

# TDK-DOLGOZAT

CSERHÁTI RÉKA

SZEGEDI TUDOMÁNYEGYETEM  
TERMÉSZETTUDOMÁNYI ÉS INFORMATIKAI KAR  
SZÁMÍTÓGÉPES ALGORITMUSOK ÉS MESTERSÉGES INTELLIGENCIA  
TANSZÉK

# FEDŐNEVEK-ÁGENSEK NYERS SZÖVEGKORPUSZ ALAPJÁN

*Készítette:*

CSERHÁTI RÉKA  
Matematika BSc III. évfolyam

*Témavezető:*

DR. BEREND GÁBOR  
egyetemi adjunktus

SZEGED

2022

# Tartalomjegyzék

|                                                                      |           |
|----------------------------------------------------------------------|-----------|
| <b>Bevezetés</b>                                                     | <b>1</b>  |
| <b>1. A Fedőnevek játék</b>                                          | <b>3</b>  |
| 1.1. Játékszabályok . . . . .                                        | 3         |
| 1.2. Kapcsolódó irodalom . . . . .                                   | 3         |
| <b>2. Hasonló szavak, hasonló szövegkörnyezetek</b>                  | <b>5</b>  |
| 2.1. Együttelőfordulások . . . . .                                   | 5         |
| 2.2. PMI . . . . .                                                   | 6         |
| 2.3. Vektoros szóreprézntációk . . . . .                             | 7         |
| 2.3.1. Word2vec . . . . .                                            | 8         |
| 2.3.2. FastText . . . . .                                            | 9         |
| 2.3.3. GloVe . . . . .                                               | 10        |
| <b>3. Szavak kapcsolódása a Fedőnevekben</b>                         | <b>10</b> |
| 3.1. Szóasszociációk és együttelőfordulások . . . . .                | 10        |
| 3.2. Saját hasonlósági mértékeink . . . . .                          | 11        |
| 3.2.1. NPMI . . . . .                                                | 12        |
| 3.2.2. $NPMI^2$ . . . . .                                            | 12        |
| 3.2.3. NPMI-gráf . . . . .                                           | 13        |
| 3.3. Hasonlósági mértékeink összehasonlítása és értékelése . . . . . | 14        |
| <b>4. Lehetséges utalások pontozása</b>                              | <b>15</b> |
| 4.1. A pontozás célja . . . . .                                      | 15        |
| 4.2. Korábbi pontozófüggvények . . . . .                             | 15        |
| 4.3. Saját pontozófüggvényeink . . . . .                             | 16        |
| 4.3.1. <i>KoyyRestrict</i> . . . . .                                 | 16        |
| 4.3.2. <i>Harmonikus</i> . . . . .                                   | 17        |
| 4.3.3. <i>Harmonikus-hányados</i> . . . . .                          | 17        |
| <b>5. Tesztelés és eredmények</b>                                    | <b>18</b> |
| <b>Összegzés és további lehetőségek</b>                              | <b>23</b> |
| <b>Irodalomjegyzék</b>                                               | <b>24</b> |

## Absztrakt

A Fedőnevek napjaink egy igen népszerű társasjátéka, melyben fontos szerepet játszanak többek között a világról alkotott ismeretek és a játékosok közti kooperáció. Egy "kémfőnökként" játszó játékos, illetve ágens feladata olyan szavak (utalások) keresése, amelyeket egy emberi csapattárs bizonyos előre megadott szavak közül minél többhöz kapcsolódónak talál, de más megadott szavakhoz nem. Ezt a mai fejlett nyelvtechnológiai módszerekkel is kihívást jelent jól megvalósítani.

Tanulmányunkban kémfőnök-ágenseket hozunk létre négyféle szóhasonlósági mérőszám segítségével, melyek előállításához kizárólag nyers szövegtörzsekre van szükség. Létrehozunk új, együttlőfordulásokon alapuló hasonlósági mértékeket, amelyek a FastText szóbeágyazásvektorok szerinti hasonlóságnál nagyobb korrelációt mutatnak az emberek által érzékelt hasonlósággal. A célzások generálását a hasonlósági mértékek alapján négy különböző pontozófüggvénnyel végezzük. Az ágenseket két nyelven, angolul és magyarul teszteljük egy online játékban, valós, emberekkel folytatott játékok keretében. A konfigurációink felülmúlják a korábbi ágenseket a csak nyers törzseket használó módszerek között.

## Bevezetés

Játékokat nemcsak játszani, de kutatni is érdemes — erre enged következtetni az a rengeteg tanulmány, amelyeket különféle játékok vizsgálatából írtak, és írnak máig lankadatlanul többek között matematikusok, pszichológusok, filozófusok és mesterséges intelligenciával foglalkozó kutatók.

A mesterséges intelligencia kutatásának már régóta egyik központi témája a különféle (kombinatorikus, sztochasztikus, számítógépes) játékokat emberi szinten, vagy annál jobban játszó ágensok fejlesztése. Például az egyik első, a közvélemény számára is nagy hírértékű áttörés volt a mesterséges intelligencia kutatásában, amikor 1997-ben az IBM *Deep Blue* nevű, sakkozásra kifejlesztett szuperszámítógépe legyőzte az akkori legsikeresebb sakkozót, Garry Kasparovot [9].

Szintén híres eredmény, hogy a 2010-es évek második felében a DeepMind által megalkotott, megerősítéses tanulással fejlesztett *AlphaGo* programnak sikerült legyőznie az emberek között legsikeresebb Go játékosokat. Ez azért tekinthető nagy eredménynek, mert Goban a lehetséges lépések, stratégiák és mintázatok száma nagyságrendekkel meghaladja a sakkét, emiatt az emberi intuíciónak sokkal nagyobb szerepet tulajdonítanak ebben a játékban. A sakkban és a Goban viszont közös, hogy teljesinformációs kombinatorikus játékok – azaz a kezdőállástól a lehetséges végállásokba vezető lépéssorozatok egy véges fával leírhatók, és mindkét fél számára elérhető minden információ, így a játékosok tisztában vannak egymás lehetséges stra-

tégiáival. Ez a struktúra sokak szerint számítógéppel könnyebben modellezhető, így ágensek is könnyebben fejleszthetők hozzá.

Számos kutatás irányul másfajta játékokat játszó ágensek fejlesztésére is. Például backgammonra és pókerre is sikerült már az emberi teljesítményt meghaladó ágenst fejleszteni, pedig mindkettőben szerepe van a véletlennek, póker esetén pedig a játékosok által ismert információk sem egyeznek meg. Azonban az sem igaz minden játékra, hogy a döntési lehetőségek halmaza diszkrét: egy valós időben játszott videojátékban például nagyon sokat számít, hogy egy játékos folytonos időben pontosan mikor hajt végre egy akciót. A megerősítéses tanulás ilyen játékokra is alkalmazható; egy híres eredmény például, hogy a DeepMind *AlphaStar* ágense felülmúlta az emberi teljesítményt a *StarCraft II* videojátékban.

Ha viszont nem emberi játékosok ellen, hanem velük kooperálva kell egy ágensnek játszania, ahhoz elengedhetetlen, hogy az emberi játékosok viselkedését kellő pontossággal megtudja jósolni. Természetesen mélytanulási módszerekkel ez is megvalósítható, de csak akkor, ha szinte korlátlan mennyiségű adat áll rendelkezésünkre emberi játékokról, mert ilyen esetben megerősítéses tanulással, "önmagával játszva" az ágens nem tud olyan stratégiákat megtanulni, amelyek általánosan, a másik játékos stratégiájától függetlenül működnek. A Fedőnevek talán emiatt sem került — nemzetközi népszerűsége ellenére — a fősodratú kutatások középpontjába ezidáig.

Ebben a tanulmányban olyan Fedőnevek-ágenseket hozunk létre, melyekhez emberek által korábban lejátszott játékok adataira nincs szükség, de a teljesítményük emberi játékosokkal folytatott játékokban a véletlenszerű tippelést magasan meghaladja, és a korábbi hasonló módszereknél is jobban teljesít.

# 1. A Fedőnevek játéka

A Fedőnevek (angol nevén Codenames) egy szóasszociáción alapuló csapatjáték, amely 2015-ben jelent meg. A játékot Vlaada Chvátil tervezte, és már 38 nyelvre (köztük magyarra is) lefordították. A Fedőnevek 2016-ban megnyerte Az Év Játéka díjat, fent szerepel a Board Game Geek top 100-as, illetve a magyar Gém Klub top 50-es listáján is.

## 1.1. Játékszabályok

Az eredeti Fedőnevekben két csapat versenyez egymással. Egy 25 szókértől álló táblán vannak a kék vagy a piros csapathoz tartozó, illetve semleges kártyák, valamint egy azonnali vereséget jelentő kártya (fekete). Egy csapat akkor nyer, ha az ő csapatukhoz tartozó összes kártya felfedésre kerül a másik csapat kártyáinál korábban, vagy ha az ellenfél felfedi a fekete kártyát. Azt viszont, hogy melyik kártya milyen színű, mindkét csapatból csak egy-egy ember (a kémfőnök) tudja. Ezért a kémfőnökök minden körben egy-egy utalást adnak a csapatnak, mely egy utalószóból és egy számból áll. A csapat többi tagja (ügynökök) egymással egyeztetve sorban felfedik a táblán azokat a kártyákat, amelyeket az utalószóhoz kapcsolódónak gondolnak, amíg rossz kártyára nem tippelnek, vagy el nem éri a kémfőnök által számként megadott limitet.

## 1.2. Kapcsolódó irodalom

A Fedőnevek játékról szóló irodalom két fő, nem teljesen elválasztható kategóriába sorolható. Az első témájában leginkább kognitív modellezéshez tartozik, célja az emberi játékosok döntései, és ezen keresztül az emberi kognitív szemantikus lexikon megértése, modellezése. A másik irány célja, hogy a játékot emberrel játszani képes ágenseket hozzon létre különböző, főként számítógépes nyelvészeti módszerek segítségével, valamint ezáltal a szóreprezentációs eljárások és az emberi felfogás kapcsolatát tanulmányozza. Jelen dolgozat inkább ez utóbbihoz tartozik, bár a két irány nyilvánvalóan összefonódik, mivel végső soron mindkettő olyan modellt keres, amellyel az emberi döntések megjósolhatók.

Az emberi viselkedés modellezése itt elsősorban annak kiszámítását jelenti, hogy az emberek általánosan mely szavakat mennyire tartják egymáshoz kapcsolódónak. Ez a feladat meglehetősen kapcsolódik a pszicholingvisztikában régóta és sokat kutatott szóasszociáció-modellezéshez [26, 19], azonban semmiképp sem nevezhető vele ekvivalensnek. A szóasszociációs kísérleteknél az alanyoknak a lehető leggyorsabban kell adott szóhoz egy hozzá kapcsolódó tetszőleges szót megnevezni, de ebben az esetben a kémfőnök feladata egy olyan szó megtalálása, ami minél több szóhoz kapcsolódik a sajátjai közül, a többi szóhoz viszont nem, vagy jelentősen kevésbé szorosan. A feladatra szánható ideje is legfeljebb nagyon lazán (a többi játékos türelmében) van korlátozva, és — személyes tapasztalatok alapján — a kémfőnökök gyakran

több perc gondolkodási időt is felhasználnak a megfelelő utalás kigondolására. Emiatt előfordul, hogy az összekapcsolt szavak bonyolult módon, akár csak áttételesen függenek össze. Az ügynökök feladata — a tábla szavai közül az utalószóhoz kapcsolódók megkeresése — ennél jobban hasonlít egyszerű asszociációkhoz, de az idő itt sem lényeges, és az asszociációknak fontos korlátját képezi a táblán szereplő szavak száma. A Fedőnevek emiatt lehetőséget nyújt az emberi szemantikus memória vizsgálatára egy, a szóasszociációs kísérletektől és más klasszikus módszerektől különböző szempontból.

Shen és mtsai. [31] célja a — legjobb tudomásunk szerint — első Fedőnevekkel foglalkozó tanulmányban az emberi asszociációk modellezése volt. Az ő egyszerűsített játékukban a tábla mindig három főnévből áll, ezek közül előre meghatározott kettőhöz kell megfelelő utalószót találni, ami viszont csak három melléknév közül lehet az egyik. Az alábbi öt különböző szóhasználati mértéket felhasználva próbálták az emberi játékosok döntését modellezni:

- Bigramok (szövegben egymás után következő szavak) valószínűsége a szavak gyakoriságához viszonyítva,
- Skip-gram [20] és GloVe [28] szóbeágyazások szerinti koszinusz hasonlóság,
- A ConceptNet5 [32] tudásgráf szerinti kapcsolat,
- Topikmodellezéssel számított távolság.

Úgy találták, ezek közül a bigramok valószínűsége áll a legközelebb az emberek döntései szerinti szóhasználati mértékekhez.

Kim és mtsai. [16] célja már a játékot jól játszó ágensek építése volt. Metrikáik háttereként a

- CBOW, Skip-gram [20] és GloVe [28] szóbeágyazásokat (több konfigurációban),
- valamint a WordNet gráf alapú tudásbázist [22]

használták, utóbbit számos különböző távolságfüggvénnyel. Tanulmányukban azonban nem emberi adatokkal értékelik az ágensek teljesítményét, hanem kémfőnök- és ügynökágensek párosításával, amivel viszont csak két ágens működésének hasonlósága deríthető ki, az emberekkel való együttműködési képességüktől függetlenül.

Jaramillo és mtsai. [14] a távolságot a következő reprezentációk alapján számítják:

- Wikipédia-cikkekből és szótári definíciókból számolt TF-IDF hasonlóság,
- a szavak naiv-Bayes alapú osztályozása, illetve
- a GPT2 transzformer nyelvi modell [29] első rétegéből kinyert szóbeágyazások.

Ezen módszerek közül ők a GPT2 vektorait találják legalkalmasabbnak az emberekkel való együttműködésre.

Az egyik legfrissebb a témában Koyyalagunta és mtsai. [17] cikke, amelyben a korábban is használt Skip-gram és GloVe szóbeágyazások mellett a szavak távolságának modellezésére

- a FastText-et [6],
- a BERT modellt [13],
- valamint a BabelNet tudásgráfon [23] egy erre a célra kifejlesztett, szavakat speciális szabályok mentén egymáshoz rendelő keretrendszert

is felhasználnak. Az ő eredményeik szerint a FastText szerint koszinusz hasonlóságok bizonyultak a legjobb metrikának.

Továbbá Kumar és mtsai. [18] vizsgálta, hogy az emberi játékosok döntései megjósolhatók-e a Fedőnevek egy módosított változatában. Az előrejelzésekhez Skip-gram és GloVe szóbeágyazásokat, valamint több szabad asszociációs adatbázis (SWOW [12] és USF [24]) alapján számított hasonlóságot használtak. Úgy találták, a SWOW-ban a véletlen sétákon alapuló hasonlóság teljesített a legjobban, ebből arra következtettek, hogy ebben a játékban nem csak a közvetlen asszociációk, hanem a közvetett kapcsolatok is fontosak. Viszont minden, szóasszociációkon alapuló modell jobban teljesített a szóbeágyazásoknál, ami megkérdőjelezi a korábbi munkákat, amelyek nyelvtchnológiai szórepresentációk segítségével próbálnak Fedőnevek-ágenseket létrehozni annak ellenére, hogy ezek feladata nem az emberi asszociációk modellezése.

## 2. Hasonló szavak, hasonló szöveggörnyezetek

A leggyakrabban használt szórepresentációk közös vonása a modern nyelvtchnológiában, hogy a szöveggörnyezetük alapján karakterizálják a szavakat. Hogy ez lehetséges, azt egészen a J.R. Firth által híressé tett disztribúciós hipotézisig (1957) szokás visszavezetni: *"a word is characterized by the company it keeps"*.

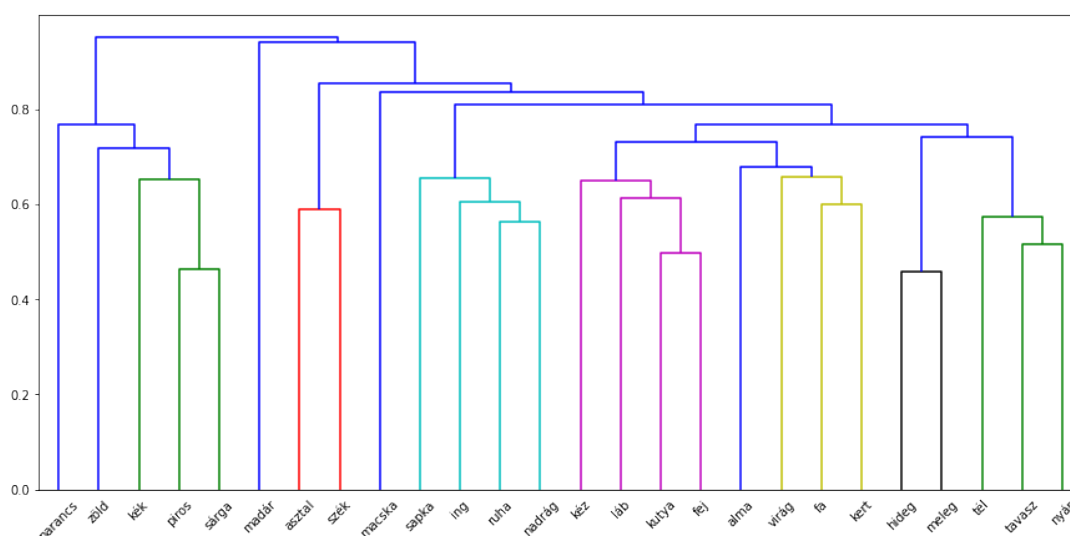
Ez a fejezet részben a szerző korábbi TDK dolgozatának felhasználásával készült [11].

### 2.1. Együttelőfordulások

A disztribúciós hipotézis közvetlen alkalmazása, hogy a szavak közti együttelőfordulások számát használjuk a szavak reprezentálására. Az együttelőforduláson alapuló modellek létrehozásakor először meghatározzuk, a szövegben mekkora környezetét szeretnénk számításba venni egy szónak. Ez a paraméter (window size – ablakméret) többnyire egyetlen szám, a célszótól általában jobbra és balra is ennyi szomszéd előfordulását számoljuk. Így egy  $|V| \times |V|$  méretű szimmetrikus  $C$  mátrixot kapunk (kookkurrencia-mátrix), amely tartalmazza az együttelőfordulási gyakoriságát az összes szónak minden másikkal:  $C_{i,j} = \text{count}(i,j)$ , ahol  $\text{count}(i,j)$  az a szám, ahányszor az  $i$ . szó szerepelt a  $j$ . szó "ablakában" (vagy fordítva, mert a két érték ugyanaz).

Az így kapott  $C$  mátrix oszlop- vagy sorvektorai önmagukban is hordoznak információt a szavak jelentéséről, és hasznosíthatók gépi tanulási módszerekben egy-egy szó vektoros repre-

zentációjaként. A példa kedvéért az 1. ábrán egy hierarchikus klaszterezés dendrogramja látható. Az ábrán látható szavak vektorai a szavak súlyozott együttelőfordulásai (a számok mindig a két szó távolságának reciprokával nőnek, legfeljebb 10 távolságig) a Magyar Webkorpusz 2.0 [25] közel 5000 leggyakoribb szavával, egység hosszúvá normalizálva. Ennyi információ alapján is, a vektorok távolságából képzett hierarchikus struktúra sok szempontból hasonlít az ábra szavainak emberi felfogás szerinti kapcsolataira. Többnyire az egy jelentésmezőhöz kötődő szavak kerülnek egy klaszterbe, és a klaszterek közötti kapcsolatok is meglepően jól reprezentálódnak a dendrogramon, például összekapcsolódnak a hőmérsékletek és az évszakok, tágabban pedig a természethez köthető szavak.



1. ábra. Hierarchikus klaszterezés dendrogramja együttelőfordulásokat tartalmazó szóvektorokon

## 2.2. PMI

A nyers együttelőfordulások hiányossága viszont, hogy nem jellemzi két szó tényleges összetartozásának mértékét. Ha egy adott szónak sok közös előfordulása van valamelyik másikkal, nem mindegy, hogy ez azért van, mert a másik szóval szoros kapcsolatban áll, vagy mert az a szövegben mindenhol megtalálható (pl. egy névelő). Mivel az első eset sokkal értékesebb információt hordoz, ahhoz szeretnénk nagyobb értéket rendelni, mint a másodikhoz.

Statisztikában és információelméletben elterjedt módszer a PMI (Pointwise Mutual Information) kiszámítása annak mérésére, mennyire különbözik két diszkrét véletlen változó együttes eloszlásában egy kimenetel valószínűsége attól, amelyet a változók függetlensége esetén tapasztalnánk. Legyenek ezek a véletlen változók  $X$  és  $Y$ ,  $X$  egyik lehetséges kimenetele  $x$ ,  $Y$

lehetséges kimenetele  $y$ . Ekkor az  $x$  és  $y$  közötti PMI-t a következőképp definiáljuk:

$$\text{PMI}(x, y) = \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)},$$

ahol  $p(x)$  a  $P(X = x)$  valószínűséget jelenti. A PMI 0 akkor, ha az  $X = x$  és az  $Y = y$  események függetlenek, és mennél távolabb állnak a függetlenségtől, az abszolút értéke annál nagyobb.

Szavak esetén a  $p(x)$  egy adott  $w_x$  szó előfordulásának (empirikus) valószínűsége,  $p(x, y)$  valószínűség pedig annak a valószínűsége, illetve előfordulási aránya, hogy két  $w_x, w_y$  szó egymáshoz egy bizonyos értéknél (ablakméret) közelebb szerepel a szövegkorpuszban. Emiatt a  $p(y)$ -nak annak a valószínűségnek kell lennie, ahányszor  $w_y$  bármely szóval egy ablakban szerepel. Két szó esetén akkor kapunk magas PMI-t, ha a szövegben nagy arányban fordulnak elő egymás közelében, tehát feltételezhetően jelentésbeli kapcsolat van köztük. Mivel csak a szövegben ténylegesen előforduló szavakkal dolgozunk, a  $p(x), p(y)$  empirikus valószínűségek pozitívak, és nem okoznak problémát a PMI nevezőjében.

A  $C$  együttlőfordulási mátrix sorait és oszlopait át tudjuk alakítani úgy, hogy az együttlőfordulások helyett a PMI értékeket tartalmazza, ekkor  $C'_{i,j} = \text{PMI}(i, j)$ . Ezzel a mátrixszal már valóban értékes információk birtokába jutunk a korpuszról és a szavakról, csak hogy a mátrix nagy mérete ( $|V| \times |V|$ , ahol  $|V|$  többszáz ezer lehet) nehézséget okoz az ilyen mátrixok tárolásában és feldolgozásában. Az ilyen nehézségek csökkentésének egyik módja, hogy a PMI-mátrixnak csak a pozitív elemeit tartjuk meg:

$$C''_{i,j} = \begin{cases} \text{PMI}(i, j), & \text{ha } \text{PMI}(i, j) > 0 \\ 0, & \text{ha } \text{PMI}(i, j) \leq 0. \end{cases}$$

Mivel fontosabb az, hogy mely szó párok fordulnak elő a várhatónál gyakrabban, mint hogy melyek ritkábban, ezzel a módosítással nem veszítünk értékes adatokat. Az így kapott ritka mátrixot a továbbiakban PPMI (Positive PMI)-mátrixnak nevezzük.

## 2.3. Vektoros szóreprezentációk

Bármilyen szövegfeldolgozási feladat esetén szükséges, hogy a rendelkezésre álló szövegeket az eredeti betűsoroknál kezelhetőbb formátumúvá alakítsuk. A szöveg szavakra vagy szótöredékekre bontásával keletkező egységeket érdemes valamilyen módon számok formájában tárolni és feldolgozni.

Ezt megtehetjük például a fenti módszerek segítségével, egy-egy szó vektorának tekintve az együttlőfordulásait, illetve a PMI értékeit más szavakkal. Ennek a hátránya viszont, hogy a vektorok dimenziója a szótár mérete lesz, ami általában többszáz ezres nagyságrendű, ez pe-

dig hatalmasra teheti az ilyen vektorokat felhasználó algoritmusok memóriaigényét. Emiatt az ilyen vektorokat általában dimenziószám-csökkentő eljárásoknak, például szingulárisérték-felbontásnak vetik alá.

Azonban az ilyen, együtelfordulásokon alapuló szóreprezentációkat mára szinte teljesen kiszorították a neurális hálók segítségével létrehozott, alacsony dimenziószámú szóreprezentációs vektorok. Ezekkel a módszerekkel, melyeket az angol word embedding kifejezés után szóbeágyazásoknak nevezünk, hasonló vagy jobb minőségű reprezentációk hozhazók létre sokkal kisebb számítási kapacitással.

### 2.3.1. Word2vec

A szóbeágyazások közül az első és talán legnagyobb áttörést Mikolov és mtsai. [20] 2013-as Continuous Bag-of-Words (CBOW) és Continuous Skip-gram (Skip-gram) algoritmusai jelentette. Az ezekkel együtt publikált word2vec nevű szoftver miatt a két modell word2vec néven vált ismertté, és még ma is az egyik legszélesebb körben használt szóbeágyazások közé tartozik.

A szavakhoz tartozó vektorok előállításához mindkét algoritmus egy egyszerű neurális hálót használ, amely egy bemeneti, egy rejtett és egy kimeneti rétegből épül fel. A legegyszerűbb esetben a modell egyszerre csak egy szót dolgoz fel. Ekkor a bemeneti réteg  $V$  értéket tartalmaz, ahol  $V$  a szótár mérete. Ezen  $\{x_1, x_2, \dots, x_V\}$  értékek között egy 1-es szerepel, azon az indexen, ami a feldolgozott szó indexe a szótárban, a többi érték 0.

A középső, rejtett réteg  $N$  értéket tartalmaz, ahol  $N$  a kapni kívánt vektorok dimenziója (többnyire 50 és 300 közötti szám). A rejtett réteg értékeit a bemeneti réteg alapján kapjuk úgy, hogy az összes bemeneti érték hatással van a rejtett réteg minden értékére, és a köztük lévő kapcsolat lineáris. Ezt felfoghatjuk úgy, hogy a rejtett réteg a bemeneti réteg és egy  $V \times N$  méretű  $\mathbf{W}$  mátrix szorzataként áll elő. Viszont mivel a bemeneti rétegben egy kivételével minden elem 0, a rejtett réteget egy bemenetre a mátrix egyik sorvektora határozza meg. Ennek a mátrixnak az optimális értékeit még nem ismerjük, ezek megtalálása csak később lesz lehetséges.

Hasonlóan, a rejtett réteg minden értéke lineáris kapcsolatban áll a kimeneti réteg egységeivel, ez tehát felírható a rejtett réteg és egy  $N \times V$  méretű  $\mathbf{W}'$  mátrix szorzataként.  $\mathbf{W}$  és  $\mathbf{W}'$  így az összes lehetséges bemenet esetén hozzárendel egy-egy  $u_j$  értéket a szótár szavaihoz. Innen elérhetjük, hogy ezek összege 1 legyen, így egy valószínűségi eloszlást alkossanak. Ezt megtehetjük a softmax függvénnyel a következőképpen:

$$p(w_j|w_a) = y_j = \frac{\exp(u_j)}{\sum_{i=1}^V \exp(u_i)}.$$

Ezáltal úgy tekinthetjük, mintha a  $w_j$  szóhoz társított érték mindig annak a valószínűsége lenne, hogy  $w_j$  lesz a szövegben következő szó  $w_a$  után. A modell betanítása során a célunk az, hogy a kimeneti réteg értékei valóban a fenti feltételes valószínűséget közelítsék. Ha ezt sikerül elérni,

$\mathbf{W}$   $i$ . sora és  $\mathbf{W}'$   $i$ . oszlopa a szótár  $i$ . szavának két különböző  $N$ -dimenziós reprezentációja lesz.

A betanításhoz lehet használni a neurális hálónál szokásos gradiens módszert. Eszerint a szövegen (általában többször is) végigmenve  $\mathbf{W}$  és  $\mathbf{W}'$  értékeit úgy állítjuk át, hogy a modell a következő szó valószínűségének becslését minél pontosabban végezze el. Ehhez először definiálnunk kell egy célfüggvényt, amelyet minimalizálni akarunk. Intuitív módon, ez lehetne minden iterációban minden szóra a jósolt valószínűség és a valós előfordulás (0 vagy 1) különbsége összegezve – a valóságban egy kicsit bonyolultabb függvényt, a predikált és valódi értékek eloszlása közötti keresztentrópiát használjuk. Ezután a klasszikus gradiens módszer szerint, a célfüggvény parciális deriváltjait felhasználva tudjuk minimalizálni a célfüggvényt, és megkaphatjuk  $\mathbf{W}$  és  $\mathbf{W}'$  optimális értékeit.

Csak hogy ehhez minden lépésben (minden feldolgozott szó után) a szótár összes szavának jósolt valószínűségét össze kell hasonlítanunk a valódi előfordulásával, majd ez alapján módosítani a hozzá tartozó vektort, ami nagyon költséges eljárás. Emiatt a gyakorlatban másként használjuk a gradiens módszert, és nem végezzük el a szótár minden szavára a hozzá tartozó vektor módosítását, hanem csak a szövegben valóban következő pozitív példá(k)ra, és néhány véletlenszerűen kiválasztott másik szóra (negatív példák). Ehhez egy valószínűségi eloszlásra is szükség van, amely alapján a negatív példákat generáljuk: a word2vec megvalósításában minden szó kiválasztásának valószínűsége a gyakoriságának  $\frac{3}{4}$ -edik hatványával arányos. A negatív példák száma szabadon választható, Mikolov és mtsai. [21] szerint a korpusz méretétől függően 2 és 20 között optimális. Ennek az egyszerűsítésnek köszönhetően a fenti célfüggvény helyett sokkal egyszerűbbet is használhatunk, és a betanításhoz jóval kevesebb számítási kapacitásra van szükség.

Természetesen lehetséges és előnyös is, ha a modell nem egyetlen szó alapján próbál megjósolni egy másikat, hanem figyelembe vesz több másikat is egy szó környezetében, és azok együttesen módosítják a két mátrixot. Ennek a word2vec esetén két megvalósítása is van: a CBOW egy nagyobb szöveggörnyezet alapján ad becslést egy szó előfordulásának valószínűségére, míg a Skip-gram egy szó alapján próbálja meghatározni az annak a környezetében szereplő szavakat. A tapasztalatok szerint a Skip-gram a CBOW-nál jobban teljesít a ritkább szavak reprezentálásában, ami nem meglepő, hiszen a CBOW a szöveggörnyezetben a ritka szavak figyelembe vétele helyett a többi szó felhasználásával is generálhatja a kimenetet; így lehetséges, hogy a ritka szóhoz tartozó vektor egyszerűen olyanná alakul, ami nem befolyásolja jelentős mértékben a kimeneti réteg értékeit. Ezért a Skip-gram vált elterjedtebbé a két módszer közül.

### 2.3.2. FastText

Születtek megvalósítások arra, hogy a word2vec-hez hasonló modellek a nyers előfordulásokon kívül más információt is tároljanak, például a szavak jelentésegyértelműsített alakját [35, 27]. Ezek viszont nem bizonyultak hasznosnak annyira, hogy megérje a modellek komplexitását

növelni miattuk, így nem kerültek be a köztudatba.

A szótöredékek alapján létrehozott vektorok viszont sikeres új irányt jelentettek, ezek közül a FastText [6] nyújtotta a legjobb teljesítményt, és vált a legnépszerűbbé. Ez egy olyan word2vec (általában Skip-gram) modell, amely betanításnál minden szót önmaga és a benne lévő 3-6 hosszú karakterláncok halmazával reprezentál. Az ilyen karakterláncokhoz és a szavakhoz is társít egy-egy vektort, és egy szó végső reprezentációja ezen vektorok átlaga lesz. Ez a módszer főleg a ritka és a morfológiailag összetett szavak esetén jelent előnyt. A FastText a legtöbb feladatban az egyszerű Skip-gramnál jobb modellnek bizonyult, és Koyyalagunta és mtsai. [17] eredményei szerint ezek közül a Fedőnevek-ágensek létrehozása sem kivétel. Ezért az itt bemutatott ágensek egyik távolságmértéke a FastText vektorok koszinusz hasonlósága, azaz két vektor bezárt szögének koszinusza lesz.

### 2.3.3. GloVe

Nem sokkal a word2vec után megjelent a GloVe (Global Vectors) [28] nevű modell is, melynek célja, hogy a word2vec és a korábbi számláláson alapuló módszerek előnyeit egyesítse. A GloVe az egész korpusz együttlőfordulásait dolgozza fel (innen a Global Vectors név), a vektorok betanítása pedig a word2vec-hez hasonlóan lokális környezet alapján történik. Ennek során a cél az, hogy ha  $w_i$  egy  $i$  szóhoz,  $\tilde{w}_k$  pedig ennek potenciális  $k$  szomszédjához tartozó vektor, akkor legyen

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(1 + \text{count}(i, k)),$$

ahol  $b_i$   $i$ -től és  $\tilde{b}_k$   $k$ -től függő konstans (bias), az 1 hozzáadására pedig a log 0 értékek elkerüléséhez van szükség.

## 3. Szavak kapcsolódása a Fedőnevekben

A Fedőnevek-ágensek létrehozásában a legfontosabb feladat a szavak közti távolságok modellezése úgy, hogy az az emberek szerinti távolságoknak a lehető legjobban megfeleljen. A korábbi tanulmányokban már számos nyelvtechnológiai módszert használtak e célra, azonban ezek közül kevés bizonyult rá valóban alkalmasnak. Mivel ezek a módszerek eredetileg nem az emberi asszociációk modellezésére készültek, ez nem is meglepő. A korábban használt módszerek közül általában a szóbeágyazások teljesítették a legjobban, viszont Kumar és mtsai. [18] szerint ezek is alulmaradnak az asszociációs adatbázisok alapján definiált távolságokhoz képest.

### 3.1. Szóasszociációk és együttlőfordulások

A szóasszociációkat régóta, számos okból vizsgálják pszichológiai, pszicholingvisztikai és nyelvészeti kutatások: a segítségükkel vizsgáltak bizonyos betegségeket, használták a memória és

kognitív folyamatok működésébe való betekintésre, valamint a kognitív lexikonnak és a nyelv bizonyos folyamatainak modellezésére is [5].

Kumar és mtsai. [18] eredményeiből következően valószínűleg Fedőnevek-ágenseket építeni is az asszociációs adatbázisok alapján lehetne a leghatékonyabban. Ennek viszont fontos korlátját jelenti az erre megfelelő adatbázisok meglétének igénye, ami miatt például magyar nyelvre sem lenne ez a módszer alkalmazható.

Emiatt számunkra nagyon fontosak lesznek Spence és Owens [33] eredményei, melyek megmutatták, hogy egy korpuszban a szavak együttlőfordulásainak mennyisége jól jelzi a köztük lévő szemantikai kapcsolat szorosságát, és asszociációk erősségének mérésére is alkalmas. Az eredményeik szerint a szavak szövegben való távolsága pedig negatívan korrelál a köztük lévő asszociáció erősségével.

Jelen munkánkban ezért a szavak kapcsolódását modellező távolságfüggvényt súlyozott kokkurrenciák alapján valósítjuk meg.

### **3.2. Saját hasonlósági mértékeink**

Spence és Owens [33] eredményei magyarázatot nyújtanak arra, hogy Shen és mtsai. [31] a bigramok valószínűségét találta a legalkalmasabbnak az emberi döntések előrejelzésére a játékban. Az is érthetővé válik, hogy más tanulmányokban pedig a szóbeágyazások jobban teljesítenek más típusú szóreprzentációknál, hiszen ezek is a szavak szöveggörnyezetét, azaz együttlőfordulásokat használnak fel.

A Fedőnevekben viszont, ahogy Kumar és mtsai. [18] eredményei is megmutatták, nem csupán a közvetlen asszociációk, hanem bonyolultabb, indirekt kapcsolatok is szerepet játszhatnak. Ennek figyelembe vételével a szavak közti kapcsolatokat — egy FastText modell mellett — szövegbeli együttlőfordulásokkal modellezzük, de a témában elsőként nemcsak két szó kölcsönös együttlőfordulásait figyelembe véve, hanem ezt többféle módon kiterjesztve közvetett kapcsolatokra is.

Az együttlőfordulások számlálására, valamint a FastText modell létrehozására egy angol és egy magyar korpuszt használunk. Angol korpuszként a teljes angol Wikipédia, valamint az angol OpenSubtitles korpuszokat használjuk összefűzve, hogy így a tudományos és hétköznapi szövegbeli sajátságok is megjelenjenek a korpuszban. Ez a korpusz összesen 5,692 milliárd tokenet tartalmaz. Magyar nyelvre a Magyar Webkorpusz 2.0 [25] lemmatizált változatát használjuk, mely szintén tartalmazza a teljes magyar Wikipédiát, és ezen kívül vegyes szövegeket az internetről, összesen 1,414 milliárd token nagyságban. Angolul 15 000, magyarul 10 000-es szótárméretet használunk, vagyis ennyi szót tartunk meg a leggyakoribbak közül, és mindkét nyelven eltávolítjuk az önálló jelentést nem hordozó, nagyon gyakori funkcionális szavakat (stopszavakat).

### 3.2.1. NPMI

Két szó asszociáltságának legelterjedtebb mérőszáma a fentebb bemutatott PMI. Ennek a mérőszámnak viszont hátrányai is ismertek, mint például a fix alsó és felső korlát hiánya, vagy a ritka szavak asszociáltságának felülértékelése. Ezek orvoslására Bouma [7] bevezette a normalizált PMI-t:

$$\text{PMI}_{\text{norm}}(x, y) = \left( \log \frac{p(x, y)}{p(x)p(y)} \right) / -\log p(x, y). \quad (1)$$

Vagyis a normalizált PMI-t úgy kapjuk, hogy a PMI értékét elosztjuk a két szó együttesfordulási valószínűsége logaritmusának  $-1$ -szeresével. Belátható, hogy ez a mérőszám a  $[-1; 1]$  intervallumban vehet fel értékeket, és a nevezőnek köszönhetően a ritkább szavak PMI-je általánosan kisebb lesz. Az érték előjele ugyanaz marad, mint PMI esetén, és az ott bevált módon itt is lehetséges a negatív értékek helyett 0-t megtartani ( $\text{PPMI}_{\text{norm}}$ ) — mi is ezt tesszük.

Szóbeágyazások kiértékelésére régebben szokás volt emberek által annotált, szópárok hasonlóságát vagy kapcsolaterősségét tartalmazó adatbázisokat is használni. Később kiderült, hogy az ezeken az adatbázisokon elért eredmények nem mindig függenek össze a valóságos alkalmazásokban való eredményességgel [10], hiszen az alkalmazásokban hasznos információk nem feltétlenül esnek egybe az emberek által érzékelt összefüggésekkel. Viszont ezek az adatbázisok hasznosak lehetnek számunkra, ha éppen az emberek által érzékelt összefüggéseket szeretnénk modellezni.

A normalizált PPMI-ből kapott kapcsolaterősségi mértéket összehasonlítva az ilyen adatbázisokban (MEN [8] és WS-353 relatedness [1]) található, emberektől szerzett adatokkal, azt tapasztaltuk, hogy a normalizált PPMI helyett annak négyzetgyökével jobban korrelálnak az emberi adatok. A PPMI négyzetgyökét véve a Pearson-féle korrelációs együttható 0.72-ről 0.76-ra emelkedik a MEN, és 0.57-ről 0.63-ra a WS-353 esetén. Emellett a most következő módszerekben előnyös, ha a hasonlóságok értékei nem koncentrálnak 0 köré, ezért ezután mindig a normalizált PPMI négyzetgyökét használjuk:

$$\text{NPMI}(x, y) = \sqrt{\text{PPMI}_{\text{norm}}(x, y)}. \quad (2)$$

### 3.2.2. NPMI<sup>2</sup>

Ahogy korábban említettük, a Fedőnevekben a kémfőnököknek a játékban való előrejutás érdekében olyan célzásokat kell adniuk, amelyek a célzott szavakkal akár csak távoli, közvetett kapcsolatban állnak (például az *ünnep* szó nem kapcsolódik közvetlenül a *fa* szóhoz, de mivel mindkettő erősen kapcsolódik a karácsonyhoz, az *ünnep* is jó célzás lehet a *fára*).

Az ilyen indirekt kapcsolatok modellezése érdekében a szavak közti NPMI értékekből kapott mátrixot, melynek átlóját mesterségesen csupa 1-esekkel töltjük fel, négyzetre emeljük, és az így kapott  $S'$  mátrix értékeit használjuk két szó kapcsolaterősségének mérésére.

Másképp,  $w_x$  és  $w_y$  szavak hasonlósága az ezen szavak más szavakkal való NPMI-jéből alkotott vektorok skalárszorzata:

$$\text{NPMI}^2(x, y) = \langle \mathbf{x}_{\text{NPMI}}, \mathbf{y}_{\text{NPMI}} \rangle,$$

ahol  $\mathbf{x}_{\text{NPMI}}$  vektor a  $w_x$  szónak az összes szóval kiszámolt NPMI értékét tartalmazza. Disztribúciós szemantikában ezt a hasonlóságot másodrendű hasonlóságnak is nevezik.

Egy másik szempontból tekintve, a skaláris szorzat definíciója szerint

$$\text{NPMI}^2(x, y) = \sum_{k=1}^{|V|} \text{NPMI}(x, k) \cdot \text{NPMI}(k, y),$$

vagyis ha  $G_0$ -t egy olyan gráfként definiáljuk, melynek szomszédsági mátrixa az NPMI mátrix, akkor  $S'_{ij}$  a  $G_0$ -ban  $v_i$  és  $v_j$  között található, kettő hosszú  $v_i - v_k - v_j$  utakon található súlyok szorzatának az összege. Mivel az átlóban lévő egyesek minden csúcsban egy 1 súlyú hurokét jelentenek, ezzel lényegében az 1 hosszú utakat, vagyis a közvetlen kapcsolatokat is figyelembe vesszük. Mivel minden élsúly 0 és 1 közötti szám, egy út súlyát a rajta lévő élsúlyok szorzatának tekintve továbbra is érvényes kapcsolódási mértékeket kapunk: a hosszabb utak, illetve azok, amelyeken kisebb súlyok (vagyis kevésbé összekapcsolódó szavak) vannak, kisebb súlyt fognak kapni.

Ez a módszer részben összefügg Artetxe és mtsai. [3] eredményeivel, akik megmutatták, hogy a szavak közti hasonlóság, illetve a szavak távolabbi összekapcsolódásának mérésére a szóbeágyazás-mátrixok különböző hatványai előnyösek. Ilyen esetben is az összekapcsolódás méréséhez nagyobb hatvány volt előnyös, mint a hasonlóság esetén.

Ennek a módszernek az is előnye (bár memóriahasználat szempontjából hátránya is), hogy növeli a mátrix 0-tól különböző értékeinek számát. Ez fontos egy ügynök-ágensnél például, mivel ha túl sok 0 érték van a mátrixban, előfordulhat, hogy az utalószó és a tábla szavai között minden hasonlóság 0 lesz. Az pedig egy kémfőnök-ágens esetén is fontos, hogy az utalószóhoz leginkább kapcsolódó szavak közül elég soknak ismerjük az utalószóhoz való közelségét.

### 3.2.3. NPMI-gráf

Az előbbi módszer során már definiáltunk egy kapcsolaterősségi mértéket, amely egy NPMI-értékekből létrehozott gráfot használ fel. Ebben egy út súlya a rajta lévő élsúlyok szorzata volt, és két csúcs között az összes ilyen út súlyának összegét vettük számításba. Ilyen módon egy nagyobb érték a két csúcs között erősebb kapcsolatot jelent.

Kapcsolatok gráfos modellezésekor viszont gyakoribb az az eset, amikor az élsúlyok távolságokat reprezentálnak, és egy út súlyát az élsúlyok összege jelenti, így az erősebb kapcsolatokhoz rendelünk kisebb értékeket. Mivel az NPMI értékek 0 és 1 közé esnek, definiáljuk a  $G$

|                         | NPMI  | NPMI <sup>2</sup> | Gráf  | FastText |
|-------------------------|-------|-------------------|-------|----------|
| <b>NPMI</b>             |       | 0.495             | 0.820 | 0.393    |
| <b>NPMI<sup>2</sup></b> | 0.349 |                   | 0.578 | 0.621    |
| <b>Gráf-távolság</b>    | 0.442 | 0.602             |       | 0.427    |
| <b>FastText</b>         | 0.295 | 0.524             | 0.319 |          |

1. táblázat. Pearson (felső háromszög) és Spearman (alsó háromszög) korrelációs együtthatók a kapcsolaterősségi értékek között.

|                         | MEN          |              | WS-353       |              |
|-------------------------|--------------|--------------|--------------|--------------|
|                         | Pearson      | Spearman     | Pearson      | Spearman     |
| <b>NPMI</b>             | <b>0.761</b> | <b>0.749</b> | 0.632        | <b>0.649</b> |
| <b>NPMI<sup>2</sup></b> | 0.627        | 0.670        | 0.502        | 0.545        |
| <b>Gráf-távolság</b>    | 0.754        | 0.735        | <b>0.650</b> | 0.647        |
| <b>FastText</b>         | 0.732        | 0.737        | 0.562        | 0.564        |

2. táblázat. Korrelációk az általunk definiált kapcsolódási értékek és az emberi annotációk között.

gráfot a következőképpen: egy él  $e(v_1, v_2)$  a  $w_1$  és  $w_2$  szavak között pontosan akkor létezik, ha  $\text{NPMI}(w_1, w_2) > 0$ , és ekkor a súlya  $w(e(v_1, v_2)) = 1 - \text{NPMI}(w_1, w_2)$ . Így a távolságot  $w_1$  és  $w_2$  között a  $v_1$  és  $v_2$  közötti legrövidebb úttal adhatjuk meg (kihasználva, hogy az NPMI maximális értéke 1, tehát az élsúlyok nemnegatívak):

$$d_G(w_i, w_j) = \min_{\pi \in \Pi_G(v_i, v_j)} \sum_{e_k \in \pi} w(e_k), \quad (3)$$

Ezeket pedig 1-ből kivonva távolságból a korábbiakhoz illeszkedő kapcsolaterősségi mértékké alakíthatjuk:

$$s_G(w_1, w_2) = 1 - d_G(w_i, w_j). \quad (4)$$

Így két erősen összekapcsolódó szó esetén, melyek között a legrövidebb út egyetlen él, az NPMI felső korlátjának köszönhetően visszkapjuk az NPMI értéket. Ez tehát a két fentebbi módszer kombinációjának is tekinthető, amely mindkettő előnyös tulajdonságaiból tartalmaz valamennyit.

### 3.3. Hasonlósági mértékeink összehasonlítása és értékelése

A fenti hasonlósági mértékek közti kapcsolat felderítése érdekében kiszámítjuk a korrelációkat a pontszámok között, amelyeket 100 000 véletlenszerűen választott szópárhoz rendelnek. Amint az 1. táblázatban látható, a mértékek közel sem ekvivalensek, de van köztük pozitív korreláció.

A szövegből kiszámított értékek magas korrelációt érnek el a MEN [8] és WS-353 [1] adat-

bázisok kapcsolaterősségi értékeivel is, ezt a 2. táblázat mutatja. Az NPMI és az NPMI-gráf szerinti hasonlóság ebben a FastTextet is felülmúlja, ami reménytelnek tűnik a Fedőnevek-beli alkalmazhatóságukra nézve.

## 4. Lehetséges utalások pontozása

Ha már vannak távolságfüggvényeink, amelyek két tetszőleges (az adatbázisunkban szereplő) szóhoz egy közelségi értéket rendelnek, ezzel tulajdonképpen az ügynök-ágenst is megvalósítottuk, hiszen adott utalószó esetén mindig azt a szót kell a táblán felfedni, amelyik ahhoz a függvény szerint a legközelebb áll. A kémfőnök-ágens létrehozásakor viszont még szükségünk van egy másik függvényre is, ami a lehetséges utalásokat valósághűen rangsorolja aszerint, hogy arra várhatóan hány jó tippet tud majd adni egy emberi ügynök-játékos.

### 4.1. A pontozás célja

Mondjuk azt, hogy az ágens a kék csapatban játszik, azaz a kék szavak egy részhalmazához (célzott szavak) kapcsolódó utalásokat szeretnénk generálni, természetesen a fenti távolságfüggvények alapján. Ha a távolságfüggvényeink jók, akkor minél közelebb van egy szó az utalószóhoz a távolságfüggvények szerint, *többnyire* annál nagyobb valószínűséggel tippelnek rá az ügynök-játékosok (bár a pontos kapcsolatot a közelség és a megfelelő tippelés valószínűsége között nem ismerjük, csak nagyon sok adat birtokában tudnánk róla pontosabban mondani).

Azt is figyelembe kell viszont venni, hogy a távolságfüggvények által adott értékek csak közelítései lehetnek az emberek szerinti tényleges kapcsolódási erősségeknek, így még ha a fenti feltevést igaznak tekintjük is, mennél kisebb a különbség két szó utalástól való távolsága között a mi távolságfüggvényünk szerint, annál nagyobb valószínűséggel lesz az emberi játékos szerint fordított a két szó sorrendje az utaláshoz való hasonlóságban. Ezért az utalásokat rangsoroló pontozófüggvényeknek két célt kell teljesítenie:

1. a célzott kék szavak legyenek minél közelebb az utalószóhoz,
2. a célzott kék szavak utalószóhoz való közelsége legyen egyértelműen nagyobb, mint a rossz (más színű) szavak utalószóhoz való közelsége.

### 4.2. Korábbi pontozófüggvények

A korábban bemutatott, Fedőnevek-ágensekről szóló munkák a szavak közti távolság számításán kívül az utalások pontozófüggvényeiben is eltérnek. Koyyalagunta és mtsai. [17] jelöléséhez igazodva, legyen  $\tilde{c}$  egy lehetséges utalószó,  $I_n$  a célzott szavak halmaza, vagyis a  $\tilde{c}$  utalószóhoz legközelebbi  $n$  db kék szó,  $R$  az összes, nem a csapathoz tartozó (rossz) szó halmaza, és  $s(\cdot, \cdot)$  a két szó hasonlóságát számító függvény.

Kim és mtsai. [16] pontozófüggvénye ekkor

$$g_{Kim}(\tilde{c}, n) = \begin{cases} \min_{b \in I_n} s(\tilde{c}, b), & \text{ha } \min_{b \in I_n} s(\tilde{c}, b) > \max_{r \in R} s(\tilde{c}, r) \\ 0, & \text{különben.} \end{cases} \quad (5)$$

Jaramillo és mtsai. [14] ugyanezt a függvényt veszik át, emellett kiegészítik egy veszélyességi súlyozással a kártyák színe szerint. Koyyalagunta és mtsai. [17] viszont definiálnak egy másikat is:

$$g_{Koyy}(\tilde{c}, n) = \left( \sum_{b \in I_n} s(\tilde{c}, b) \right) - \lambda \left( \max_{r \in R} s(\tilde{c}, r) \right), \quad (6)$$

ahol  $\lambda$  egy beállítható paraméter, ami a rossz szavaktól való távolságtartást szabályozza, a szerzők 0,5-et használnak.

Ezenkívül pedig bevezetnek egy másik módszert is, amellyel nem kizárólag a szóhasználatok alapján pontozzák az utalásokat, hanem a gyakoriságuk, és a szótári definícióikból létrehozott vektorok (Dict2vec [34]) hasonlósága alapján is – de ez tulajdonképpen inkább a távolságfüggvény módosításának tekinthető.

Mindkét fenti célnak viszont ezek közül egyik függvény sem felel meg teljesen. A  $g_{Kim}$  függvény olyan utalószót választ, melyhez az 1. célnak megfelelően minden célzott szó közel lesz, viszont bár az utalószó távolsága a rossz szavaktól nagyobb lesz, mint a célzott szavaktól, a kettő közti különbség bármilyen kicsi is lehet, ami nem felel meg a 2. célnak. A  $g_{Koyy}$  függvényben ennek kijavítására szolgál a legközelebbi rossz szó pontszámának kivonása (egy  $\lambda$  súllyal megszorozva). Itt viszont, mivel a képletben elől szereplő összeg várhatóan sokkal nagyobb, mint a második tag (feltéve, hogy a  $\lambda$  nem nagy, de a szerzők javaslata szerint nem az), a függvény akár úgy is adhat magas értéket, ha az utalószóhoz legközelebbi rossz szó közelebb van az egyik célzott szónál. Ráadásul az 1. cél teljesülése sem garantált, mert az összeg egyetlen nagyon magas hasonlóság miatt is lehet nagy, így az sem biztos, hogy a többi célzott szó közel lesz az utalószóhoz.

Ennek ellenére a  $g_{Koyy}$  függvény teljesített jobban a gyakorlatban Koyyalagunta és mtsai. eredményei szerint, ezért mi ezt használjuk összehasonlítási alapként a saját függvényeinkhez.

### 4.3. Saját pontozófüggvényeink

#### 4.3.1. *KoyyRestrict*

Mivel Koyyalagunta és mtsai. [17] pontozófüggvénye nem garantálja, hogy minden célzott szó közelebb lesz az utalószóhoz minden rossz szónál, mi definiáljuk a *KoyyRestrict*-nek elnevezett függvényt, amely ettől csak annyiban különbözik, hogy 0 pontszámot rendel azokhoz az utalásokhoz, amelyekben ez a kritérium nem valósul meg:

$$g_{KoyyR}(\tilde{c}, n) = \begin{cases} g_{Koyy}(\tilde{c}, n), & \text{ha } \min_{b \in I_n} s(\tilde{c}, b) > \max_{r \in R} s(\tilde{c}, r) \\ 0, & \text{különben.} \end{cases} \quad (7)$$

#### 4.3.2. Harmonikus

A másik probléma a *Koyyalagunta*-függvénnyel az, hogy a benne lévő összeget egyetlen magas érték is naggyá teheti úgy, hogy közben a többi célzott szó nincs elég közel az utalószóhoz. Ráadásul ehhez az összeghez viszonyítva a kivont érték 0.5-es  $\lambda$  mellett túl kicsi lesz, így keveset számít a rossz szavak távolsága az utalószótól.

Ezekre a problémákra megoldást jelenthet, ha az összeget (ami az optimalizás szempontjából adott  $n$  esetén a számtani középpel ekvivalens) a célszavak utalószóhoz való közelségének harmonikus közepével helyettesítjük, hiszen a harmonikus közép ismert tulajdonsága, hogy a számtani középpel ellentétben "a kisebbek felé húz", kevés nagy érték miatt nem vehet fel bármilyen nagy értéket. Tehát ahhoz, hogy a harmonikus közép, így a *Harmonikus* pontozófüggvény által egy utalásra adott pontszám magas legyen, minden célzott szónak közel kell lennie az utalószóhoz. Ezt a pontozófüggvényt tehát a következőképpen definiáljuk:

$$g_H(\tilde{c}, n) = \begin{cases} H(\{b | b \in I_n\}) - \lambda \cdot \max_{r \in R} s(\tilde{c}, r), & \text{ha } \min_{b \in I_n} s(\tilde{c}, b) > \max_{r \in R} s(\tilde{c}, r) \\ 0, & \text{különben,} \end{cases} \quad (8)$$

ahol  $H$  egy számhalmaz harmonikus közepét adja meg:

$$H(\{x_1, x_2, \dots, x_n\}) = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}.$$

A  $\lambda$  értékét a tesztelésnél megtartjuk 0,5-nek, fontosabbnak tartva azt, hogy a célzott szavak közel álljanak az utalószóhoz, mint azt, hogy a rossz szavak távol legyenek tőle. A két tag nagysága így is hasonlóbb, mint a *Koyyalagunta*-függvény esetében.

#### 4.3.3. Harmonikus-hányados

Végül pedig a fenti függvénynek létrehozunk egy másik verzióját is, melyben a legközelebbi rossz szó utalástól való távolsága és a harmonikus közép különbsége helyett e két szám arányát használjuk:

$$g_{HH}(\tilde{c}, n) = \frac{H(\{b | b \in I_n\})}{\max(\max_{r \in R} s(\tilde{c}, r), \varepsilon)}. \quad (9)$$

A következő fejezetben játékokban is teszteljük Fedőnevek-ágensekként az összes pontozófüggvényt az összes távolságfüggvénnyel kombinálva.

## 5. Tesztelés és eredmények

Minden távolságfüggvényt (FastText, NPMI, NPMI<sup>2</sup>, NPMI-gráf) minden pntozófüggvénnyel (*Koyyalagunta*, *KoyyRestrict*, *Harmonikus*, *Harmonikus-hányados*) összepárosítva összesen 16 ágenszt kapunk. Minden ágenssel generálunk utalásokat  $n = 2$ , illetve 3 célzott szóra úgy, hogy (a táblán lévő szavakat kivéve) a szótár összes szavát lehetséges utalásnak tekintjük. Mivel minden szóra egyszerű műveletekkel meghatározható a célzott szavak halmaza és a pontszám is, ez nem költséges eljárás.

Koyyalagunta és mtsai. [17] módszerét követve, a játék tábláit 20 szóból állítjuk össze, melyek közül 10 kék (azaz felfedendő) és 10 rossz (ezek között nem teszünk különbséget, nincs fekete kártya). Ilyen táblákból előre létrehozunk 100-at, az eredeti Fedőnevek játék szavai közül véletlenszerűen sorsolva, és mindegyikre generálunk utalásokat a fenti 32 konfiguráció felhasználásával. Ez 1304 különböző utalást eredményez angolul, magyarul pedig 1399-et.

A kiértékelést egy online játékban<sup>1</sup> végezzük, ahol a játékosok minden körben egy új táblát kapnak egy hozzá tartozó utalással véletlenszerűen, és be kell jelölniük a szerintük célzott szavakat. A játékosok nem ismerik az ágensek működését, és annak elkerülésére, hogy játék közben erre rátanulnak, a kör végén csak az általuk bejelölt kártyák színét fedjük fel.

Ilyen játékokból angolul összesen 443, magyarul pedig 1365 lejátszott kört gyűjtöttünk. Így az angol kiértékelésre ágensenként átlagosan 31.5, magyarra átlagosan 64 játék áll rendelkezésre. Egy kör lejátszásával a játékosok átlagosan 39 másodpercet töltöttek az angol játékban, míg a magyarban 37-et. Megjegyezzük, hogy mivel a magyar játék tesztelői minden bizonnyal magyar anyanyelvi beszélők, az angoloról viszont ez nem mondható el, a magyar adatokat tekintjük jobb minőségűnek.

Szintén Koyyalagunta és mtsai. [17]-hoz hasonlóan, a játékosok tippjeinek pontosságát egy ágenssel való játékban a következő módon számítjuk ki:

$$P@célzott = \frac{|I_n \cap U|}{n},$$

ahol  $I_n$  a célzott szavak halmaza,  $U$  pedig a játékos által megjelölt  $n$  szó.

Viszont a pntozófüggvények az utalásokat arra optimalizálják, hogy a rossz szavaktól legyenek távol, nem pedig a nem célzott kék szavaktól, amelyek így nagyjából olyan közel is lehetnek, mint a célzott szavak. Ha egy játékos ilyen, nem célzott szót jelöl meg, attól az ágens teljesítményét pozitívan kell értékelni. Ezért kiszámítunk egy másik pontosságot is, amelyben nemcsak a célzott szavakat, hanem az összes kék szót jó választásnak tekintjük:

$$P@mind = \frac{|A \cap U|}{n},$$

<sup>1</sup><http://fedonevek.herokuapp.com/>

| Értékelés | Távolság          | Koyy         | KoyyR        | HM           | HM-h         | Koyy         | KoyyR        | HM           | HM-h         |
|-----------|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|           |                   | 2 célszó     |              |              |              | 3 célszó     |              |              |              |
| P@mind    | FastText          | 0.764        | 0.757        | 0.740        | <b>0.829</b> | 0.710        | 0.712        | <u>0.756</u> | <b>0.759</b> |
|           | NPMI              | 0.747        | 0.747        | 0.776        | 0.715        | 0.707        | 0.708        | <u>0.733</u> | 0.695        |
|           | NPMI <sup>2</sup> | 0.722        | 0.742        | 0.725        | 0.744        | 0.666        | 0.696        | <u>0.746</u> | <u>0.729</u> |
|           | Gráf              | <u>0.795</u> | <u>0.795</u> | <u>0.827</u> | 0.715        | <u>0.727</u> | <u>0.735</u> | <b>0.759</b> | 0.695        |
| P@célzott | FastText          | 0.558        | <u>0.567</u> | <u>0.581</u> | <b>0.625</b> | 0.531        | 0.518        | <b>0.585</b> | 0.582        |
|           | NPMI              | 0.504        | 0.504        | 0.519        | 0.546        | 0.515        | 0.513        | 0.503        | 0.495        |
|           | NPMI <sup>2</sup> | 0.529        | 0.547        | 0.554        | 0.479        | 0.503        | 0.513        | <u>0.556</u> | <u>0.550</u> |
|           | Gráf              | 0.533        | 0.533        | <u>0.574</u> | 0.546        | 0.541        | <u>0.542</u> | 0.511        | 0.495        |

(a) Eredmények a magyar nyelvű játékban.

| Értékelés | Távolság          | Koyy         | KoyyR        | HM           | HM-h         | Koyy         | KoyyR        | HM           | HM-h         |
|-----------|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|           |                   | 2 célszó     |              |              |              | 3 célszó     |              |              |              |
| P@mind    | FastText          | <u>0.707</u> | <u>0.726</u> | <b>0.783</b> | <u>0.722</u> | <u>0.711</u> | <u>0.742</u> | <u>0.755</u> | <u>0.760</u> |
|           | NPMI              | <u>0.727</u> | <u>0.727</u> | 0.670        | 0.682        | <b>0.764</b> | <b>0.764</b> | <u>0.725</u> | <u>0.716</u> |
|           | NPMI <sup>2</sup> | 0.611        | 0.583        | 0.604        | <u>0.729</u> | 0.645        | 0.583        | 0.638        | 0.649        |
|           | Gráf              | <u>0.714</u> | <u>0.714</u> | 0.679        | 0.682        | <u>0.750</u> | <u>0.750</u> | <u>0.723</u> | <u>0.716</u> |
| P@célzott | FastText          | <u>0.487</u> | <u>0.535</u> | <b>0.581</b> | <u>0.555</u> | <u>0.549</u> | <u>0.495</u> | <b>0.577</b> | 0.520        |
|           | NPMI              | 0.420        | 0.420        | 0.397        | 0.426        | <u>0.549</u> | <u>0.549</u> | <u>0.541</u> | <u>0.508</u> |
|           | NPMI <sup>2</sup> | 0.377        | 0.361        | 0.372        | 0.445        | 0.354        | 0.369        | 0.370        | <u>0.470</u> |
|           | Gráf              | 0.392        | 0.392        | 0.384        | 0.426        | <u>0.552</u> | <u>0.552</u> | <u>0.533</u> | <u>0.508</u> |

(b) Eredmények az angol nyelvű játékban.

3. táblázat. Az emberi játékosok által adott helyes tippek aránya az online játékban. A kategória legjobb eredményének konfidenciaintervallumába tartozó számokat aláhúzással jelöljük.

|         |           |         |       |         |
|---------|-----------|---------|-------|---------|
| KÖZPONT | APRÓ      | HALÁL   | GUMI  | POHÁR   |
| BARÁT   | TORTA     | GYÉMÁNT | VONAL | KUTYA   |
| SZIKRA  | EGÉSZSÉG  | HATALOM | CIPŐ  | NŐVÉR   |
| BANDA   | TEHERAUTÓ | FORRÁS  | LEVES | SZAKASZ |

(a) Egy a kiértékeléshez használt tábla.

| Távolság          | Pontozás            | Szám | Utalószó        | Célzott szavak         |
|-------------------|---------------------|------|-----------------|------------------------|
| FastText          | Koyyalagunta        | 2    | macska          | kutya, halál           |
| FastText          | Koyyalagunta        | 3    | macska          | kutya, halál, cipő     |
| FastText          | KoyyRestrict        | 2    | végpont         | szakasz, vonal         |
| FastText          | KoyyRestrict        | 3    | fázis           | szakasz, vonal, forrás |
| FastText          | Harmonikus          | 2    | végpont         | szakasz, vonal         |
| FastText          | Harmonikus          | 3    | bukás           | halál, hatalom, banda  |
| FastText          | Harmonikus-hányados | 2    | összekötő       | vonat, szakasz         |
| FastText          | Harmonikus-hányados | 3    | korruptió       | hatalom, banda, forrás |
| NPMI              | Koyyalagunta        | 2    | esküvői         | torta, cipő            |
| NPMI              | Koyyalagunta        | 3    | esküvői         | torta, cipő, gyémánt   |
| NPMI              | KoyyRestrict        | 2    | esküvői         | torta, cipő            |
| NPMI              | KoyyRestrict        | 3    | esküvői         | torta, cipő, gyémánt   |
| NPMI              | Harmonikus          | 2    | korai           | szakasz, halál         |
| NPMI              | Harmonikus          | 3    | ellenséges      | vonat, hatalom, banda  |
| NPMI              | Harmonikus-hányados | 2    | korai           | szakasz, halál         |
| NPMI              | Harmonikus-hányados | 3    | rakpart         | vonat, szakasz, cipő   |
| NPMI <sup>2</sup> | Koyyalagunta        | 2    | macska          | kutya, banda           |
| NPMI <sup>2</sup> | Koyyalagunta        | 3    | ruha            | cipő, torta, gyémánt   |
| NPMI <sup>2</sup> | KoyyRestrict        | 2    | uralom          | hatalom, halál         |
| NPMI <sup>2</sup> | KoyyRestrict        | 3    | kegyetlen       | halál, hatalom, banda  |
| NPMI <sup>2</sup> | Harmonikus          | 2    | összekötő       | vonat, szakasz         |
| NPMI <sup>2</sup> | Harmonikus          | 3    | kegyetlen       | halál, hatalom, banda  |
| NPMI <sup>2</sup> | Harmonikus-hányados | 2    | meghosszabbítás | szakasz, vonat         |
| NPMI <sup>2</sup> | Harmonikus-hányados | 3    | bűncselekmény   | halál, banda, hatalom  |
| Gráf              | Koyyalagunta        | 2    | korai           | szakasz, halál         |
| Gráf              | Koyyalagunta        | 3    | esküvői         | torta, cipő, gyémánt   |
| Gráf              | KoyyRestrict        | 2    | korai           | szakasz, halál         |
| Gráf              | KoyyRestrict        | 3    | esküvői         | torta, cipő, gyémánt   |
| Gráf              | Harmonikus          | 2    | korai           | szakasz, halál         |
| Gráf              | Harmonikus          | 3    | ellenséges      | vonat, hatalom, banda  |
| Gráf              | Harmonikus-hányados | 2    | korai           | szakasz, halál         |
| Gráf              | Harmonikus-hányados | 3    | rakpart         | vonat, szakasz, cipő   |

(b) A fenti táblához generált utalások adott távolság- és pontozófüggvényekkel.

ahol  $A$  az összes kék szó halmaza.

A 3. táblázat mutatja a játékosok átlagos pontosságait az egyes ágensekkel való játékban. Minden (nyelv, kiértékelés módja és a célszavak száma által meghatározott) kategóriában egy 0,95 megbízhatósági szintű konfidenciaintervallumot konstruálunk a legjobb átlagos pontosságra *bootstrap* módszerrel, és az ebbe az intervallumba eső számokat a kategórián belül aláhúzással jelöljük.

Az összes konfiguráció közül a FastText hasonlóság és a *Koyyalagunta* pontozás kombinációja volt már értékelve (2 célszóra) *Koyyalagunta* és mtsai. [17] által, ahol ez volt a legjobb ágens nyelvspecifikus erőforrások felhasználása nélkül, azaz kizárólag nyers szövegtörzs felhasználásával. Az eredmények szerint ezt a mi konfigurációink közül több is felülmúlja.

A FastText hasonlóságon jelentőst javulást ér el a *Harmonikus* és a *Harmonikus-hányados* pontozófüggvény. A *FastText* + *Harmonikus-hányados* konfiguráció bizonyul a legjobbnak olyan tekintetben is, hogy ezek eredményei minden esetben beleesnek a legjobb átlagpontosság konfidenciaintervallumába. Ezenkívül csak egy kategóriában maradnak el a *FastText* + *Harmonikus*, illetve a *Gráf* + *Koyyalagunta* és *Gráf* + *KoyyalaguntaRestrict* konfigurációk.

Látható, hogy különböző távolságfüggvények különböző pontozófüggvények mellett teljesítenek jól. Ennek a pontos oka és mibenléte érdekes kérdés, mellyel a későbbiekben még tervezünk foglalkozni, de a harmonikus és a számtani közép különbsége biztosan szerepet játszik benne. A harmonikus közép például biztosan előnyös olyan esetekben, ahol a többihez képest kiugróan magas hasonlóságértékek lehetnek, ez pedig jellemzőbb lehet a FastText-re, mint a többi távolságfüggvényre.

A 4b. táblázatban lévő példákon is megfigyelhető a pontozófüggvények különböző viselkedése. Látható például a *Koyyalagunta* függvény általunk bevezetett szigorításának szükségessége: a *kutya* és *macska* szavak közötti magas hasonlóság miatt a *Koyyalagunta* függvény több távolságfajta esetén is a *macska* utalószót adja meg optimálisként, pedig ehhez ellenséges szavak is közelebb vannak, mint a második legközelebbi kék szó. A példák között viszont számos olyat is találunk, ami emberi játékosoktól származó utalásként is jónak számíthat: például ilyen az *esküvői* utalószó a *torta*, *cipő*, *gyémánt* célszavakra, a *kegyetlen* a *halál*, *hatalom* és *banda* szavakra, vagy a *korruptió* a *hatalom*, *banda*, *forrás* szavakra célozva.

Érdekes módon a 3.3. részben látott korrelációk a távolságfüggvények által adott és az emberileg annotált kapcsolaterősségek között nem jelzik előre a távolságfüggvények teljesítményét a játékban. Azokban a kísérletekben a FastTextet felülmúlta az NPMI és az NPMI-gráf szerint számolt kapcsolaterősségi érték is, míg az  $NPMI^2$  egyértelműen alulmaradt a többihez képest. Ezek a különbségek szinte egyáltalán nem mutatkoznak meg a Fedőnevek-ágensek teljesítményében.

A két különböző nyelven született eredmények között is megfigyelhetők különbségek. Például angolul az  $NPMI^2$  és az NPMI-gráf szerinti hasonlóság általában rosszabbul teljesít, mint

a két másik távolságfüggvény, ami nem figyelhető meg magyar nyelv esetén. Sejtésünk szerint ennek az oka az, hogy ezek a távolságmértékek jobban hordozzák az indirekt kapcsolatokat, amit a nem anyanyelvi beszélők nehezebben vesznek észre. Ezenkívül az is elképzelhető, hogy a korpusz méretétől is függ, hogy melyik távolságfüggvények alkalmazhatók jobban.

## Összegzés és további lehetőségek

Jelen munkánkban kísérletet tettünk a Fedőnevek társasjáték matematikai leírására egy kémfőnök-ágens feladatának két részre bontásával. Az emberekkel való együttműködéshez először egy olyan távolságfüggvény megadása szükséges, amely kellően jól közelíti az emberek megítélése szerinti kapcsolatokat, majd ehhez egy olyan pontozófüggvényt kell definiálni, amely a lehetséges utalásokat aszerint rangsorolja, hogy azokra hány jó tippet fog várhatóan egy emberi játékos adni.

Korábbi, asszociációkkal kapcsolatos kutatások alapján a távolságfüggvényeinket egy korpusz szavai közötti együttlőfordulások alapján hoztuk létre. Ezeket emberileg annotált kapcsolaterősségeken is kiértékeljük, melyekkel a FastText szerinti koszinusz hasonlóságnál magasabb korrelációt sikerült elérnünk két távolságfüggvénnyel is. Meglepetésünkre viszont ezek az eredmények nem bizonyultak prediktívnek a távolságfüggvényeknek a játékban elért teljesítményére.

A pontozófüggvények között is vezettünk be újításokat, egyrészt Koyyalagunta és mtsai. [17] pontozófüggvényének módosításával, másrészt pedig a harmonikus közép használatával, amely jobban megfelel a pontozófüggvényekkel szembeni *a priori* elvárásoknak, és az ágensek teljesítményét is jelentősen javította.

A legjobb ágenseink a FastText szerinti koszinusz hasonlóság és a harmonikus közép alapján működő pontozófüggvények, valamint az NPMI-gráf és számtani közép alapján működő függvények kombinációjaként jöttek létre. Ez felveti a kérdést, hogy pontosan min múlik az, hogy mely távolságfüggvényhez mely pontozófüggvény a legalkalmasabb, amivel a jövőben még tervezünk foglalkozni.

Továbbá szeretnénk adatokat gyűjteni emberi kémfőnök-játékosoktól, és ezeken tesztelni a ügynök-ágenseket, ami lehetővé teszi közvetlenül a távolságfüggvények optimalizálását.

Bár számos nyelvtechnológiai módszer került már felhasználásra a távolságfüggvények létrehozásához, vannak még olyanok, amiket érdemes lenne kipróbálni. Ilyen például az asszociációk gráfba vagy vektorokba foglalása [4, 5] és a GraphGlove [30]. Kumar és mtsai. [18] eredményei alapján egyértelműen érdemes asszociációs adatbázisokat felhasználni – ezt viszont az adatbázis eredeti nyelvétől eltérő nyelven megvalósítani nem könnyű, de nagyon érdekes feladatnak tűnik. Egy lehetséges megoldási irány lehet, hogy az asszociációkból létrehozott gráfhoz minél hasonlóbb gráfot próbálunk létrehozni szövegtörzsből, akár számlálás alapú, akár neurális módszerekkel, úgy, hogy utána ugyanez a módszer más nyelven is alkalmazható legyen. Másrészt számos módszer (pl. [2, 15]) létezik különböző nyelvű szóvektoros modellek összeillesztésére, elsősorban többnyelvű modellek létrehozása céljából. Így érdemes lenne megpróbálni, hogy egy asszociációkból létrehozott vektoros modellhez hasonlóvá tudunk-e tenni nyers szövegek alapján létrehozott vektoros modelleket, melyek akár más nyelvűek is lehetnek.

További érdekes irány még a különböző távolságfüggvények aggregálásából kapott új távolságfüggvények létrehozása. Ilyen módon létrehozhatunk például egyszerre nyers szövegeken, asszociációkon és tudásgráfokon alapuló távolságfüggvényeket is. A különböző távolságok kapcsolatának vizsgálata pedig hasznos tanulságokat tartogathat kognitív modellezésben és a neurális szóreprzentációk értelmezhetőségében is.

## Köszönetnyilvánítás

Köszönöm a témavezetőmnek, Berend Gábornak, hogy ezt a munkámat is minden lehetséges módon támogatta. Köszönöm Kolláth Istvánnak az online játék megvalósítását. Továbbá köszönöm mindenkinek, aki a játék tesztelésével segített a kiértékeléshez szükséges adatok összegyűjtésében.

Az Innovációs és Technológiai Minisztérium ÚNKP-21-1 - kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készült. A dolgozatban szereplő kutatási eredmények létrejöttét az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal is támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

## Hivatkozások

- [1] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado, June 2009. Association for Computational Linguistics.
- [2] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019, 2018.
- [3] Mikel Artetxe, Gorka Labaka, Iñigo Lopez-Gazpio, and Eneko Agirre. Uncovering divergent linguistic information in word embeddings with lessons for intrinsic and extrinsic evaluation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 282–291, Brussels, Belgium, October 2018. Association for Computational Linguistics.
- [4] Gemma Bel-Enguix. Retrieving word associations with a simple neighborhood algorithm in a graph-based resource. In *Proceedings of the 4th Workshop on Cognitive Aspects*

- of the Lexicon (CogALex)*, pages 60–63, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.
- [5] Gemma Bel-Enguix, Helena Gómez-Adorno, Jorge Reyes-Magaña, and Gerardo Sierra. Wan2vec: Embeddings learned on word association norms. *Semantic Web*, 10(6):991–1006, 2019.
  - [6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
  - [7] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. 2009.
  - [8] Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145, Jeju Island, Korea, July 2012. Association for Computational Linguistics.
  - [9] Murray Campbell, A. Joseph Hoane, and Feng hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1):57–83, 2002.
  - [10] Billy Chiu, Anna Korhonen, and Sampo Pyysalo. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6, Berlin, Germany, August 2016. Association for Computational Linguistics.
  - [11] Réka Cserhádi. Szóbeágyazás-modellek geometriai tulajdonságainak összehasonlító vizsgálata, 2020.
  - [12] Simon De Deyne, Danielle J Navarro, Amy Perfors, Marc Brysbaert, and Gert Storms. The “small world of words” english word association norms for over 12,000 cue words. *Behavior research methods*, 51(3):987–1006, 2019.
  - [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186, 2018.
  - [14] Catalina Jaramillo, Megan Charity, Rodrigo Canaan, and Julian Togelius. Word autobots: Using transformers for word association in the game codenames. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pages 231–237, 2020.

- [15] Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium, October–November 2018. Association for Computational Linguistics.
- [16] Andrew Kim, Maxim Ruzmaykin, Aaron Truong, and Adam Summerville. Cooperation and codenames: Understanding natural language processing via codenames. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pages 160–166, 2019.
- [17] Divya Koyyalagunta, Anna Sun, Rachel Lea Draelos, and Cynthia Rudin. Playing codenames with language graphs and word embeddings. *Journal of Artificial Intelligence Research*, 71:319–346, 2021.
- [18] Abhilasha A. Kumar, Mark Steyvers, and David A. Balota. Semantic memory search and retrieval in a novel cooperative word game: A comparison of associative and distributional semantic models. *Cognitive Science*, 45(10):e13053, 2021.
- [19] David McNeill. A study of word association. *Journal of Verbal Learning and Verbal Behavior*, 5(6):548–557, 1966.
- [20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.
- [21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [22] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [23] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225, 2010.
- [24] Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407, 2004.
- [25] Dávid Márk Nemeskey. *Natural Language Processing Methods for Language Modeling*. PhD thesis, Eötvös Loránd University, 2020.

- [26] David S Palermo and James J Jenkins. Word association norms: Grade school through college. 1964.
- [27] Maria Pelevina, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko. Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [28] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [30] Max Ryabinin, Sergei Popov, Liudmila Prokhorenkova, and Elena Voita. Embedding words in non-vector space with unsupervised graph learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7317–7331, 2020.
- [31] Judy Hanwen Shen, Matthias Hofer, Bjarke Felbo, and Roger Levy. Comparing models of associative meaning: An empirical investigation of reference in simple language games. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 292–301, 2018.
- [32] Robert Speer and Catherine Havasi. Conceptnet 5: A large semantic network for relational knowledge. In *The People’s Web Meets NLP*, pages 161–176. Springer, 2013.
- [33] Donald P Spence and Kimberly C Owens. Lexical co-occurrence and association strength. *Journal of Psycholinguistic Research*, 19(5):317–330, 1990.
- [34] Julien Tissier, Christophe Gravier, and Amaury Habrard. Dict2vec: Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, 2017.
- [35] Andrew Trask, Phil Michalak, and John Liu. sense2vec - a fast and accurate method for word sense disambiguation in neural word embeddings, 2015.