben. Ha valamelyik szerepelt benne, a hozzátartozó tf-idf értékkel közelebb került az adott műfajhoz. Miután a teljes könyvet végigvizsgálta, megmondta, hogy melyikhez áll a legközelebb, és azt a műfaji besorolást tippeli neki az algoritmus.

Ennek a tesztelését lefuttattam, mind a 14 szétosztáson. Az eredmények 140 tulajdonságszó/műfajjal átlagosan a következők:

| Műfaj | Átlag találati arány | Szórás |
|---|---|---|
| detective | 67,6296% | 4,0675% |
| romance | 37,5449% | 4,1313% |
| fantasy | 57,5183% | 3,6319% |
| scifi | 95,1476% | 0,8917% |
| western | 96,3914% | 1,6570% |

Confusion mátrix:

| | detective | romance | fantasy | scifi | western |
|---|---|---|---|---|---|
| detective | 67,6296% | 6,2291% | 1,0101% | 16,2222% | 8,0909% |
| romance | 13,3535% | 37,5449% | 14,9700% | 15,5688% | 18,5628% |
| fantasy | 17.489% | 5.4878% | 57,5183% | 14,6341% | 4,8708% |
| scifi | 1,6897% | 0.4219% | 0,6316% | 95,1476% | 2.1092% |
| western | 0,5417% | 0% | 0% | 3,0669% | 96,3914% |

A szórásból látható, hogy a találati arány egészen konzisztens, ebben már nem szerepelnek olyan szintű torzítások, mint az előző próbálkozásokban.

Az átlag találati aránynál látszódik, hogy melyik műfajoknak vannak igazán műfajspecifikus szavai.

Nézzünk rá néhány, a műfajokat legjobban jellemző szavakra:

**scifi**:

'transcriber', 'spaceships', 'spaceman', 'earthman', 'venusian', 'blaster', 'spaceport', 'humanoid', 'galactic', 'interstellar', 'spacesuit', 'planets'

**western**:

'cowman', 'greasewood', 'mesa', 'corral', 'mustang', 'vaqueros', 'cowboys', 'cañon', 'broncho', 'arroyo', 'coulee', 'comanches', 'punchers'

**detective**:

'policer', 'marple', 'boathouse', 'poirot', 'riverview', 'criminologist', 'lecoq', 'swag', 'craig', 'oakvale', 'airport', 'mademoiselle', 'cheslow'

**romance**:

'toboso', 'rocinante', 'mancha', 'dulcinea', 'sancho', 'hamete', 'fanaticals', 'cide', 'ringbearer', 'enchanters', 'morisco', 'altisidora', 'panza'

**fantasy**:

'courtiers', 'glinda', 'oz', 'dwarves', 'ozma', 'munchkin', 'distaff', 'trolls', 'wroth', 'princeps', 'munchkins', 'mayst', 'quadling', 'lovecraft'

Látható, hogy a kiemelkedő eredményeket elérő műfajoknál, mint a scifi és a western, környezethez kötődő szavak lettek a jellemzők, míg a sikertelenebbeknél, mint a romance és a fantasy, nem ugyanolyan környezetben játszódnak, ezért olyan szavak nem is láthatóak ott, helyette viszont előjönnek a könyvsorozatok szereplőinek és helyszíneinek a nevei, mint a romance könyveknél a don quijote-ből Sancho Panza vagy a fantasynál Óz a nagy varázsló.

Bár itt nem látszik, a fantasy könyveknél jött ki olyan hiba, amit korrigáltam utólag, hogy az azt jellemző szavak döntő többsége arab volt, annak ellenére, hogy nem voltak arab könyvek a korpuszban, és miatt ennél is rosszabb eredményeket hozott. Mint kiderült, az 1001 éjszaka meséiből kb 30 könyv szerepelt a szövegek között, ami képes volt teljesen felborítani a műfajt jellemző szavak gyűjteményét.

Olyan esetekben, amikor kevesebb tulajdonság szót használtam a könyvek jellemzésére, kevéssel lett csak pontatlanabb az eredmény:

60 szó/műfaj:

| Műfaj | Átlag találati arány | Szórás |
|---|---|---|
| detective | 66,6531% | 2,1511% |
| romance | 31,9960% | 4,4082% |
| fantasy | 53,5167% | 4,8717% |
| scifi | 92,8270% | 1,4015% |
| western | 94,2507% | 2,4723% |

30 szó/műfaj

| Műfaj | Átlag találati arány | Szórás |
|---|---|---|
| detective | 72,4444% | 3,4244% |
| romance | 22,6546% | 3,3437% |
| fantasy | 47,5028% | 5,2015% |
| scifi | 89,8593% | 2,0286% |
| western | 87,4617% | 3,2073% |

## Fejlesztési lehetőségek:

Több továbbfejlesztési lehetőség is van, ami segíthet az eredmények javításában.

Az eddig megírt program nem vizsgálja a szövegkörnyezetet, vagy a szórendet. Ki lehetne bővíteni azzal, hogy ne csak az önálló szavakat vizsgálja a program, hanem szókapcsolatokat is, több szóból álló részleteket egyszerre, azaz N-gramokat.

Megadott szófajú szavakat vizsgálhatnánk a korpuszokban.

Tulajdonneveket ki lehetne szűrni a szövegekből.

Ki lehetne szűrni az olyan hibákat, mint az 1001 éjszaka meséi okoztak, hogy megvizsgáljuk, hogy a tulajdonságszavak nem ugyanabból a könyvből származtak-e, és ha igen, akkor azoknak a többségét elvetjük, mert nem az egész műfajt definiálják.

## Irodalomjegyzék:


[1] Natural Language Processing (NLP) with Python — Tutorial ,https://pub.towardsai.net/natural-language-processing-nlp-with-python-tutorial-for-beginners-1f54e610a1a0#7d22

[2] Sentiment Analysis Horror Books, https://www.kaggle.com/donyoe/sentiment-analysis/report

[3] Project Gutenberg, https://www.gutenberg.org/

[4] Standardized Project Gutenberg Corpus ,https://github.com/pgcorpus/gutenberg

[5] Library of Congress Classification Outline , https://www.loc.gov/catdir/cpso/lcco/

[6] Bag of Words, https://en.wikipedia.org/wiki/Bag-of-words_model

[7] tf-idf, https://en.wikipedia.org/wiki/Tf%E2%80%93idf