Neusch Gábor Loránd

Munkaerőpiaci adattárház tervezése kompetenciatrendek elemzésére

INFORMÁCIÓRENDSZEREK TANSZÉK

Témavezető: Borbásné Dr. Szabó Ildikó

BUDAPESTI CORVINUS EGYETEM

Gazdaságinformatika Doktori Iskola

Munkaerőpiaci adattárház tervezése kompetenciatrendek elemzésére

doktori értekezés

Neusch Gábor Loránd

Budapest

2021

Tartalomjegyzék

Ábrajegyzék

Táblázatok jegyzéke

Köszönetnyilvánítás

Köszönetet szeretnék mondani témavezetőmnek, Borbásné Dr. Szabó Ildikónak, aki a dolgozat elkészítése során folyamatosan támogatott mind szakmailag tanácsaival és javaslataival, mind emberileg egy-egy bíztató, baráti szóval. Rendkívül hálás vagyok az iránymutatásért, a támogatásért és a rengeteg segítségért Dr. Gábor Andrásnak. Nélkülük most nem tarthatná a kezében ezt a dolgozatot az olvasó.

Szeretnék köszönetet mondani Dr. Ternai Katalinnak és Dr. Fehér Péternek, akik bevontak az Információrendszerek Tanszéken folyó szakmai munkába, bizalmukkal támogattak és számtalan lehetőséggel segítettek az évek során.

Köszönettel tartozom továbbá Dr. Kovács Erzsébetnek a logisztikus regresszió területén, és Soltész Péternek a *Scikit-learn* programcsomag használatában nyújtott felbecsülhetetetlen segítségéért.

Hálával és hatalmas köszönettel tartozom páromnak, Paveszka Dórának, aki végtelen türelemmel és szeretettel támogatott a munkámban. Neki köszönhető, hogy mondataim – legalább részben – követik a magyar nyelv szabályait.

I'm extremely grateful for the inspiring friendship of Christian Weber. The curiosity, faith and love he shows for everything, completely changed my attitude and my thinking.

1 Bevezetés

1.1 Megbontó trendek a munkaerőpiacon

Globális szinten gondolkodó értelmiségieknek egy növekvő csoportja próbálja felhívni a figyelmet azokra a lehetséges negatív szcenáriókra, melyeket a mesterséges intelligenciának (*AI, MI*) és a gépi tanulásnak (*ML, GT*) egyre nagyobb térnyerése okozhat az élet minden területén. Bár néha jogosan hívják e gondolkodók némelyikét alarmistának, az általuk gyakran festett tudományos fantasztikumba illő katasztrófa-jövőképek miatt, azonban a felvetett forgatókönyvek közül egyik-másik valós veszélyeket, és ezekhez kapcsolódóan rengeteg megoldandó feladatot tartogat a közeljövőre. Ilyen például az, hogy az algoritmusaink egyre inkább képesek elvégezni azokat a feladatokat, melyeket tradicionálisan és egyelőre zömében jelenleg is emberek végeznek – és ráadásul mindezt, az esetek nagy részében gyorsabban, precízebben, hatékonyabban teszik.

Yuval Noah Harari izraeli történész sokat foglalkozott ezekkel a kérdésekkel bestseller könyveiben az elmúlt évtizedben. Utolsó könyvében, mely a "21 lecke a 21. századra" (Harari, 2018) címet viseli, a szerző előrevetíti az úgynevezett "haszontalan osztály" (*useless class*) megszületését és "felemelkedését". A haszontalan osztály tagjainak nincsen gazdasági „haszna", mivel nincs rájuk egyáltalán igény a munkaerőpiacon, így elveszít(het)ik társadalmi státuszukat és politikai befolyásukat is. Ennek okai pedig az olyan, jelenleg nagy sebességgel fejlődő technológiák, mint az MI és a gépi tanulás, és az egyre jobban teljesítő algoritmusok, mivel azok az általuk életre hívott megbontó innovációkon keresztül katasztrofális hatást gyakorolnak a munkaerőpiacra. Grundke és szerzőtársai osztják Harari nézőpontját, miszerint a jövőben a munkaerőpiacon jelentős változások fognak történni a digitális technológiák térnyerésének folytatódásával, ami azt fogja eredményezni, hogy egyes munkákat teljesen automatizálnak, míg mások esetében a feladatok és a munka természete fog jelentős változáson keresztülmenni (Grundke *et al.*, 2018).

Hasonló módon, mint ahogy a fizikai erőt igénylő feladatok egyre növekvő részét gépesítették az első ipari forradalom során, az egyre költséghatékonyabb termelés érdekében, a mesterséges intelligencia forradalma során azok a szolgáltató szektorban

dolgozók, akiknek a feladatai elsősorban kognitív készségeket, képességeket és tudást igényelnek, valamint könnyen algoritmizálhatók, elveszíthetik az állásukat, de legalábbis számolniuk kell az automatizáció által generált nagyívű változások hatásaival.

A kognitív képességeket és szaktudást igénylő feladatok jelentős hányada könnyen két részre bontható, melyek egyrészt a probléma analízise köré épülnek, másrészt az eredményeknek – az elemzésre épülő – értelmezése alapján, valamiféle döntés meghozatala során kerülnek végrehajtásra. Beck és Libert (2017) a következőképpen írja le az ilyen jellegű feladatok folyamatát.

1. Adatok összegyűjtése,
2. adatok elemzése,
3. eredmények értelmezése,
4. a szükséges intézkedések megállapítása,
5. a szükséges műveletek végrehajtása, implementáció.

Általánosságban kijelenthető, hogy az ilyen jellegű, jól algoritmizálható feladatokban a gépek jobban teljesítenek, mint az emberek, többek között azért, mert jobban skálázhatók, például igény esetén viszonylag egyszerű egy addicionális hardverelemet hozzáadni egy adatbázisklaszterhez. Továbbá az egyes gépi adattárak viszonylag könnyen naprakészen tarthatók, mindig tartalmazhatják egy adott szakterület legfrissebb, hozzáférhető eredményeit, míg egy emberi szakértő számára legalábbis nehéz, ha nem lehetetlen, hogy területének összes innovációjával lépést tartson. A gépeink ráadásul sohasem fáradnak el (adott esetben persze elromolhatnak, illetve a modellek esetleges pontatlansága miatt tévedhetnek), míg emberek esetében gyakran előfordulhat, hogy alvásmegvonás miatt katasztrofális hibát vétenek. Végül, de nem utolsósorban a gépek rugalmasabban, gyorsabban tanulnak bizonyos feladatok, például mintafelismerés esetében, míg egy embert behatárol a saját, szükségszerűen limitált tapasztalata és tudáskészlete, attitűdje, kognitív torzításai[1], sőt még a hangulata

[1] Mivel a legtöbb gépi tanulási algoritmus emberi inputok alapján „tanul”, ezért a kognitív torzítások és sztereotípiák egyelőre gyakran átszivárognak ebbe a szférába is. Jó példa erre a Google Fordító jelenlegi „nemi elfogultsága” (*gender bias*), például, hogy maszkulinnak ítéli a doktor szót és ha egy nemtől független nyelven beírjuk az „ő egy orvos” kifejezést, azt minden esetben „*he is a doctor*”-ként fordítja angolra. Ezzel ellentétben az „ő egy takarító” kifejezést az alkalmazás „*she is a cleaner*”-ként fordítja.

is (Beck és Libert, 2017). Jó példa az előzőekre az IBM által kifejlesztett, Watson névre keresztelt mesterségesintelligencia-rendszer, melybe orvostudományi kutatások és esetek millióit rögzítették 2011 óta, és 2016-ban került a média érdeklődésének középpontjába, amikor mindössze tíz perc alatt diagnosztizálta a leukémia egy ritka formáját egy 60 éves japán beteg szervezetében, amit az orvosok előzőleg több mint egy évig nem tudtak beazonosítani (Rohaidi, 2016).

Persze nem biztos, hogy a legrosszabb forgatókönyv bekövetkezik, azaz hogy a magasan képzett munkaerő által jelenleg végzett feladatok ellátását teljesen átveszi a mesterséges intelligencia. Sampson egy, a humánerőforrás számára kedvezőbb lehetőséget vázol fel, méghozzá hogy ezeket a "gondolkodást igénylő munkákat" a későbbiekben alacsonyabban fizetett munkaerő fogja ellátni, akiknek nem feltétlenül kell magasan képzetteknek lenniük, hiszen az információtechnológia egyre hatékonyabban fogja tudni támogatni a feladatvégzésüket (Sampson, 2018). Továbbá a bonyolultabb, nehezen algoritmizálható előrejelzéseket, vagy intuíciót, kreativitást, illetve stratégiai gondolkodást igénylő munkakörök esetében valószínűleg a közeljövőben sem kell számottevő automatizációval számolnunk. Kai-Fu Lee ezt egy az optimalizációra alapuló munkáktól a kreativitáson alapuló munkákig terjedő tengelyen szemlélteti. A legnagyobb veszélyben a repetitív vagy rutin feladatokat és optimalizációt végző munkavállalók vannak, míg a komplex munkákat ellátók, például vállalkozó, közgazdász stb., illetve kreativitásukból élő emberek, mint a művészek vagy tudósok helyzete biztosabb (Lee, 2018).

Nagyon valószínű, hogy bárhogy is lesz a jövőben, a 21. század emberének – ahogy Harari is jósolja – pár évente meg kell újítania magát, folyamatosan új tudásra és készségekre kell szert tennie az egyre gyorsuló technológiai fejlődés közepette, olyan ütemben, ahogy "új munkakörök alakulnak ki a digitális forradalomnak köszönhetően", és "az egyes munkakörök tartalma, természete és a szükséges készségek köre folyamatosan változik" (Grundke *et al.*, 2018, p. 6). Az élethosszig tartó tanulás (*lifelong learning*) lehet az emberek egyetlen esélye, hogy helyt tudjanak állni egy olyan munkaerőpiacon, ami a technológiai fejlődéssel párhuzamosan állandó változásban van.

Az „élethosszig tartó tanulás" kifejezés, nem csak azt jelenti, hogy az embereknek folyamatosan, minden életszakaszban képezniük kell magukat, hanem azt is, hogy ennek a képzésnek az élet összes területére ki kell terjednie (Laal, 2011). Általános

trend, hogy a gyerekeket egyre korábban kezdik el komolyan tanítani, például idegen nyelvekre, és az (ön)képzés gyakran még a nyugdíjazás után is folytatódik. Mindemellett ki kell terjedjen az élet minden területére, így nem csak az iskolában – formális oktatás során – tanulunk már új készségeket, de a munkahelyen – nem formális képzéseken – vagy éppen a közösségeinkben, illetve családjainkban – informális módon, kapcsolataink révén – is[2]. Koper és Tattersall (2004, p. 1) definíciója alapján az "élethosszig tartó tanulás fogalma azokra a tevékenységekre utal, amit az emberek az életük során végeznek, annak érdekében, hogy folyamatosan fejlesszék tudásukat, készségeiket és kompetenciáikat egy adott területen, valamilyen személyes, társadalmi vagy munkaerőpiaci motivációból kifolyólag".

Az élethosszig tartó tanulás irányába mutató trend – azaz hogy az embereknek folyamatosan meg kell újítaniuk magukat, nem csak az ipari-, de a szolgáltatói szektorban is – az 1960-as években kezdődött (Field, 2001), és az 1990-es években gyorsult fel igazán, amikor például a telefontársaságok operátor pozícióit automatizálták, a hangfelismerő technológiák fejlődésének köszönhetően. A közepesen képzett (*medium skilled*) munkaerőt érintette legrosszabbul a komputerizáció, mivel ők leginkább "rutin jellegű, ismétlődő lépésekből álló feladatokat végeznek gyártáshoz kapcsolódó, illetve irodai és adminisztratív támogató munkakörökben" (Acemoglu és Autor, 2010, p. 49) "amikre nagyobb veszélyt jelent a technológiavezérelt innováció" (Grundke *et al.*, 2018, p. 7). Fazekas megerősíti az előző állítást, hogy "jelentősen javult a felsőfokú szakképzettségűek helyzete és némiképp javult az alacsony szinten szakképzettek […] relatív munkaerőpiaci pozíciója", míg a közepesen képzettek helyzete az elmúlt évtizedekben jelentősen romlott (Fazekas, 2017, p. 7).

A gyorsuló technológiai fejlődés eredményeképpen a 2000-es évek közepére világossá vált tehát, hogy a 20. századra jellemző élethosszig tartó foglalkoztatás – hogy valaki ugyanazt a munkát végzi attól kezdve, hogy kilép a formális oktatásból, egészen a nyugdíjig, ugyanazokat a feladatokat ellátva, ugyanazokat a készségeket és tudást alkalmazva – aligha fenntartható a 21. században.

[2]Ezek a trendek mindig is jellemzőek voltak egyes szocio-ökonómiai rétegekben. Továbbá a tanulási folyamat sok esetben természetesen megy végbe. Azonban a felvázolt jelenségek hatására sokkal inkább előtérbe került, nagyobb hangsúlyt kapott az élethosszig tartó tanulás szükségessége az utóbbi időben. Az emberek sokkal tudatosabban, tervezettebben állnak hozzá a nem a formális oktatás keretei között történő tanuláshoz is.

Az előzőekben leírt változások következményei az informális- és az élethosszig tartó tanulásra kritikus fontosságúak, és a kormányoknak ki kell dolgoznia programokat, melyekkel ezek a célok támogathatók, például előmozdítják a munkájukat az automatizáció hatására elvesztett emberek átképzését. Jelen tézis ehhez a törekvéshez kíván hozzájárulni, azonban csak a felvázolt trendek formális oktatásra, pontosabban a felsőoktatásra gyakorolt hatásaival foglalkozik, illetve az ott jelentkező problémák megoldásához kíván hozzájárulni.

A rendkívül változékony környezetben zajló extrém gyors változások közepette, mikor a technológia exponenciális ütemben fejlődik – habár a Moore által előrevetített ütem mára lassulni látszik (Waldrop, 2016) – , nagyon nehéz a következő 10-20 évre, hosszú távra előrejelezni, és olyan javaslatokat adni a felsőoktatási intézményeknek, melyek alapján a munkaerőpiaci kereslethez alkalmazkodó tanterveket tudnak kidolgozni. Akik mégis megpróbálkoznak ilyen hosszú távra javaslatokat tenni, azt hangsúlyozzák, hogy az érzelmi intelligenciához és a szervezőkészséghez kapcsolódó tanulói kompetenciákat kell elsősorban fejleszteni (Beck és Libert, 2017), és egy olyan erős alapot kell biztosítani a hallgatóknak, amelyre könnyedén építhetnek, amikor az aktuális munkaerőpiaci keresletnek megfelelően új készségeket és tudást sajátítanak el. Munkaerőpiaci kutatások is megerősítik, hogy a nem rutin kognitív és szocio-emocionális kompetenciákra irányuló kereslet jelentősen növekedett az elmúlt évtizedekben, míg az ismétlődő fizikai, illetve rutin kognitív készségeket igénylő feladatok aránya csökkent. Mindez annak köszönhető, hogy míg a rutinfeladatok automatizációja egyre elterjedtebb, addig a “magas kooperációs készségeket […], személyes kapcsolatokat, érzelmi intelligenciát, “puha”, nem kognitív készségeket igénylő feladatokat” ez a trend még nem érinti (Autor *et al.*, 2003; Fazekas, 2017, p. 9).

Fazekas (2017) szerint három olyan terület van, elsősorban a szolgáltatási szektorban, ahol az automatizáció hatása jelenleg kevésbé érvényesül, és várhatóan a gépesítés nem is fogja tudni a közeljövőben kiváltani az emberi munkát. Mind a három területen elsősorban a puha készségek (*soft skill*) fontosak a munkavállalók hatékony feladatvégzése szempontjából. Ezek a szektorok az egyre gyorsabban növekvő urbanizáció miatt virágzó szolgáltatói szektor, az elöregedő népesség miatt egyre jobban igényelt beteg- és idősgondozás, illetve általánosan azok a munkahelyek, melyek magasan fejlett puha készségeket (például kooperativitás, érzelmi intelligencia

stb.) követelnek meg a munkavállalóktól, így nincsenek "kitéve az új technológiák munkaerő-kiszorító hatásának" (Fazekas, 2017, p. 13).

Jelen tézisben rövid távra igyekszem fókuszálni egy olyan keretrendszer felvázolásával, melynek segítségével a munkaerőpiaci trendek és a keresletet leíró adatok valós időben elemezhetőek. Egy munkaerőpiaci „adattárház" koncepciójának kidolgozására teszek kísérletet, melyben a munkaerőpiaci keresletre vonatkozó, álláshirdetésekben megjelenő információk összegyűjthetőek. Megvizsgálok továbbá olyan módszereket, melyek segítségével az állásajánlatokban explicit és implicit módon megjelenő, kompetenciakeresletet tükröző információk kinyerhetőek. Egy ilyen jellegű információforrás alapján – esetlegesen kiegészítve egyéb forrásokból származó adatokkal, piaci és iparági elemzésekkel, riportokkal, mint például a *Gartner Hype görbéje*[3] stb. – a felsőoktatás döntéshozói folyamatosan hozzá tudnák igazítani a kurzusok tematikáját a munkaerőpiaci kereslethez, és olyan kompetenciák elsajátítására tudnak lehetőséget kínálni a hallgatóknak, amit azok karrierjük első éveiben sikeresen értékesíteni tudnak a piacon. A jelenleg zajló, folyamatos diszkontinuitást okozó technológiai fejlődés közepette, mikor az *MI* és *GT* megbontó hatása folyamatosan átalakítja a munkaerőpiacot, ezek az információk jelentős versenyelőnyt jelenthetnek egy oktatási intézmény, illetve magasabb szinten akár egy nemzetgazdaság számára.

Tehát bár egyik oldalról az említett technológiák jelentős veszélyeket rejtenek magukban, másik oldalról ugyanezek a technológiák felhasználhatók arra is, hogy segítségükkel a döntéshozóknak hasznos információkat biztosítsunk. A mesterséges intelligencia és gépi tanulási algoritmusok segíthetnek a munkaerőpiaci trendek elemzésében, valamint valós idejű információk biztosításával támogathatják, hogy a tematikák és az akadémiai programok fejlesztésekor objektív, tényekre alapozott döntések szülessenek (Chang *et al.*, 2018).

[3] A Gartner Hype görbéje az egyes technológiák érettségét ábrázolja az elvárások és az idő függvényében. Amikor egy üzleti igény vagy egy innovációs eredmény elindítja egy új technológia fejlődését, az életciklus ilyenkor jellemzően négy fázisból áll. A túlzott elvárásokat általában egy kiábrándulási fázis követi, majd az adott termék elérkezik abba a fázisba, ahol már láthatóak azok a felhasználási esetek, melyek esetében a technológia hozzáadott értéket tud termelni. Ez a megvilágosodás szakasza. A kiábrándulási fázis persze eredményezheti a termék eltűnését is. A megvilágosodás szakaszát általában a technológia beépülése és termelékennyé válása követi.

1.2 Problémafelvetés

A bevezetésben részletezett trendeknek is köszönhetően, a munkaerőpiaci kereslet nagyon volatilis, rendkívüli ütemben változik. A kompetenciák és a tudás a kereslet szempontjából gyorsan elavulnak, egyre újabb és újabb ismeretekre van szükség. Az automatizáció is egyre inkább teret nyer mind több területen, a gépi tanulás és a mesterséges intelligencia alkalmazásához kötődő gépesítés és robotizáció sok, kognitív készségeket igénylő feladat esetében is kiváltja az emberi munkaerőt. Az egyén szintjén erre a problémára elméletben különböző megoldások létezhetnek, például az élethosszig tartó tanulás koncepciója és ehhez kapcsolódóan az emberek folyamatos megújulása és átképzése, vagy a feltétel nélküli alapjövedelem stb. Ugyanakkor nem elégséges csak az egyén szintjén kezelni ezt a kihívást, az egyes államoknak, illetve, amennyiben az oktatás területére koncentrálunk, úgy az oktatási rendszereknek, és azok intézményeinek fel kell készülniük, hogy támogassák ezeket a törekvéseket megfelelő infrastruktúrával, képzésekkel, tanácsadással, *coaching*gal és nem utolsósorban azzal, hogy eladható és – legalábbis rövid távon – időtálló kompetenciákat oktatnak. Azonban szemben a munkaerőpiac rapid keresletváltozásával, a kínálati[4] oldalon nagyon nehezen változó, nehezen alkalmazkodó intézmények igyekeznek ennek a keresletváltozásnak reaktív módon „megfelelni”.

A képzőintézmények többségére a merev, hierarchikus felépítés, a bürokrácia, a belső és külső politikai környezetnek való kitettség jellemző. A tanterveket általában kizárólag az oktatók saját, szükségszerűen korlátozott tapasztalata, esetlegesen szűkebb kutatási területe, elfogultságai stb. alapján állítják össze. Objektív információk hiányában a szak- és tantervek a legjobb szándék mellett sem tudják a piaci igényeket[5] megfelelő mértékben tükrözni. A képzési tervek akkreditálási eljáráson mennek át, ami hatalmas bürokratikus szervezetekben történik. Ez szintén megnehezíti az igényváltozásra való gyors reagálást.

[4] A munkaerőpiaci kínálat természetesen számos szocio-ökonómiai rétegből áll össze, de jelen dolgozatban, az egyes képzőintézmények által kibocsátott végzett hallgatókat értem bele ebbe a kínálatba, és az esetleges kivételes eseteket külön jelezem.

[5] Sok esetben természetesen nem (csak) ez a cél, például az alapozó tárgyak, vagy általánosságban az alapképzés esetében, amikor nem a szakmaspecifikus ismereteket, hanem az azok megalapozására szolgáló általános kompetenciákat kívánja az oktatás erősíteni. Emellett természetesen számos egyéb, például pedagógiai, társadalompolitikai stb. szempontot is figyelembe kell venni a tantervfejlesztés során.

Az előzőeken felül, a felsőoktatási intézmények esetében további nehézséget jelent, hogy tulajdonképpen nem is a jelenlegi piaci igényeket kellene figyelembe venniük, hanem a jövőbelieket, hiszen az a diák, akinek a képzési tervét egy adott évben összeállítják, csak évekkel később fog kilépni a munkaerőpiacra. Viszont a jövőbeli igényekre való felkészülés, pontosan a munkaerőpiaci igények rendkívül gyors változása miatt, rengeteg bizonytalansággal terhelt.

A kereslet és a kínálat munkaerőpiaci összeegyeztetésének további nehézsége, hogy a kereslet rendkívül heterogén. Ez a heterogenitás például jól látszik a földrajzi eltérésekben. Az egyes földrajzi régiókban különböző kompetenciákat, különböző tudást, képességeket keresnek. Ennek megfelelően a képzőintézményeknek alkalmazkodniuk kell az adott régió sajátos igényeihez, követelményeihez is.

További probléma az is, hogy a munkaerőpiaci igények és a kínálat sok esetben más absztrakciós szinten van megfogalmazva, ami szintén megnehezíti a dolgát annak, aki a megfeleltetésükön dolgozik. Míg a képzőintézmények képzési kimeneti követelményekben (*learning outcome*) gondolkodnak, mely gondolkodásmódban az átadott kompetenciák egy absztrakt szintet képviselnek, addig a munkaerőpiac sokkal praktikusabb megközelítés alapján, pragmatikusan tekint a kompetenciákra, és absztrakt kompetenciaosztályok helyett sokkal inkább azok gyakorlati, az adott feladat megoldásához illeszkedő instanciáit, azaz specifikus tudást keres. Így fontos kérdés, hogy hogyan, milyen módszerekkel lehet segíteni a kereslet és a kínálat megfeleltetését, közelítését. Például míg a képzési kimeneti követelmények között egy kompetencia úgy jelenik meg, hogy a hallgató „tudjon programozni”, vagy „legyen képes algoritmusokban gondolkodni”, addig a munkaerőpiac „Java fejlesztőt” keres, aki „tudja alkalmazni a Spring keretrendszert”. Sok esetben természetesen az absztrakt megfogalmazás, és ismeretátadás elégséges, hiszen arra építve, az alkalmazások[6] már könnyen elsajátíthatók. Azonban ez az eltérés az absztrakciós szintekben gyakran komoly kihívást jelent akkor, amikor össze akarjuk hasonlítani a munkaerőpiaci keresletet és kínálatot, illetve következtetéseket akarunk levonni azokról.

Az előzőekben ismertetett eset – melyben a fókusz a keresleti oldalról azon van, hogy a vállalat hogyan tudja megszerezni az igényeinek megfelelő emberi erőforrást, míg

[6]Az egyes elvont fogalmak konkrét implementációi. Például, ha valaki megtanulja az objektumorientált programozás alapjait, jellemzően valamely konkrét nyelv megismerésén keresztül, úgy más implementációk, más OO nyelvek esetében „csak” egy eltérő szintaxist kell elsajátítania.

kínálati oldalon csak a tanulmányaikat éppen befejezők, a munkaerőpiacra most kilépők jelennek meg – azonban a teljes problématérnek csak egy kisebb részét mutatja be. Ugyanebbe a problémaosztályba tartozik egy másik, a vállalati belső munkaerő-allokáció esete. A belső munkaerő-allokáció azt jelenti, hogy egy vállalatban egy nyitott pozícióra vagy egy projektcsapatba a már meglévő humán tőkéből szeretnénk megtalálni a legmegfelelőbb jelöltet. Itt a kínálati oldalon a vállalati belső képzési rendszerek és programok jelennek meg. Az a kérdés, hogy mit kell tudni valakinek egy adott pozíció betöltéséhez, ebben az esetben például úgy fogalmazódhat át, hogy melyik munkavállalónak milyen képzésre van szüksége.

A keresletnek megfelelő kínálat biztosítása versenyképességi kérdés is, mely mind az egyes munkavállalók, családjaik és a vállalatok szintjén, mikro szinten, mind aggregáltan makro szinten kifejti hatását. Azonban a képzőintézményi kínálat és a munkaerőpiaci kereslet közelítése, fontosságán túl rendkívül bonyolult feladat is, számos problémával és kihívással, amik a munkaerőpiaci kereslet egyes attribútumaiból, illetve a kínálat sajátosságaiból adódnak. Egy olyan rendszer, amely objektív és validálható inputokat nyújt az oktatási döntésekhez, annak érdekében, hogy a tantervek minél inkább tükrözzék a munkaerőpiaci valóságot – azaz minél több, a keresleti igények alapján kijelölt kompetenciát erősítsenek a kurzusok – jelentős hozzáadott értéket tud a képzéstervezéshez adni.

Általánosítva tehát a tágabb kutatásom ahhoz a problémához kapcsolódik, hogy milyen módszerekkel, hogyan lehet elemezni és előrejelezni a munkaerőpiaci keresletet, vizsgálni a kínálat kereslethez való illeszkedését. Továbbá összehangolni azokat, elérni, hogy a képzési kimeneti követelmények egyre jobban tükrözzék az álláshirdetésekben megjelenő kompetenciaigényeket. Ahogy az előzőekben részleteztem, kínálat alatt itt nem az egyének (munkavállalók) összességét értem, hanem valamiféle képzés által nyújtott kimeneti minőséget, míg a keresleti oldalon kompetencia- (mint erőforrás) szükségletről beszélek. Mivel a teljes kutatás kidolgozása túlnyúlna egy doktori értekezés keretein, ezért az értekezésben a tágabb kutatást megalapozó keretrendszer alkalmazhatóságát mutatom be. Azaz a jelen tézis és a kutatási kérdések a teljes problématérnek csak egy részletére kívánnak megoldásokat találni, a munkaerőpiac keresleti oldalára fókuszálva, arra, hogy miként lehet a kompetenciakeresletet jelző információkat hatékonyan feltárni és tárolni, ezzel

megalapozva olyan elemzéseket, melyek később a kínálat jobb illesztését lehetővé teszik.

A kutatás célja tehát egy olyan munkaerőpiaci adattárház koncepciójának, illetve egyes komponenseinek a kidolgozása, amely segítségével az oktatási szektorban dolgozó döntéshozók, pl. szakfelelősök, tárgyfelelősök elemezni tudják az aktuális munkaerőpiaci kompetenciaigényeket, valamint mintázatokat, trendeket is azonosíthatnak benne a későbbi fejlesztések eredményeként. Mindezen ismereteket fel tudják használni a képzés vagy a tananyagok fejlesztése során. A kutatás kontextusba helyezéséhez a következő részben alapfogalmak és kapcsolódó kutatások bemutatására kerül sor.

1.3 Alapfogalmak

Mielőtt a dolgozat célját és kereteit részletesen bemutatnám, röviden ki kell térnem azoknak a fogalmaknak a definiálására és értelmezésére, melyeket a dolgozat során végig használni fogok. A következő alfejezetben szakirodalmi források alapján ismertetem a készség, képesség és kompetencia egyes definícióit, továbbá kitérek arra, hogy jelen dolgozatban hogyan fogom ezeket a fogalmakat értelmezni és használni. Kitérek továbbá a képzési kimeneti követelmények témakörére, bemutatom, hogy a kompetenciák hogyan, illetve milyen szinten jelennek meg a képzési kimeneti követelményekben. Ezzel azt próbálom megvilágítani, hogy milyen információkat várunk egy olyan információrendszertől, mely a bevezetőben és a problémafelvetésben megfogalmazott módon arra törekszik, hogy a tárgy- és szakfelelősöket támogassa a keresletvezérelt tananyagfejlesztésben.

1.3.1 Készség, képesség, kompetencia

Grundke és szerzőtársai (2017) megkülönböztetnek *kognitív* (*cognitive*) és *nem kognitív* (*non-cognitive*) készségeket (*skill*). A kognitív készségek értelmezésükben azok, melyeket el lehet sajátítani, meg lehet tanulni, jellemzően valamiféle oktatás során, míg a nem kognitív készségek általában olyanok, mint a személyiségvonások, vagy veleszületettek, vagy a szocializáció során sajátítjuk el őket, mintegy tudattalanul. Kognitív készségek többek között a számolás, az írás, egyes problémamegoldáshoz kapcsolódó készségek, illetve a jelen dolgozat szempontjából fontos IKT (*információs és kommunikációs technológiák*hoz kapcsolódó) készségek. Ebből a szempontból a tanulmány nem teljesen következetes, mert az IKT készségeket

olykor a feladatspecifikus készségek közé sorolja, melyet egyértelműen megkülönböztet a kognitív készségektől. Feladatspecifikus készségek ebben az értelmezésben azok, melyek egy adott feladat elvégzéséhez szükségesek (Grundke *et al.*, 2018).

A nem kognitív készségek közé többek között az irányításhoz, vezetéshez kapcsolódó, a kommunikációs és a szervezőkészséget sorolják a tanulmány szerzői, és úgy határozzák azt meg, mint amit a „vizsgált dolgozó által a munkavégzés során ellátott feladatokból nyert információk felhasználásával lehet mérni" (Grundke *et al.*, 2018, p. 5). A „nem kognitív készség" kifejezés ebben a megfogalmazásban nem teljesen pontos, hiszen az előbb felsorolt viselkedéshez szorosan köthető készségek nagyon is igénylik a kognitív folyamatok közreműködését, mivel az érzelmek befolyásolják a viselkedést, mely érzelmek irányításában pedig részt vesznek a magas szintű kognitív folyamatok (Scorza *et al.*, 2016).

Az előzőekkel ellentétben, pedagógiai értelemben az olvasást, írást, számolást az *alapkészségek* közé, míg a kognitív *képességek* közé például a figyelmet, emlékezést, gondolkodást stb. sorolják. A kognitív képességek ebben az értelmezésben tehát primerebbek, mint az alapkészségek, azaz ezek teszik lehetővé az alapkészségek iskolai elsajátítását.

Egy korábbi tanulmányukban a szerzők szintén megkülönböztetnek szocio-emocionális készségeket, amilyen például a "tanulásra való hajlandóság" és a "kreatív problémamegoldás", és úgy találták az elemzett PIAAC[7] felmérések adatai alapján, hogy e készségek szintje valóban korrelál a válaszadók szociális hátterével (Grundke *et al.*, 2017).

Egyes szerzők megkülönböztetnek szakmához kötődő (*vocational*) készségeket is, melyekre munkatapasztalat, vagy speciális szakoktatás során tehetünk szert. Ezek jellemzően nem kapcsolódnak végzettséghez, nem részei a képzési nómenklatúráknak, mint például az ISCED (*International Standard Classification of Education*), mivel feladatspecifikusak (Wowczko, 2015).

[7]A „Nemzetközi Felnőtt Képesség- és Kompetenciamérési Program" (*Programme for the International Assessment of Adult Competencies*) az OECD kutatási programja, melynek során, a résztvevő országokban felnőttek készségeit, problémamegoldó képességét stb. vizsgálják (Hanushek *et al.*, 2015).

Egy másik megközelítésben, az irodalomban megkülönböztetnek “kemény” (*hard*) és “puha” (*soft*) avagy karakter (*character*) készségeket. Kemény készségek azok, melyek egy feladat ellátásához feltétlenül szükségesek, amilyenek például az alapkészségek és az azokat kiegészítő szakmai tudás, míg a puha készségek azok az előzőeken felüli készségek, melyeket Grundke és munkatársai a nem kognitív, illetve a szocio-emocionális készségek közé sorolnak mint például a problémamegoldás, kreativitás, motiváció, szervezési és beilleszkedési készségek stb., ezek azok a karakterhez kapcsolódó készségek melyeket nem lehet IQ és egyéb felmérő tesztekkel mérni (Heckman és Kautz, 2013). A puha készségeket hivatkozzák még „élet- és 21. századi készségek” neveken is (Scorza *et al.*, 2016). A kemény készségek értelmezése nagyjából átfedésben van a kognitív és feladat specifikus készségek „metszetével”.

Az elmúlt években jelentősen megnövekedett a puha készségek iránti érdeklődés és kereslet, nagyjából összhangban azzal, ahogy a bevezetőben is említett módon a rutin kognitív készségeket igénylő feladatok automatizálhatósága egyre inkább lehetővé vált. A puha készségek meghatározásával és megkülönböztetésével a pszichológia területe foglalkozik elsősorban, azonban a megfogalmazások és definíciók sokszor átfedőek, a koncepciók nem teljesen tiszták, az egyes szerzők között nincsen megegyezés, az egyes készségek definiálása és értékelése nehézkes (Scorza *et al.*, 2016). Az emberierőforrás-menedzsment irodalomban elsősorban azzal foglalkoznak, hogy hogyan lehet ezeket a nem teljesen konkrétan körülírható készségeket mérni, illetve hogyan lehet a cégeket kiválasztási folyamataikban támogató eszközöket, például személyiségteszteket kidolgozni. A neurológia tudománya a puha készségekhez kapcsolódó agyi funkciókat kutatja, míg a közgazdasági kutatások érdeklődése elsősorban a puha készségekbe befektetett tőke megtérülésére, annak hozamaira terjed ki (Fazekas, 2017).

Ellentétben a jobban megfogható kemény készségekkel, melyeket IQ és egyéb felmérő tesztekkel viszonylag egzakt módon mérni lehet, illetve akár szint (alap, szakterülethez kapcsolódó) vagy terület alapján rendszerezni, a puha készségek definíciójában, csoportosíthatóságában, rendszerezésében és mérésében nincs egyetértés. Azzal azonban minden szerző egyetérteni látszik, hogy ezekre a készségekre a 21. századi munkavállalóknak egyre nagyobb szükségük lesz. Ez a helyzet az IKT területen is, a sikerességhez nem elégséges csak a technikai, technológiai készségek megléte, a puha

készségek értéke ezen a területen is egyre növekszik (Ahmed *et al.*, 2012; Wowczko, 2015).

A kompetencia definíciójában sincs teljes egyetértés az irodalomban. Scorza és szerzőtársai (2016, p. 1) például úgy hivatkoznak a kompetencia fogalmára, mintha az a készség fogalmának egy időszerűbb, modernebb, találóbb változata lenne. Adam úgy fogalmaz, hogy többen nagyon korlátozottan látják a kompetenciákat, és csak olyan készségekkel társítják őket, melyeket képzések útján lehet elsajátítani (Adam, 2004; Kennedy *et al.*, 2007, p. 6). Kennedy is azt hangsúlyozza, hogy nincs egyetértés a kompetencia definíciójában, és így nem is teljesen világos, hogy mit is értünk alatta.

A legtöbb szerző azonban egyfajta ernyőfogalomként értelmezni a kompetenciát, mely magában foglal készséget, képességet, tudást, attitűdöt és motivációt, illetve azoknak a dinamikus kombinációját, amelyre egy egyénnek egy adott feladat elvégzése, vagy probléma hatékony kezelése érdekében van szüksége (Hecklau *et al.*, 2016; Kennedy *et al.*, 2007). Falus (2010, p. 6) értelmezésében a „kompetencia átfogó fogalom […], pszichikus képződményeknek (tudás, attitűdök, képességek) egy olyan rendszere, amely lehetővé teszi valaki számára, hogy egy adott területen eredményesen tevékenykedjen". Hecklau és munkatársai (2016) kifejezetten a munkával kapcsolatban felmerülő feladatok és kihívások megoldásának képességéhez kötik a kompetenciákat, azaz hogy egy munkavállaló képes és hajlandó végrehajtani egy adott feladatot, rendelkezik a szükséges kemény készségekkel és tudással, a megfelelő hozzáállással; hogy például az ügyfelek elvárásait vagy éppen a stresszt adekvát módon tudja kezelni, megfelelően tudja az eredményeket kommunikálni stb.

Ilyen értelemben a kompetencia egy metafogalom, melynek instanciái lehetnek. Oktatási szempontból egy-egy kompetenciának a megalapozása, kiépítése akár több kurzuson átívelő feladat is lehet, így a kompetencia fogalma átfed a tanulási eredmények fogalmával, amennyiben egy-egy kompetenciát, mint egymásra épülő tudás- és készségelemet fogunk fel. Ebből a megközelítésből egy kompetenciára példa lehet az objektumorientált programozás, míg kapcsolódó instanciára az egységbezárás (*encapsulation*) tudáseleme, de egy rekurzív függvény megírására való képesség is. Derényi és Vámos (2015, p. 14) ettől némileg eltérően úgy fogalmaz, hogy a tanulási eredmények megfogalmazásában egyes kompetenciák fejlesztésének célja jelenik meg.

A kompetenciákat is több szempont szerint szokták osztályozni. Az egyik megközelítés alapján elkülöníthetünk szakterület-specifikus és általános kompetenciákat (Kennedy *et al.*, 2007). A kompetenciafogalom ezen megkülönböztetése megjelenik a tanulási eredmények megfogalmazásával kapcsolatban is, ugyanis a végzett hallgatóknak nem csak a jól strukturált, szakmaspecifikus problémákkal kell tudniuk megbirkózni, de kezelniük kell tudni a félig vagy rosszul strukturált, előre nem látható, kreativitást, elvonatkoztatást, illetve a megoldáshoz különböző, akár egymáshoz szorosan nem kapcsolódó kompetenciák ötvözését, kombinálását igénylő feladatokat is (Derényi és Vámos, 2015).

Hecklau és szerzőtársai (2016) megkülönböztetnek technikai, módszertani, szociális és személyes kompetenciákat. A technikai vagy szakmai kompetenciák közé sorolják a munkához, feladatvégzéshez szükséges tudást és készségeket, míg a módszertani kompetenciák az általános problémamegoldásban játszanak szerepet. Ez a felosztás eddig megfelel az előzőekben bemutatott osztályozásnak, ahol a kompetenciákat szakterület-specifikus és általános csoportokra osztják. Azonban Hecklauék megkülönböztetnek továbbá olyan kompetenciahalmazokat is, melyek közelebb álnak a puha készségek definíciójához, és az előző duális felosztást tekintve talán inkább az általános osztály további bontásának tekinthetők. A szociális vagy közösségi kompetenciák közé a kommunikációt, az együttműködés különböző formáit, illetve a vezetői készséget sorolják, míg a személyes kompetenciák között többek között a motiváció, a stressztűrőképesség vagy éppen a rugalmasság jelennek meg (Hecklau *et al.*, 2016).

Jelen tézisben a kompetencia kifejezést, mint ernyőfogalmat használom, mely alá készség, képesség, tudás, attitűd és autonómia elemek is tartoznak, és elsősorban azokat a kompetenciákat vizsgálom, melyek leginkább mint kemény-, vagy technológiai készségek kerültek megfogalmazásra. Ezek azok a jól mérhető, könnyen körülhatárolható, és szilárd határokkal rendelkező, azaz jól megfogalmazott készségek és tudáselemek, melyek beazonosítása szövegbányászati eszközökkel, jó hatásfokkal lehetséges, mivel megfogalmazásukban viszonylagos egyetértés van. Továbbá ezek általában azok közé a szakmaspecifikus készségek közé tartoznak, melyeket egy szakképzés, jelen esetben például egy egyetemi kurzus célja lehet a hallgatóknak átadni. Ezek az ismeretek magas szinten a képzési kimeneti követelményekben szoktak manifesztálódni.

1.3.2 Képzési kimeneti követelmények

Annak eldöntése, hogy mit tanítsanak a hallgatóknak a magyar egyetemeken, egy olyan iteratív folyamat eredménye, amely számtalan egyetemi hierarchiaszintet érint. Mindemellett igényel kormányzati részvételt is, ami instrukciókon, iránymutatásokon (rendeletek), akkreditáción és finanszírozáson keresztül valósul meg. Ezeknek a döntési szinteknek megfelelően a képzési kimeneti követelmények „rendszere" maga is hierarchikus (1. ábra), magas szinten általánosításokat fogalmaz meg, míg a szak- illetve tárgyleírásokban ezek az általánosítások kerülnek egyre inkább kidolgozásra, egyre specifikusabb formában.

A hierarchia legmagasabb szintjein az Európai Unió, illetve a tagországok nemzeti szintű képzési kimeneti követelményei (*KKK*, *Education and Outcome Requirements*) találhatóak, melyek általános szinten, absztrakt formában vannak megfogalmazva. 1999-ben 29 európai ország aláírta a Bolognai Nyilatkozatot, és létrehozta az Európai Felsőoktatási Térséget (EHEA, *European Higher Education Area*), többek között azzal a céllal, hogy megkönnyítsék a fiatal végzettek mobilitását. Célul tűzték ki, hogy a diákok a térség országaiban összehasonlítható tartalmú és minőségű kompetenciákat kapjanak az egyes képzési szinteken, például az egyetemen. Ahhoz azonban, hogy az egyes intézmények által kibocsátott diplomák megfeleltethetőek legyenek egymásnak, az egyes képzéseket kimeneti fókusszal kellett megfogalmazni. Ez az elvárás praktikusan azt jelenti, hogy a munkáltatók innentől kezdve biztosak lehetnek benne, hogy adott végzettségű fiatalok ugyanazokkal a kompetenciákkal rendelkeznek, függetlenül attól, hogy Brüsszelben, vagy Budapesten jártak-e egyetemre. Ennek megfelelően a nemzeti oktatási keretrendszereket és az egyes akadémiai programokat az Európai Felsőoktatási Térség országaiban 2010-ig képzési kimeneti követelményekben rögzítve, „tanulási eredmény" (Derényi *et al.*, 2015, p. 105) alapon kellett átdolgozni. A tanterveket, kurzusleírásokat is ennek megfelelően kellett újratervezni, hogy azt tükrözzék, hogy a hallgatók milyen kompetenciákkal rendelkeznek a képzés végére, ahelyett, hogy pusztán a megszerzett kreditek vagy a képzésen töltött idő alapján tudnánk következtetni a diákok képességeire (Kennedy *et al.*, 2007).

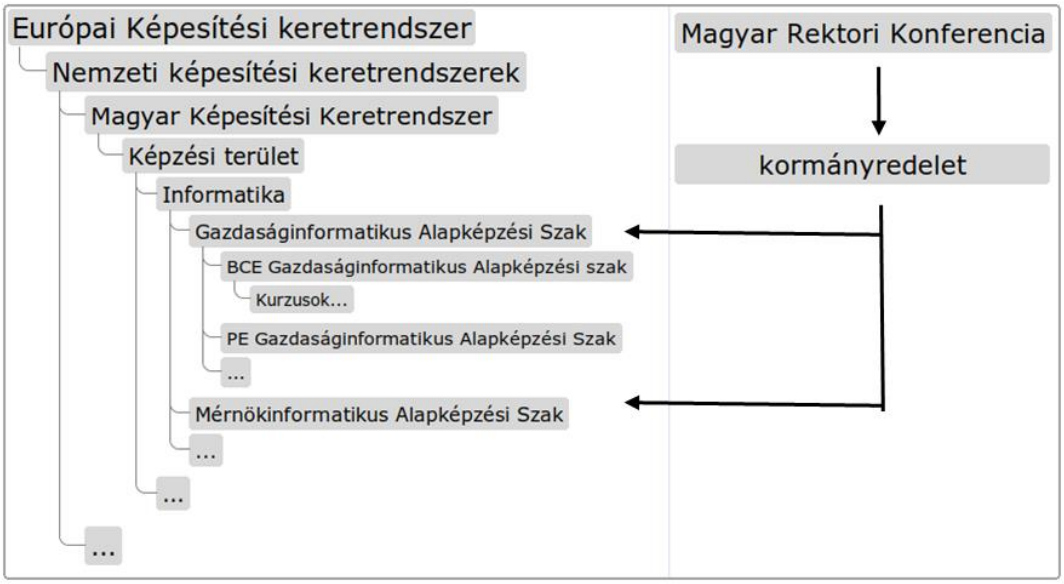
Kennedy és munkatársai (2007, p. 4) a tanulási eredmény (*learning outcome*) fogalom számos definícióját gyűjtötték össze, azonban ezek rendkívül hasonlóak, és megegyeznek abban, hogy a kimeneti követelmények a tanuló szempontjából vannak

megfogalmazva, azaz azt mondják meg, hogy a hallgató milyen tudással fog rendelkezni, illetve mit fog tudni elvégezni a képzés végére. A klasszikus bemenetorientált felfogással ellentétben, mely azt nézi, hogy mi volt a tanár célja, milyen anyagot adott le, milyen résztémákat érintett az előadások során (tanárközpontú megközelítés), a hallgatóközpontú, kimenetorientált megközelítés arra koncentrál, hogy a diák mit ért el, mire képes az adott akadémiai program, vagy annak egy modulja végére.

Adam (Adam, 2004, p. 8) alapján „a tanulási eredmények gyakran, mint kompetenciák vannak megfogalmazva", azonban ezt a véleményt Kennedy (2007, p. 7) nem osztja, mivel a kompetencia fogalom definíciójában nincs egyetértés, és nem eléggé világos, hogy mit is értünk kompetencia (*competence, competency*) alatt. Az azonban biztos, hogy a kimeneti követelmények általában absztrakt formában, felső szinten vannak megfogalmazva. Például az információtechnológia területén egy ilyen megfogalmazás lehet, hogy „megérti a hallgató az objektumorientált paradigmát". Az így megfogalmazott követelmények mérése azonban nehéz, általában csak következtetni tudunk arra – a teszteredményeken keresztül – hogy mi állt össze a hallgatók fejében. Egyszerűbben mérni az ismeretek, képességek tárgyiasult formájú megjelenését, a kompetenciák implementációját (*learning output*) lehet, amilyen például egy programozási feladat, vagy egy tesztkérdés. A hallgató vagy jól meg tudja válaszolni az adott kérdést, illetve helyesen tud implementálni egy adott bemenetből meghatározott kimenetet előállító kódsort, vagy nem.

Jó esetben a konkrét kompetenciainstanciák és a kimeneti követelmények között logikai kapcsolat áll fenn, azaz a tantárgyak keretében oktatott specifikus tudáselemeket a magas szintű irányok alapján határozzák meg, így a képzés végére egy hallgató teszteredményei alapján viszonylagos biztonsággal kijelenthető, hogy megfelel-e az adott program kimeneti követelményeinek vagy sem. Az, hogy az egyes tárgyak és adott intézmény szakleírása között hogyan építik fel a kimeneti követelmények és a konkrét kompetenciainstanciák közötti kapcsolatot, az részint az egyes egyetemek autonómiája, másrészt azonban a képzéseket – minőségbiztosítási okokból – akkreditáltatni is szükséges, mely folyamat során a Magyar Akkreditációs Bizottság értékeli, hogy az adott képzés megfelel-e a nemzeti sztenderdeknek. A Magyar Képesítési Keretrendszer (*MKKR*, *HuQF*) teremti meg az intézményi programok – például a Gazdaságinformatikus szak a Pécsi Tudományegyetemen,

illetve a Budapesti Corvinus Egyetemen – közötti, kimeneti követelmény alapú kapcsolatot. Az egyes tagországok képesítési rendszereit európai szinten az Európai Képesítési Keretrendszer (*EKKR, EQF*) hivatott összekapcsolni (Derényi *et al.*, 2015).



1. ábra: A képzési kimeneti követelmények lebontásának rendszere

Magyarországon, nemzeti szinten a Magyar Rektori Konferencia határozza meg, hogy egy szaknak (például Gazdaságinformatikus alapképzés) milyen kompetenciákat kell átadni a hallgatóknak, amit aztán rendeletben is rögzítenek. Például egy magas szintű tanulási eredmény (kimeneti követelmény), hogy a képzés végére a hallgató "rendelkezik az információrendszerekkel kapcsolatos alapvető ismeretekkel, érti az architektúra szervezési elveket, és összefüggéseiben képes értelmezni a számítástechnikai és információ architektúra összetevőit" ahogy az a 18/2016. (VIII. 5.) Emberi Erőforrások Minisztériuma (EMMI) rendeletben megfogalmazásra került (Wolters Kluwer Kft., 2016).

Intézményi szinten, a szakleírásokban még mindig magas absztrakciós szinten, bár már valamelyest az adott egyetem céljainak, missziójának megfelelően konkretizálva vannak megfogalmazva ezek a követelmények. Ezt befolyásolhatja például a régió, ahol az intézmény működik, vagy éppen a vállalati és egyéb szponzorkapcsolatok stb. Például a Budapesti Corvinus Egyetem Gazdaságinformatikus képzése esetében egy – az előzőleg idézetthez kapcsolódó – kimeneti követelmény az, hogy a hallgató "ismeri az információ architektúra különböző rétegeinek (tranzakció-feldolgozás, operatív működés támogatása, döntéstámogatás, csoportmunka, munkafolyamat) alapvető jellemzőit és a közöttük levő összefüggéseket" (Budapesti Corvinus Egyetem, 2018).

Habár általában még mindig egy magasabb, absztrakt szinten van megírva, de a szakleírás már részletezi a kompetenciákat és tartalmaz konkrétabb – tudás, képesség, attitűd, autonómia és felelősség formában megfogalmazott – elemeket is. A szakleírás meghatározza továbbá a törzsanyagot, ami alapján aztán az egyes kurzusok során oktatott ismeretek levezetésre kerülnek.

A következő szint a szakleírások alatt az egyes kurzusleírások szintje, ahol már egészen konkrétan a tudásterületek szerint is meg van fogalmazva a kurzus célja, azaz hogy mit kíván átadni a hallgatóknak. Természetesen elméletileg ezek a tudáselemek, a fenti absztrakt kimeneti követelményekből vannak levezetve, és amennyiben összegezzük a konkrét elemeket, úgy azoknak megfelelő lefedését kell adniuk az egyes kimeneti elvárásoknak. Például a Gazdaságinformatikus szak Szoftver Engineering kurzusa többek között lefedi a következő témaköröket "objektumorientált tervezés", "integrációs tesztelés", vagy ezeknél még specifikusabban például "állapotdiagramm" vagy "öröklődés" etc. (Budapesti Corvinus Egyetem, n.d.).

A képzési kimeneti követelmények rendszerének egyik problémája, ha azt a munkaerőpiaci kereslet szempontjából vizsgáljuk, az absztrakciós szintekben, illetve az absztrakt kompetenciákat megalapozó konkrét tudáselemek felcserélhetőségében rejlik. Azaz abban, hogy bár az absztraktan megfogalmazott képzési kimeneti követelmények, elsősorban általánosságuknak köszönhetően, könnyen kiállják az idő próbáját, és hosszú időn keresztül érvényesek tudnak maradni, addig tényleges tartalmuknak változnia kell, annak megfelelően, ahogy a piaci kereslet változik. Tehát lehet, hogy egy adott időpillanatban A tudáselem- és képességhalmazzal lefedett képzési kimeneti követelményt, egy későbbi időpillanatban, munkaerőpiaci szempontból már egy másik, B halmaz fedi le. Az előző példa szempontjából nevezhetjük a képzési kimeneti követelményeket absztrakt kompetenciáknak, míg az azokat megtestesítő tartalmat, az oktatott konkrét tudáselem- és képességhalmazokat kompetenciainstanciák halmazának.

A tartalomváltozást, melyet az egyes munkakörökben végbemenő feladattartalom-változás indukál, a felsőoktatási kínálatnak ideális esetben követnie kellene. Például a kilencvenes évek végétől nagy paradigmaváltások történtek a szoftverfejlesztés területén. Historikusan ahhoz, hogy valakiről elmondható legyen, hogy "képes elemezni, modellezni és implementálni üzleti követelményeket", szükség volt arra, hogy az illető ismerje és értse a vízesés modellt, illetve tudja azt alkalmazni. Manapság

viszont az IT szektor egyre inkább az agilis metodológiák irányába mozdul el, míg a vízesés modellhez kapcsolódó kompetenciaelemek implementációs szempontból egyre kevésbé fontosak. A példa természetesen sarkított, de ez a jelenség tapasztalható számos egyéb koncepció és technológia esetében.

Tehát míg a képzési kimeneti követelmények általános megfogalmazásukból kifolyólag viszonylag stabilak és időtállók, addig az azokat megalapozó konkrét tudáselemekről, kompetencia instanciákról, azaz a tantárgyi implementációról ez már nem minden esetben mondható el. Így a szak- és tárgyfelelősöknek gyakran kellene szinkronizálniuk a tematikákat, hogy az kielégítse a munkaerőpiac változó kompetenciaigényét. Jelen tézis alapfeltevése, hogy az érintett szak- és tárgyfelelősöknek az állásajánlatok – azaz a munkaerőpiaci kompetencia-keresletet adott időpillanatban legjobban reprezentáló objektumok[8] – elemzésével, érvényes és hasznos előrejelzések adhatók az egyes tantárgyak, vagy azok összefüggő rendszerének kompetenciatartalmára, illetve annak változására vonatkozóan. Mindez elősegíti a tervezést, így időszerű és pontos adatok birtokában rugalmasabban reagálhatnak a munkaerőpiaci kompetenciaigények mind gyorsabb változására.

A munkaerőpiaci kereslet térbeli és időbeli elemzésével a felvázolt kimeneti követelményrendszer absztraktabb szintjeit tervezők számára is hasznos, objektív információk adhatók. A képzési kimeneti követelmények kialakítási folyamatának jelenlegi működését nagymértékű szubjektivitás jellemzi, hiszen az egyes szakok, képzési programok kimeneti követelményei szakmai egyeztető értekezletek során alakulnak ki, és a jelenlévők korlátozott információk birtokában kialakított jövőbeli elvárásait, piacról alkotott véleményét tükrözik. Egy informatikai rendszer, mely ellenőrizhető, valós piaci adatokra alapozott, objektív inputokkal szolgálhatna ehhez a folyamathoz, nagyban segíthetné a tervezést.

Egy ilyen rendszernek, melynek célja tehát a felsőoktatási kínálat és a munkaerőpiaci kereslet összehangolása, a kompetenciák megfogalmazásának minden absztrakciós

[8]Jelen tézisben azzal a feltételezéssel élek, hogy számomra leginkább elérhető módon az állásajánlatok reprezentálják a munkaerőpiaci igényeket (2. fejezet bevezetőjében szakirodalmi forrásokat is idézek melyek alátámasztják ezt a feltételezést). Természetesen az állásajánlatokban nem jelenik meg számos – a kereslet szempontjából releváns – aspektus, például azon igények, melyeket a vállalatok belső képzésekkel, erőforrás-átcsoportosítással, munkaerő-átképzéssel vagy -kölcsönzéssel kívánnak kielégíteni. Így a fenti állítás, mely szerint az állásajánlatok reprezentálják legjobban adott időpillanatban a munkaerőpiaci kompetenciakeresletet, csak megszorításokkal fogadható el.

szintjét figyelembe kell vennie, de az összehasonlítás legfontosabb szintje az lesz, ahol a konkrét instanciák megjelennek, tehát oktatási (kínálati) oldalon az egyes kurzusleírások szintje. Ez az az absztrakciós szint, amin jellemzően a kompetenciaigények meg vannak fogalmazva az álláshirdetésekben is, mivel a munkáltatók konkrét pozícióra keresnek erőforrást, például Java programozót, illetve az egyes feladatok tartalmában is ezen a szinten leírt kompetenciákat találunk a vállalati folyamatmodellekben. Egy ilyen, az álláshirdetések kompetenciatartalmát kinyerni kívánó rendszer megvalósíthatóságának szempontjából fontos továbbá, hogy az egyes – a dolgozatban kompetencia-szótárként hivatkozott – forrásokban (például ESCO[9] stb.) is jellemzően ezen a szinten vannak a kompetenciaelemek megfogalmazva.

[9]Az ESCO (European Skill, Competences, Qualifications and Occupations) praktikus szempontból az európai munkaerőpiaci- és oktatási rendszer reprezentációja ontológia formájában. Az ESCO részletes ismertetése külön alfejezetben található a dolgozatban.

2 A kutatás tárgya és kerete

Jelen dolgozatban egy olyan keretrendszer részben elméleti megalapozására, részben megvalósíthatóságának – a kompetenciákat kinyerni hivatott modellek kidolgozásával és tesztelésével való – bizonyítására törekszem, ami implementációja után a felsőoktatás döntéshozói – elsősorban az oktatók, illetve a szakfelelősök – számára nyújthat majd segítséget olyan tantervek létrehozásában, melyek által a hallgatók számára kínált kompetenciák még akkor is érvényesek és eladhatóak lesznek, amikor az, aki ma kezdi egyetemi karrierjét, kilép a munkaerőpiacra. Ez elsősorban azt igényli, hogy a tantervek olyan elemekből álljanak, melyek biztosítják a naprakész és a munkaerőpiaci kereslet alapján megfogalmazott kompetenciák átadását a hallgatók számára.

A tanterveknek tehát folyamatosan összhangban kellene lennie a munkaerőpiaci kereslettel, amit számos tényező befolyásol, így előrejelzése egyáltalán nem triviális. Az érintettek (a tárgy- és szakfelelősök) legnagyobb nehézsége ebből a szempontból az, hogy nem látják előre az igények változását. Ha objektív képet kaphatnának arról, hogy a kompetenciák iránti kereslet hogyan alakul a munkaerőpiacon az időben, akkor következtetéseket tudnának levonni a jövőbeli trendekre vonatkozóan. A legfontosabb információ, amire egy ilyen előrejelzéshez szükség van, az a munkaerőpiaci kereslet reprezentációja az egyes pillanatokban, idősorosan rögzítve. Ezt tulajdonképpen „pillanatképek” összességének is felfoghatjuk az egyes időpillanatokban igényelt kompetenciahalmazokról.

Ahhoz azonban, hogy megfelelően árnyalt és informatív képet kapjunk, az igényeket több dimenzió, például földrajzi, gazdasági stb. mentén is ábrázolnunk kell. A gazdasági ciklusok és a jelentősebb – például Brexit súlyú – események is befolyásolhatják a munkaerő iránti igényeket.

Tehát annak megfelelően, ahogy a kompetenciák iránti igényt is számtalan faktor alakítja, a tárgy- és szakfelelősök információszükséglete is nagyon széles körű lehet. Az elsődlegesen fontos idődimenzió mellett – azaz, hogy az időben hogyan alakul az egyes kompetenciák iránti kereslet – a javasolt rendszer hosszú távon felkészíthető a döntéshozók információigényének más dimenziók mentén, például regionális vagy iparági stb. kielégítésére is.

Földrajzi szempontból például fontos lehet, hogy egy adott régióban milyen a munkaerőpiaci kereslet összetétele, dominálja-e azt pár nagyobb vállalat, melyeknek jól meghatározható, speciális kompetenciaigényei vannak, vagy sok kis cég versenyzik, és a kereslet nagymértékben heterogén. Ha tudja egy egyetemi döntéshozó, hogy az adott régióban versenyző vállalatok milyen kompetenciaigényekkel lépnek fel a piacon, úgy meg tudja ítélni, hogy a képzési kibocsátás mennyiben fedi le azt.

Iparági szinten az egyes ágazatok közötti összehasonlításon kívül fontosak lehetnek az egyes területeket jellemző mutatók, mint például a kibocsátás és a kompetenciakereslet közötti kapcsolat is. Annak felderítése, hogy a különböző gazdasági események, és ezek kapcsán a fontosabb makroökonómiai mérőszámok változása milyen kapcsolatban van a kompetenciakereslet alakulásával, szintén hasznos információkkal szolgálhat az előrejelzéshez. További hozzáadott értéke lehet a foglalkozási nómenklatúra-kategóriákon, vagy specifikusabban a foglalkozásokon, vagy akár az egyes munkakörökön belüli vagy közötti összehasonlításnak is.

Az információgyűjtés során azt az alapfeltevést használom, hogy jelen tézis keretei között elérhető módon, a kompetenciák iránti kereslet legjobban az álláshirdetésekben képződik le. Mivel az emberierőforrás-szükségletet, azaz azt, hogy milyen munkaerőre van szüksége egy adott cégnek, az elvégzendő feladat, illetve a betöltésre szoruló pozíció alapján lehet legjobban megfogalmazni, és kommunikálni a potenciális jelöltek felé, így az álláshirdetésekben valamilyen mértékben szükségszerűen meg kell jelennie a vállalatok kompetencia-szükségletének.

Ezt a feltevést megerősítik a szakirodalomban például Pitukhin és szerzőtársai, akik azt írják, hogy a legtöbb szakmai követelmény, amit a munkáltató a jelentkezővel szemben állít, megjelenik az állásajánlatokban (2016, p. 2028). Wowczko szintén kiemeli, hogy az online toborzás megjelenésével hatalmas mennyiségű, potenciálisan hasznos információ áll a kutatók rendelkezésére a keresett kompetenciákról (Wowczko, 2015, p. 34). Zhao és szerzőtársai (2015, p. 4013), a CareerBuilder.com kutatói, fél milliárd angol nyelvű álláshirdetésből vett minta vizsgálata alapján azt találták, hogy azok kilencven százalékában megjelennek a jelen dolgozat szempontjából kompetenciaként elfogadható kifejezések. Az előzőekkel ellentétben Nasir és szerzőtársai (2020) tapasztalatik között megemlítik, hogy az általuk

feldolgozott hirdetések inkább a pozíciók leírását (*vacancy information*) emelik ki, és kevésbé részletezik az elvárt készségeket.

A kidolgozni kívánt keretrendszer legfontosabb bemeneti adatai tehát álláskereső portálokról legyűjtött álláshirdetések. Így az inputok folyamatossága is biztosított a trendek elemzéséhez, hiszen az igények folyamatosan jelennek meg ezeken az oldalakon. A rendszert fel kell készíteni arra is, hogy a legyűjtött álláshirdetések aktivitását azok teljes életciklusa alatt monitorozza. Ilyen módon, a már legyűjtött adatokat folyamatosan nyomon követve és azokhoz egy érvényességet jelző paramétert karbantartva, követhetővé válhat, hogy egy adott hirdetésben szereplő pozíciót mikor töltöttek be. Így a meghirdetéstől a betöltésig terjedő "aktivitási" periódus is rögzíthető. Ez alapján, azaz hogy az adott hirdetés mennyi ideig volt „nyitva", szintén számos következtetés levonható a kapcsolódó pozícióhoz szükséges kompetenciákra vonatkozóan is. Például ha egy pozíció betöltéséhez a jelentkezőknek egy jól definiált kompetenciakészlettel kell rendelkezniük, és az adott pozíciótípust általában nagyon hamar betöltik, úgy feltételezhető, hogy nagyon sokan rendelkeznek a szükséges kompetenciákkal, például azért, mert az adott pozícióban használt technológia nagyon népszerű, magasan helyezkedik el a Hype görbén stb. Ez utóbbi szintén információértékkel bírhat az érintettek számára.

2.1 Kutatási kérdések

Jelen értekezésben, ahogy azt az előzőekben is kifejtettem, egy olyan rendszer tervét kívántam felvázolni, és legfontosabb, az egész megoldás alapját biztosító moduljainak megvalósíthatóságát módszerekkel és modellekkel alátámasztani, melynek fő céljául tűztem ki, hogy olyan információkat szolgáltasson, melyek az álláshirdetésekben megjelenő foglalkozásokról, az igényelt kompetenciákról, végzettségekről és képzettségekről, illetve mindezek időbeli alakulásáról adnak képet riportok, elemzések stb. formájában. Egy ilyen rendszer implementációja természetesen nem egyemberes feladat. A disszertáció keretei arra adnak lehetőséget, hogy elméleti megalapozottságú javaslatot tudjak tenni az adatok tárolásának módjára, elkezdjem az adatgyűjtést és erre építve megvizsgáljak olyan módszereket, melyek segítségével az álláshirdetések leírásaiban a kompetenciaelemek, illetve a kapcsolódó foglalkozások beazonosíthatóak. Ennek megfelelően határoztam meg a kutatási kérdéseimet is.

A kutatásom tehát feltáró jellegű. Az igazoló jellegű kutatásokkal ellentétben, „a feltáró jellegű kutatások tipikusan három célból készülnek: a téma jobb megértését biztosítják, egy későbbi alaposabb kutatás megvalósíthatóságát tesztelik, és további kutatások számára fejlesztenek alkalmazható módszereket” (Varga, 2014, p. 4; Szabó, 2000).

A rendszer elsődleges input adatai internetes álláskereső portálokon közzétett hirdetések, melyek tárolására több probléma miatt is szükség van. Egyrészt mivel azok internetes forrásból származnak, nem garantálható, hogy elérhetőek lesznek a teljes elemzési ciklus alatt. Például egy álláshirdetés esetében, mikor az többé már nem releváns a hirdető számára – azaz a pozíciót betöltötték – lekerül az állásportálról, de számunkra továbbra is fontos a benne tárolt információ, hiszen a tervezett elemzésekhez a múltbeli adatokra is szükségünk van. Másrészt az álláshirdetések zajossága miatt elő kell azokat készíteni az elemzéshez. Az előkészítésre azért van szükség, mert azok az információk, melyek az elemzéseink alapjául szolgálnak, nem mindig állnak rendelkezésre strukturált formában, vagy akár explicit módon az álláshirdetésekben, így azokat annak szövegezéséből kell valamilyen módon kinyerni, és amennyiben ez az előfeldolgozás megtörtént, érdemes a feltárt információkat letárolni. Az első kutatási kérdésem tehát azt vizsgálja, hogy hogyan érdemes kidolgozni az adatok tárolására szolgáló megoldást, illetve milyen adatköröket érdemes gyűjteni.

1. Kutatási kérdés: Mi a legmegfelelőbb eszköz (adattárolási platform) az internetes forrásokból legyűjtött – rendkívül heterogén, nagy mennyiségű, strukturálatlan adat – álláshirdetések, illetve kapcsolódó gazdasági és statisztikai tények tárolására, oly módon, hogy az így rögzített információ alapján az érintettek számára értékes elemzéseket lehessen adni, melyek a munkaerőpiacon igényelt kompetenciák időbeli, illetve egyéb dimenziók mentén történő alakulását mutatják be.

1.1. Milyen adatköröket érdemes gyűjteni és mi lehet az adatok forrása? A dolgozatban megvizsgálom az adatgyűjtéshez használható technológiákat, és bemutatom annak a „keresőrobotnak” az implementációját, mellyel a rendszer megalapozásához és elvégzett kísérletekhez szükséges adatokat összegyűjtöttem.

1.2. Milyen szempontok szerint érdemes kiválasztani az adattárolási platformot? A dolgozat elméleti részében összegyűjtöm azokat a megfontolásokat, amelyek aztán a

jelen probléma szempontjából legmegfelelőbb rendszer kiválasztásának alapját képezhetik. Megvizsgálom, hogy melyik az az adattárolási struktúra, ami legjobban szolgálja a felépíteni kívánt rendszer céljait, azaz hogy melyik az a tárolási technológia, amely a célnak legjobban megfelel, például egy hagyományos adattárház megoldás, vagy egy *big data* környezetben divatos adattó (*data lake*). Továbbá a felvázolt szempontok alapján a dolgozat első részében összehasonlítok konkrét adattárolási architektúrákat is, mint a hagyományos relációs, az oszlopalapú, *in-memory* és az idősorozat adatokra optimalizált adatbázisok, a kulcs-értékpár- és dokumentumtárak, azzal a céllal, hogy a felvázolt koncepció helyességét alátámasszam az egyes megoldásoknak a választás alapjául szolgáló szempontok szerinti összehasonlításával.

1.3 Kitérek továbbá arra is, hogy hogyan érdemes a legyűjtött adatokat betölteni és sémába rendezni, szükséges-e egyáltalán sémákat definiálni a tároláshoz, mint ahogy egyes, elterjedt adattárház architektúrák esetében, ahol általában egy *ETL* (*extract, transform, load*) folyamat során előre meghatározott relációs adatmodellbe töltik be a megtisztított és rendszerezett adatot. Ha igen, hogyan nézzen ki ez a séma? Vagy hatékonyabb az adatokat – mint egy adattó esetében – abban a formában tárolni, ahogy legyűjtöttük a forrásból – például egy elosztott fájlrendszeren – míg a logikát az elemzés, feldolgozás során alkalmazzuk, ami történhet egy *MapReduce* algoritmus vagy egy *Spark* alkalmazás segítségével?

Összegezve tehát fontos megvizsgálandó kérdés, hogy melyik megközelítés a legmegfelelőbb a célra, egy adattárház, egy adattó vagy éppen egy hibrid megoldás. Kapcsolódó kérdés továbbá, hogy érdemes-e az állásajánlatok adatait előre definiált sémákban tárolni (*schema-on-write*) és egy *ETL* folyamatot építeni az adattisztítás és betöltés elvégzésére, avagy az adatokat a legyűjtött formában érdemesebb-e inkább tárolni, majd a logikát igény esetén az adatok elemzése során alkalmazni (*schema-on-read*).

A szükséges adatkörök és a számukra legmegfelelőbb tárolási megoldás feltárása után, legyen az egy adattárház vagy más, az összegyűjtött adatok feldolgozását végzem el annak érdekében, hogy megalapozzam a kutatás későbbi szakaszait, ahol a cél a kapcsolat kiépítése lesz a munkaerőpiaci kereslet és az oktatási kínálat között, és az alap megteremtése azok későbbi harmonizációjához. Jelen tézis feltevése, hogy ez a kapcsolat kompetencia alapon építhető fel, ennek megfelelően a dolgozat egyik célja,

a keresleti oldalon, az álláshirdetésekben megjelenő kompetencia elemek beazonosítása.

2. Kutatási kérdés: Milyen információtechnológiai megoldásokkal lehet az egyes szabadszöveges leírásokban, például jelen tézis esetében álláshirdetésekben – de hasonló módon akár folyamatmodellek feladatleírásaiban (*task description*) – explicit módon megjelenő kompetencia-elemeket automatikusan beazonosítani? Illetve az egyes megoldásokkal milyen pontosság és felidézési arány érhető el?

Az álláshirdetések, vagy a folyamatmodellek feladatleírásai általában tartalmazzák a kompetenciáknak azt a legfontosabb listáját, melyekre egy munkavállalónak szüksége van az adott álláshirdetésben leírt pozíció betöltéséhez vagy adott feladat elvégzéséhez. Azonban a szabadszöveges korpuszban ezen kompetenciákat reprezentáló n-gramok (szavak és tetszőleges *n* elemű kifejezések) beazonosítása nem triviális. A kutatási kérdés vizsgálata során arra keresek megoldást, hogy hogyan, illetve milyen eszközökkel lehetséges az álláshirdetések korpuszában ezeket a releváns tudáselemeket, kompetenciákat reprezentáló, explicit megjelenő n-gramokat beazonosítani.

A kompetenciák beazonosítása történhet például szövegbányászati eszközökkel. Ezt a folyamatot külső források felhasználásával is támogathatjuk, mely források mintegy kompetenciaszótárként segíthetik az elemek beazonosítását az álláshirdetésekben. Ilyen külső források lehetnek például az ESCO ontológia készség (*skill*) pillére, vagy a STUDIO ontológia, melyeket részletesen a 4.1. alfejezetben ismertetek. Egyéb nomenklatúrákat szintén meg lehet vizsgálni, hogy hasznosíthatók-e a kompetenciaelemek beazonosítása során, akár azáltal, hogy segítenek az adott pozícióhoz kapcsolódó foglalkozás meghatározásában. Mivel ezek a statisztikai gyökerű elnevezési rendszerek általában sok helyen összehasonlítások alapját képezik, ezért rendkívül stabilnak kell lenniük, illetve az összehasonlíthatóság érdekében kellően sztenderdizáltnak. Ez egyfelől hasznos lehet jelen kutatás során, hiszen biztos alapot nyújthatnak a szövegbányászati feladatok elvégzéséhez, másfelől, mivel idejétmúlt információkat is tartalmazhatnak, így csak fenttartásokkal lehet őket kezelni.

Az állásajánlatokban kompetenciainstanciák jelennek meg, hiszen azok a vállalatok konkrét erőforrásszükségleteit tükrözik. Amennyiben ezek az instanciák a

hirdetésekben beazonosíthatók, későbbi kutatás tárgya lehet, hogy miként lehet megtalálni a hozzájuk kapcsolódó absztrakt kompetenciát, azaz hogyan lehet a kompetencia-instanciákat absztrakt szintre visszavezetni a kapcsolatok minél hatékonyabb leképezése érdekében? Egy ezt megalapozó kérdés, melyet a dolgozatban részletesebben is vizsgálni kívánok, hogy miként lehet az állásajánlatokban látensen, implicit, rejtett módon „jelen levő" kompetenciaelemeket feltárni?

3. Kutatási kérdés: Milyen módszerekkel, illetve milyen technológiák segítségével lehet az implicit (látens), az álláshirdetésekben közvetlenül nem megjelenő, de az azok által meghatározott kontextusban releváns kompetenciaelemeket feltárni? Milyen adatforrásokra lehet és érdemes támaszkodni ezen rejtett objektumok beazonosításához?

Az álláshirdetések explicit kijelölnek olyan kompetenciákat, melyek meglétére szükség van az adott pozíció ellátásához, ugyanez igaz a folyamatmodellek esetében a munkakör vonatkozásában. Ezek az expliciten megjelenő kifejezések – szemantikusan (például jelentésük vagy a köztük lévő kapcsolatok alapján) vagy egyszerű statisztikai alapon (például együttes előfordulás, az egyes elemek távolsága stb.) – kijelölhetnek olyan, expliciten nem megjelenő, látens kompetenciákat is, melyek szintén relevánsak lehetnek az adott pozíció, vagy munkakör kontextusában.

A mesterképzés során írt szakdolgozatomban azt a kérdést vizsgáltam meg, hogy miként lehet tudáselemek egy listája alapján részterületeket, és kapcsolódó kompetenciákat beazonosítani a STUDIO ontológiában (Neusch, 2014). Az ott leírt módszer alkalmas arra is, hogy az explicit az álláshirdetésekben megjelenő kompetenciák alapján – az ontológia segítségével – feltárjam azokat az immanens tudás- és készségelemeket, melyek az adott pozíció betöltéséhez szükségesek lehetnek. Jelen tézisben olyan módszereket fogok megvizsgálni, melyek szintén alkalmasak lehetnek ennek a rejtett információnak a feltárására.

2.2 A kutatás keretei

A kutatás elsődleges célja tehát egy munkaerőpiaci „adattárház" koncepció kidolgozása, amiben a legyűjtött álláshirdetések és a hozzájuk kapcsolódó, illetve belőlük kinyert egyéb dimenzió, illetve tényadatok tárolhatóak. Ez a platform lehet a „magja", központi eleme a később köré építendő bővebb rendszernek. Az adatok közül a tézis célja szempontjából legfontosabbak a kompetenciaigények, így a kutatás fontos

mérföldkövei közé tartozik azok beazonosítása, gyűjtése és szemantikus gazdagítása a szöveges bemeneti adatok alapján, egy adott objektumtípushoz – a dolgozatban felvázolt felhasználási eset (*use case*) esetében egy pozícióhoz – kapcsolódóan (ami tulajdonképpen itt a granularitást, a feldolgozás legkisebb egységét jelöli). A dolgozat jelentős része ennek megfelelően tehát az álláshirdetésekben explicit megjelenő, illetve az azokhoz implicit kapcsolódó kompetenciák beazonosításának lehetőségeit tárgyalja. A szemantikus kontextusba helyezés alatt egy ontológia felhasználását értem, annak érdekében, hogy segítségével fel tudjam tárni a meghirdetett pozíciókhoz kapcsolódó foglalkozásokat, végzettségeket, tudáselemeket és kompetenciákat és az azok között meglévő kapcsolatokat.

Az adattárháztól és a köré épülő keretrendszertől azt várom, hogy segítségükkel olyan kérdésekre is választ találhatok, hogy hogyan változik az egyes kompetenciák iránti munkaerőpiaci kereslet. Így olyan következtetéseket is le tudnának vonni a döntéshozók például, hogy ha egy adott kompetenciainstancia iránti kereslet nő, akkor érdemes azt oktatni, hiszen valószínűleg középtávon (3-5 év), az egyetemről kikerülő végzősöknek szükségük lesz az azt megalapozó készségekre és tudásra. Abban az esetben, ha az előzőekkel ellentétben egy adott kompetencia iránti kereslet jelentősen visszaesik, el lehet gondolkodni annak oktatásból való kivezetésről. Jelen dolgozathoz kapcsolódó munka során elsősorban infrastruktúra- és erőforráshiány miatt azonban nem tudom implementálni az – itt elméletileg megalapozásra kerülő – adattárházat, így a beazonosítható trendek és a tényleges elemzések bemutatása a kutatás későbbi szakaszaiban lesz csak lehetséges.

A munkaerőpiaci kínálatot sok különböző csoport alkotja, például a kezdő munkavállalók – köztük a frissen végzettek, a már tapasztalt és éppen munkahelyet váltók, vagy éppen azok, akik ilyen-olyan okokból éppen újraintegrálódnak, visszakapcsolódnak a rendszerbe stb. Jelen kutatás fókuszában ebből a szempontból a felsőoktatási szektor áll, ezen belül a friss diplomások, a dolgozat végső céljával összhangban, azaz hogy olyan információkat nyújtson az oktatók számára, ami alapján a képzési kimeneti követelmények olyan tartalommal tölthetőek meg, melyet a végzett hallgatók el tudnak adni a piacon.

A keretrendszer kidolgozása során a problémát egyszerűsítendő, a vizsgált állásajánlatok körét az informatikai szektorra szűkítem, ennek megfelelően fogom a külső ontológiák tartalmát – amit a kompetenciák beazonosításához, mintegy

szótárként használok – is megszűrni. Természetesen amennyiben a felvázolt koncepció működőképessége bizonyítást nyer (*proof of concept, PoC*), úgy a megoldás könnyen adaptálható más területekre is.

Jelen tézisben a teljes kutatási térnek csak a munkaerőpiaci kereslet oldalát érintő feladatok kerülnek kidolgozásra, tehát a munkaerőpiaci „adattárház" koncepciója és az, hogy miként lehet kinyerni a kapcsolódó kompetenciákat az álláshirdetésekből, illetve hogy egyes szemantikus forrásokat milyen módon lehet a feldolgozásba bekapcsolni a látens információ kinyerése érdekében. A dolgozatban nem foglalkozom a munkaerőpiac kínálati oldalával; az egyes tantárgyi adatlapok, illetve szakleírások feldolgozása, és a keresleti oldal elemzése során nyert információkkal való összevetése (*matching, mapping*) már túlmutat a tézis keretein.

A tézisben elsősorban a felvázolt koncepció helyességét kívánom vizsgálni, és a feldolgozott források (álláshirdetések) nyelvéül az angolt választottam. A választás egyrészt azért esett az angol nyelvre, mert az az „informatika nyelve", és számtalan szakkifejezés magyar fordításban nincs elterjedten használatban. Továbbá a magyar nyelv feldolgozása (a nyelv sajátosságai miatt) bonyolultabb és több erőforrást igényel, mint az angol nyelvé, így egy *PoC* kidolgozása során indokoltabb utóbbi használata. Végül pedig angol nyelven jelentősen több releváns álláshirdetés érhető el az interneten, így az input adatok számossága növelhető a magyar nyelvű feldolgozás esetéhez képest. Az előző okokból kifolyólag tehát a koncepció helyességének vizsgálatához jobban megfelelnek az angol nyelvű források, míg amennyiben a tézisben felvázolt modell működőképessége bizonyítást nyer, és igény mutatkozik annak eredményeire, akkor a kutatás későbbi fázisaiban a rendszert fel lehet készíteni a magyar nyelv kezelésére is. A dolgozat feltételezése továbbá, hogy az élvonalbeli technológiákhoz szükséges kompetenciákra mutatkozó igény az angol anyanyelvű piacokon jelenik meg először, mely feltevésről részletesebben a következő, a kutatás kiterjesztési lehetőségeit tárgyaló alfejezetben írok.

2.3 A kutatás jelentősége és lehetőségei

Az elmúlt években számos kutatás indult a kompetenciák témakörében több tudományterületen is. A pszichológia, a neurológia és a közgazdaságtan is vizsgálja a témát, persze más-más megközelítésből. Közgazdaságtani szempontból, új kompetenciák kiépítése a humán erőforrásban befektetés, és mint olyan, természetes

módon szeretnénk, ha ez a befektetés legalább megtérülne, de még inkább, hogy profitot termeljen. Munkáltatói szempontból ez a profit a vállalat versenyképességében, míg oktatáspolitikai szempontból az adott ország gazdasági teljesítőképességében nyilvánul meg. Az előzőekből adódik, hogy szeretnénk kontrollt gyakorolni a kompetenciákba fektetett erőforrásaink felett, és ezt a kontrollt legjobban szigorú és következetes tervezéssel tudjuk elérni.

Az oktatáspolitikusoknak kifejezetten fontos döntéseiknél figyelembe venni, hogy milyen tudásra és készségekre lesz szükség rövid- és középtávon a munkaerőpiacon, hogy miként hidalják át a „forradalmian átalakuló gazdaságban szükséges készségek és a munkakínálat között egyre növekvő" szakadékot (Fazekas, 2017, p. 6). Ezért a politikai, illetve oktatásügyben érintett döntéshozók számára felbecsülhetetlen információértékkel bírhat az, ha tudják, hogy hogyan változnak a kompetenciaigények az időben. Ennek megfelelően a kompetenciák absztrakciós szintjétől függően az értekezésben javasolt keretrendszer segítségével különböző döntéshozói szintek támogathatók. A tézisben vázolt megoldás kifejezetten a tárgy- és szakfelelősök támogatását célozza, de egy magasabb szinten az eredmények politikai döntéshozók számára is hasznosak lehetnek.

A felvázolni kívánt döntéstámogató infrastruktúra azonban nem áll rendelkezésre széleskörben elérhető módon, ami igazolja a dolgozat és a kutatás relevanciáját. Wowczko (2015) kutatása alapján azt találta, hogy bár a munkaerőpiaci kereslet és kínálat összehangolása szempontjából elengedhetetlen a keresett készségek figyelembe vétele a tantervfejlesztés során, a korábbi kutatások mégis hajlamosak kizárólag a foglalkozások végzettségigényére koncentrálni, ami persze fontos, de nem elégséges. Jelen dolgozat tehát ennek a hiánynak a pótlására indított kutatás egyik lépése.

Bár a szak- és tárgyfelelősök a dolgozatban felvázolt megoldás elsődleges érintettjei, azonban a keretrendszer által nyújtott információknak a hallgatók közvetlenül is haszonélvezői lehetnek. A kinyert adatok alapján például tanulási utak (*learning path*) ajánlhatóak számukra. Ez többféleképpen is elképzelhető. Egyrészt amennyiben egy hallgató tudja, hogy milyen munkát szeretne végezni a jövőben, azaz egy adott pályát választ és afelé orientálódik, akkor kidolgozható számára egy olyan egyedi tanulási út, aminek követésével pontosan és célzottan azokat a készségeit fejlesztheti, illetve azt a tudást szerezheti meg, ami a vágyott karrierpályának leginkább megfelel. Másrészt,

amennyiben fel tudjuk mérni egy adott szakon tanuló hallgató tudásában mutatkozó hiányosságokat, úgy tudunk számára nyújtani egy listát azokról az állásokról, amit jellemzően az adott szak elvégzése után hatékonyan be tudna tölteni, kiegészítve egy másik listával arról, hogy ahhoz, hogy egy adott pozíciót sikerrel meg tudjon pályázni, milyen kompetenciáit kell fejlesztenie, milyen tudáshiányokat kell pótolnia.

A munkavállalók számára ugyanilyen fontos lehet a kompetenciakereslet alakulásának ismerete, hiszen az élethosszig tartó tanulás zászlója alatt, a rendkívül gyorsan változó követelmények között nekik is folyamatosan naprakészen kell tartaniuk a kompetenciakészletüket annak érdekében, hogy versenyképesek maradjanak a piacon. Mivel a konkrét anyagi kiadások mellett egy új kompetencia kiépítése jelentős haszonáldozati költséggel is jár, nem mindegy, hogy az egyének mibe fektetik erőforrásaikat.

Összefoglalóan elmondható tehát, hogy a rossz vagy éppen jó befektetési döntések ezen a területen, a gazdaság összes szintjén, az egyéni, a vállalati és a nemzeti, illetve regionális versenyképességben, illetve teljesítményben is éreztetik hatásukat. A jó döntések elősegítéséhez a disszertációmban felvázolt keretrendszer felbecsülhetetlen információkat nyújthat.

A dolgozatban bemutatott használati esetnél a bemeneti adatok álláshirdetések, de a keretrendszer egyéb felhasználásai is elképzelhetőek, ahol az input adatok számos más forrásból származhatnak. Egy ilyen alternatív probléma lehet, melyre a dolgozatban felvázolt keretrendszer – apróbb módosítások után – megoldást kínálhat, a belső képzések esete. Az emberi erőforrás iránti kereslet és az allokációhoz kapcsolódó kérdések ugyanis két szinten jelennek meg. Egyrészt amíg a vállalat nem rendelkezik a megfelelő minőségű és mennyiségű emberi erőforrással, kereslete a munkaerőpiacra irányul. Illetve amennyiben a vállalat már rendelkezik a szükséges humán tőkével, a kérdés úgy változik meg, hogy miként lehet a meglévő erőforráskészletből úgy allokálni a munkavállalókat az egyes projektek és pozíciók között, hogy a legnagyobb megtérülést lehessen elérni a humántőke-befektetésen. Egy-egy céges, belső képzés akkor hatékony és kifizetődő, ha a munkavállalóknak azokat a kompetenciákat adja át, amikre a munkavégzés során szükségük lehet. Az egyes munkakörök betöltéséhez szükséges kompetenciák pedig általában a feladatleírásokban jelennek meg, kerülnek kifejtésre explicit módon. Így amennyiben egy cég rendelkezik kidolgozott folyamatmodellekkel, melyekben az egyes feladatok is megfelelően le vannak írva,

úgy feltételezhetjük, hogy ezekből a feladatleírásokból kinyerhetőek azok a kompetenciák, melyek a hatékony belső képzések szervezésének alapjául szolgálhatnak.

A dolgozatban felvázolt rendszer tervezése és képességei alapján a következő üzleti problémák megoldását is támogathatja:

- Munkaerő-kiválasztás, munkaerő-allokáció,
- munkaerő képzés (*on-the-job training*),
- folyamatfejlesztés,
- szakterületi ontológia gazdagítása.

A Budapesti Corvinus Egyetem Információrendszerek Tanszéke és a Jövő Internet Élő Laboratórium Egyesület együttműködésében számos kutatási projekt indult az oktatási típusokat (*formal, informal, nonformal*) támogató használati esetek vizsgálatára, melyeket a következő fejezetben röviden ismertetek.

Az oktatáshoz szorosan kapcsolódó használati eseteken kívül, az eredmények iparági elemzésekre is felhasználhatóak lehetnek. Amennyiben az ágazat, ahol a pozíciót meghirdető cég tevékenykedik, az állásajánlat alapján közvetlenül, vagy a foglalkozás, illetve a cég beazonosítása útján meghatározható, úgy a kompetenciák változása ezen az iparági szinten is elemezhető. Egy adott szektor kompetenciaigényeinek időbeli változása alapján például következtethetünk az adott ágazat automatizálódásának ütemére, azaz arra, hogy milyen ütemben váltja ki az élőmunkát a gépesítés.

Ez a trend valószínűleg a gyártásautomatizálás esetében a legszembetűnőbb, de egyéb, gyártást nem végző iparágakban (például a bankszektor) is beazonosítható. Itt például a kontrolling és riportoló rendszerek fejlesztésével, azok gépi tanulási algoritmusokkal való kiegészítésével a kontrollerek feladatköre lecsökkenhet, szinkronban a bankok kontrollerek iránti erőforrásigényével. Ez a trend persze valószínűleg inkább összefüggéseiben mutatható ki, azaz például egyik oldalon megnő a pénzintézetek igénye azon IT szakemberekre, akik rendelkeznek gépi tanuláshoz kötődő kompetenciákkal, míg az üzleti oldalon pedig a kontrolling- és egyéb „kiváltott” kompetenciák iránti igény csökken.

Hasonló módon elképzelhető, hogy kimutatható olyan trend, hogy az algoritmikus kereskedési megoldásokhoz kapcsolódó kompetenciákra irányuló megnövekedett kereslet együtt jár azzal, hogy a konkrét tőzsdei kereskedők iránti kereslet lecsökken. Egy ilyen forgatókönyv esetében például, a dolgozatban felvázolt koncepció alapján, a tőzsdei kereskedők azon algoritmusokhoz, illetve programozáshoz kapcsolódó hiányzó kompetenciái könnyen beazonosíthatóvá válnának, melyek elsajátításával a már meglévő munkavállalók hatékonyan átképezhetőek lennének a megváltozott kompetenciaigényeknek megfelelően. Egy ilyen esetben az átképzés több szempontból is hatékony lehet. Egyrészt azért, mert a kereskedők már rendelkeznek a szükséges üzleti ismeretekkel, másrészt azért, mert az őket alkalmazó cégek elkerülhetik az új ember alkalmazásával járó veszélyeket és nehézségeket. A probléma ilyen megoldásához addicionálisan az a szociális nyereség is társul, hogy az adott kereskedők nem veszítik el egyik napról a másikra megélhetési forrásukat, amennyiben vállalják az átképzéssel, változással járó kényelmetlenségeket.

A tézis egyik sajátossága, hogy az Egyesült Királyság piacára szánt álláshirdetések elemzése útján szándékozik a magyar felsőoktatás döntéshozói számára támogatást nyújtani. A feltevésünk az, hogy a magas hozzáadott értékű iparágakban (például informatika) igényelt kompetenciákban megjelenő keresletváltozás előbb mutatkozik meg ezekben a régiókban, így a legújabban keresett kompetenciák gyorsabban, illetve előbb fognak visszatükröződni az ezen – nyugati – piacokra szánt álláshirdetésekből. Amennyiben ez a feltevés megállja a helyét, és a legújabb munkaerőpiaci trendek némi eltolódással szivárognak át a magyar piacra, úgy ez lehetővé tenné a még pontosabb előrejelzések nyújtását az eredményeket felhasználó magyar döntéshozók számára, és némi helyzeti előnyt adna a tantervek kidolgozása során. Ez alatt azt értem, hogy ha hamarabb el tudják kezdeni ezen technológiák oktatását a hazai képzőintézményekben, akkor a még meg nem jelent igény későbbi felmerülése esetén proaktívan tud a magyar munkaerőpiac reagálni. Bár jelen dolgozatban ezen feltevés helyességét nem kívánom vizsgálni, a kutatás későbbi szakaszainak tárgya lehet, hogy megállja-e a helyét a hipotézis, mely szerint a kompetenciaigények változása időben eltolódva, némileg lassabban fejti ki a hatását a magyar munkaerőpiacra.

Az összegyűjtött álláshirdetések regionális összehasonlítások alapjául is szolgálhatnak a különböző régiókban megjelenő kompetenciaigények összevetése által. Egy másik érdekes és hasznos vizsgálati kérdés lehet a kutatás későbbi fázisában földrajzi alapú

összehasonlítást végezni, a feladatokat inkább kiszervező és az inkább *outsourcing* célpont országok között is. Jelenleg számos informatikai területen Magyarország *outsourcing* célpont, így a hazai döntéshozói stratégia támogatására érdemes lehet azokat a kompetenciákat beazonosítani, melyeket a fejlettebb országokban működő vállalatok inkább kiszerveznek, mindezt még akkor, mikor ezek a trendek elkezdenek kibontakozni. Ilyen kiszervezett munkakörök az elmúlt években jellemzően például a szoftvertesztelő, illetve a szoftverminőség-ellenőrző. Az ezekhez a munkakörökhöz kapcsolódó feladatokat, a nyugati szoftvercégek jellemzően kiszervezték fejlődő országokba, mint például India. Amennyiben hasonló globális munkaerőpiaci trendeket tudunk beazonosítani, úgy Magyarország helyzeti előnybe hozható a magyar munkavállalók megfelelő felkészítésével, azaz a szükséges kínálat kiépítésével.

2.4 Kapcsolódó kutatások

Jelen tézisben felvázolt problémák vizsgálata beleilleszkedik a Budapesti Corvinus Egyetem Információrendszerek Tanszék és a Jövő Internet Élő Laboratórium Egyesület által végzett, számos kutatást magába foglaló munkába.

A ProKEX[10] elnevezésű EUREKA projekt keretében kidolgozott rendszer célja a vállalatok intellektuálistőke-menedzsment tevékenységeinek támogatása, az egyes munkakörök megfelelő ellátásához szükséges tudáselemek feltérképezésével, rendszerezésével és könnyen átadható formába hozásával (Varga, 2014). A kifejlesztett alkalmazás, a folyamatmodellekben tárolt tudás kinyerésén keresztül következtet arra, hogy az egyes pozíciók milyen kompetenciákat igényelnek (Török, 2014). Ez az információ azután a STUDIO rendszerben kerül leképezésre olyan formában, mely aztán könnyedén használható belső munkaerőkiválasztáshoz vagy képzésekhez. A projekt keretében kifejlesztett rendszer támogatja a vállalatok munkaerő-kiválasztásra, illetve allokációra, munkaerő-képzésre és folyamatfejlesztésre irányuló erőfeszítéseit, azon keresztül, hogy képes:

1. a vállalati folyamatmodellekben rögzített feladatleírások alapján a munkakörökhöz szükséges tudáselemek kinyerésére,

[10] ProKEx: Integrated Platform for Process-based Knowledge Extraction, EUREKA project

2. a kinyert tudáselemhalmaz gazdagítására, olyan releváns, külső forrásból származó tudáselemekkel, melyek explicit módon nem jelentek meg a kiindulási folyamatmodellben,
3. a 2) pontban leírt módon előállt, már gazdagított tudáselemhalmaz mappelésére egy szakterületi ontológiával. És e lépés során az eredeti halmaz tovább tudáselemekkel való bővítésére a mögöttes szemantika alapján.
4. A 3) pontban leírt leképezés eredménye alapján a rendszer képes egy olyan struktúra leszabására a szakterületi ontológiából, amely egy adott munkakör betöltéséhez szükséges tudást reprezentál, és alapot nyújthat a munkavállalóknak az adott kontextusban mutatkozó tudáshiányainak feltérképezésére (Neusch és Gábor, 2014).

A SMART[11] projekt keretében a munkaerőpiaci kereslet és az oktatási rendszer kínálata közötti megfeleltetésen keresztül a cél egyes üzleti szektorokban jelentkező tréningigények feltérképezése volt. A hiánykompetenciák feltárásával olyan tantervek dolgozhatók ki a projekt keretében létrehozott rendszerrel, melyek segítségével az egyes szektorokban leginkább hiányzó szakemberek képezhetők ki. A projekt regionális fókusszal elsősorban a felnőttképzésre koncentrált, és az eredményeket az andalúziai turizmus szektor példáján tesztelték (Castello *et al.*, 2014). A SMART projektben használt központi rendszerkomponens szintén a STUDIO, melyben szakértők segítségével leképezésre került a turizmus területe részontológia formájában. A rendszerhez továbbá kifejlesztettünk egy felhasználói interfészt, melyen keresztül egy adott képzőintézmény képviselője kiválaszthatja azokat a kompetenciákat az ontológiából, melyeket az adott helyen oktatnak, fejlesztenek, így alakul ki a kínálatot reprezentáló struktúra (Caballero *et al.*, 2014), amely a STUDIO-ban úgynevezett Fogalomkörként képződik le (4.1.2. alfejezetet).

A munkaerőpiacon keresett kompetenciák szintén leképezésre kerültek – jelentős manuális munkabefektetéssel – egy taxonómiába. A két struktúra pedig egy ontológia megfeleltetési (*matching*) algoritmus segítségével összevetésre kerül, melynek eredményeképpen egy riport jön létre a képzőintézmény számára. A riport a kompetenciákat a következők szerint osztályozza:

[11] Supporting dynamic MAtching for Regional development, LLP – Leonardo da Vinci TOI projekt

- A képzőintézmény oktatja, de a piacon nem keresik.
- A képzőintézmény nem oktatja, bár a piacon lenne rá kereslet.
- A kereslet és a kínálat halmazainak metszetébe eső kompetenciák (Szabó és Neusch, 2015).

Jelen tézis számos ponton kapcsolódik az előző oldalakon bemutatott kutatásokhoz, illetve épít azok elméleti eredményeire. A ProKEX és SMART projektekhez hasonlóan, jelen kutatás során is támaszkodom például ontológiák felhasználására, azonban egy kicsit tovább is megyek, mivel felhasználok gépi tanulási módszereket is a lehetséges kompetenciajelöltek és a kapcsolódó foglalkozások álláshirdetésekben való beazonosításához. Vizsgálom továbbá nem csak az explicit megjelenő, de a látens módon kapcsolódó kompetenciák beazonosíthatóságát is, illetve feltáró jelleggel a kutatást egy tágabb kontextusba helyezem egy munkaerőpiaci adattárház architektúrájának felvázolásával.

A Budapesti Corvinus Egyetemen kívül más kutatói műhelyekben is vizsgálták a jelen dolgozat kompetenciadefiníciójának megfelelő kifejezések beazonosíthatóságát különféle dokumentumokban. Wowczko (2015) kutatásában IT álláshirdetések elemzésével, a munkaerőpiac keresleti és kínálati oldalának összehangolását célozta meg „készség" (*skill*) alapon[12], ami olvasatában egy „dinamikus változó, ami számos más aspektustól, mint például a földrajzi elhelyezkedéstől, az időtől vagy éppen az iparágtól függ" (Wowczko, 2015, p. 31). A kutatás a szakirányú szakmai tréningek (*vocational training*) tananyagfejlesztését kívánja támogatni. Wowczko a hirdetések címe alapján sorolta azokat 7 előre meghatározott foglalkozási kategóriába, mint például fejlesztő, elemző, stb. A szövegbányászat segítségével egyértelműen besorolható hirdetések alapján *k*-legközelebbi szomszéd (k-NN) modellt épített, és ezt a teljes korpuszra alkalmazva az összes hirdetést foglalkozási osztályokba sorolta. A készségeket az álláshirdetések szövegéből nyerte ki, de csak a két elemű kifejezésekkel foglalkozott. Az egy-egy adott foglalkozáshoz kapcsolódó 20 leggyakoribb bigramot szófelhőben ábrázolta. Az így generált szófelhők bár tartalmaznak irreleváns, illetve zajos adatokat, de a szerző következtetése szerint kielégítően reprezentálják a hirdetésekben megjelenő pozíciók kompetenciatartalmát.

[12] A Wowczko által használt készség fogalom értelmezése jelentősen átfed a jelen dolgozatban használt kompetencia értelmezéssel.

Pitukhin és munkatársai (2016) célja jelen munkához hasonlóan a felsőoktatási (Petrozavodszk Állami Egyetem) tanmenettervezési folyamat támogatása, egy többdimenziós, „egyesített” (*unified*) kompetenciaontológia létrehozásával. Többdimenziós alatt a szerzők azt értik, hogy az álláshirdetésekből nem csak magukat a kompetenciákat gyűjtik ki, és rendezik ontológiába, hanem mintegy metaadatként, azokhoz kapcsolódóan a kompetenciatípusokat, például „Végzettség”, „Ismeret”, „Készség” stb., a foglalkozásokat és az ismereti szinteket, például „Alap” vagy „Szakértői” is. A szerzők 100, angol nyelvű állásportálról legyűjtött hirdetésben összesen 396 kompetenciát azonosítottak be, melynek módszerére a tanulmány nem tér ki. Az előzőekben beazonosított információkból a szerzők felépítették 4 dimenziós ontológiájukat, melyre alapozva kidolgoztak egy kompetenciaelemeket automatikusan beazonosító algoritmust.

A szerzők az algoritmus részleteire szintén nem térnek ki, viszont a tanulmány alapján az elmondható, hogy a minőségi kifejezések beazonosításához felhasználták a kontextust is. A dimenziókban ábrázolt metaadatok környezetében előforduló kifejezéseket nagyobb valószínűséggel lehet elfogadni az ő terminológiájukban a „kompetencia tárgyának”. Példának hozzák a „*Bachelor's Degree in Computer Science*” kifejezést, melyből a kompetencia tárgya a „*Computer Science*”, melyet nagyobb valószínűséggel lehet információt hordozó elemként elfogadni így, hogy környezetében megjelenik a „*Degree*” kompetenciatípus (Pitukhin *et al.*, 2016, p. 2029). Jelen dolgozatban ismertetett munka során a kontextust az előzőekhez hasonló módon, a hirdetések címének felbontásakor, a foglalkozások beazonosítása során magam is felhasználtam (7.1. fejezet).

Nasir és szerzőtársai (2020) 465 darab, elemző munkakörökhöz kapcsolódóan, 2019 júniusában megjelent hirdetés címét és törzsét vizsgálták. A szerzők a szöveg előfeldolgozása során eltávolították a stopszavakat, majd meghatározták a hirdetések közötti távolságokat a dokumentumok szózsák modellben való ábrázolásának segítségével. A távolságok alapján hierarchikus klaszterezéssel csoportokba sorolták az álláshirdetéseket, hogy a közöttük potenciális meglévő kapcsolatokat feltárják. Ezzel a módszerrel a szerzők 5 nagy klasztert tudtak beazonosítani az egyes elemzői állásajánlatok között, amiket a „szenior”, „támogató”, „üzleti”, „pénzügyi” és „számviteli” kategóriacímkékkel tudtak ellátni. A létrejött csoportok leggyakoribb kifejezéseit szófelhő módszerrel vizualizálták a kutatók. Az eredmény, hasonlóan a

Wowczko által publikáltakhoz, meglehetősen zajos és általános, azonban a használt módszertan jó kiinduló pontja lehet egy hasonló irányú kutatásnak.

Gugnani és Misra (2020) egy internetről legyűjtött pozícióleírásokat önéletrajzokkal készség alapon megfeleltető ajánlórendszert vázolnak fel, melyhez egy komplex NLP és szövegbányászati módszereket is alkalmazó metodológiát dolgoztak ki. A készségeket azonosító kifejezések feltárására kidolgozott feldolgozási folyamat egyik bemenete egy szótár „halmaz", mely a computerhope.com oldal zsargonszótárából, az O*NET oldalán az egyes foglalkozásokhoz társított készségekből és a Wikipédia-gráf egyes elemeiből áll. Az iteratív folyamatban helyet kaptak különböző morfológiai alapú módszerek, mint például névelem-felismerés és szófaji egyértelműsítés[14], statisztikai, mélytanulási és szabály alapú módszerekkel kombinálva. A három készségszótár és a morfológiai elemzés (*NER*, *POS*) alapján Gugnani és Misra a pozícióleírások kifejezéseihez valószínűségi értékeket rendel, ami egy $[0, 1]$ skálán számszerűsíti, hogy az adott kifejezés mekkora valószínűséggel fogadható el kompetenciaként. Egy *Word2Vec* (*W2V*) modellt is alkalmaznak, melynek eredményét szintén összevetik a szótárak elemeivel, kiszámolva a páronkénti koszinusz távolságokat, és az adott kifejezéshez tartozó legnagyobb értéket szintén figyelembe veszik a további feldolgozás során valószínűségi értékként. Az így definiált 6 valószínűségi érték súlyozott arányaként adódik végső metrikájuk (*relevance score*), ahol a különböző módszerekkel kapott valószínűségi értékeket eltérő súllyal veszik számításba. Egy kifejezést empirikus alapon akkor fogadnak el készségként, ha a hozzá tartozó relevancia érték 0.35 fölött van. Felkért 4 független szakértő annotációja alapján a modelljük precizitása (*precision*) 0,78 míg a felidézés[13] (*recall*) aránya 0,88.

A szerzők jelen dolgozathoz hasonlóan használják az implicit készségek fogalmát, azon készségeket értve alattuk, melyek a pozícióleírásokban explicit nem jelennek meg, de a földrajzi adottságok, az iparág vagy a munkakör miatt implicit relevánsak lehetnek. Ezt az információt a hasonló pozíciók alapján próbálják feltárni.

Összességében Gugnani és Misra szerzők cikkéről elmondható, hogy részleteiben tárgyalja az alkalmazott módszereket és az eredményeket. A többi, előzőekben

[13] A *precision* és a *recall* szavak fordítása Tan et al. alapján történt (2011). A felidézés szinonimájaként Sebők és szerzőtársai (2016, p. 53) használják még a „fedés" és a „teljesség" szavakat is. A precizitást „pontosság", illetve „megbízhatóság" néven is hivatkozzák, amit az előző mű fordítója igyekszik elkerülni.

bemutatott kutatás közül áttekinthetősége és az alkalmazott módszerek kifinomultsága is kiemeli. Kutatásuknak több olyan eleme is van, például az egyes részrendszerek által függetlenül generált relevanciaérték, melyek alkalmazása véleményem szerint is növelheti egy hasonló rendszer hatékonyságát.

Zhao, illetve Hoang és szerzőtársaik (2015; 2018) SKILL nevű rendszerüket ismertetik, mely szintén az önéletrajzokban és pozícióleírásokban megjelenő készség és tudáselemek stb. beazonosítására törekszik. A szerzők készségszótáruk felépítéséhez a CareerBuilder.com adatbázisában szereplő kifejezéseket („*seed phrases*") normalizálták (egyértelműsítés és duplikáció eltávolítás[14]), melynek érdekében egy komplex feldolgozási folyamatot építettek ki. Első lépésben a Wikipedia API-t használták, és rákerestek az egyes kifejezésekre. A visszakapott kategóriacímkéket ezután megfeleltették a *Standard Occupational Classification* (*SOC*) nomenklatúra alapján képzett kulcsszavaknak. Amennyiben egyezést találtak, úgy az adott kifejezést megtartották további feldolgozás céljából. Az előzőek alapján leszűrt adatokat az értelmi egyértelműsítés érdekében átfuttatták a Google Search API-on is, illetve egy Word2Vec modellt is felépítettek, arra a feltevésre alapozva, hogy a készségeket azonosító kifejezések valószínűleg egymáshoz közel jelennek meg az input dokumentumokban (Zhao *et al.*, 2015, p. 4014). A folyamat során nem csak a kifejezések normalizálása (értelmi egyértelműsítése) történt meg, hanem metaadatokkal is kiegészítették a tárolt objektumokat a szerzők, úgy mint kapcsolódó kifejezések, illetve azoknak a normalizált formától vett koszinusz távolsága stb.

Későbbi munkájukban a szerzők leírják az egyes készségkifejezések jelentés-egyértelműsítésének érdekében eszközölt fejlesztéseiket, melynek során a készségkifejezések környezetét elemezték, és a Metropolis-Hastings algoritmus segítségével sorolták a kettő- vagy többértelmű kifejezéseket csoportokba. Az algoritmus előnyeként kiemelik, hogy más klaszterezési módszerekkel ellentétben a csoportok számát, azaz hogy a kifejezéseket hány jelentéskategóriába sorolja az algoritmus, előre nem kellett definiálniuk (Hoang *et al.*, 2018, p. 4630).

Kutatásaikat Zhao, Hoang és szerzőtársaik szakterületi szakértők segítségével értékelték ki, akik az eredmények alapján megállapították, hogy mely beazonosított kifejezések fogadhatóak el ténylegesen készségként és melyek nem. A kutatást a többi

[14] lásd 3.3.4 alfejezet

ismertetett tanulmánytól megkülönbözteti, hogy a SKILL rendszer elsősorban üzleti, és nem akadémiai célokat szolgál. Kiemelendő, hogy a modellek küszöbértékeit empirikus alapon állapították meg.

2.4.1 A disszertáció megkülönböztető sajátosságai

Jelen disszertációban részletezett kutatás számos metodológiai elemében hasonlít a fent ismertetett tanulmányokhoz, mivel több szakember is gépi tanulási módszerek használatával kísérelte meg az álláshirdetésekben előforduló kompetenciák feltárását. Az általam alkalmazott módszerek között szintén megtalálható a morfológiai, NLP eszközöket használó előfeldolgozás, a reguláris kifejezések, a foglalkozáscímkék felbontására alapuló szabályok és a gépi tanulási algoritmusok használata. Nem találkoztam azonban olyan forrásművel, ahol az általam használt konkrét módszerek, azaz a logisztikus regresszió és a döntési fák használatáról számoltak volna be, akár a kompetenciák, akár a foglalkozás-megnevezések feltárása során.

Továbbá szintén nem kísérletezett senki – a kutatásom során elemzett szerzők közül – az álláshirdetések szövegének és külső ontológiák taralmának – több, különböző metrikával – páronként kiszámított hasonlóságértékei alapján tanítani *ML* algoritmusokat. Gugnani és Misra (2020) használják a *Word2Vec* modelljük eredményének egy általuk épített készségszótár elemeitől vett koszinusz távolságértékeit, de az pusztán végső formulájuk egy összetevője, és nem építenek arra gépi tanulási algoritmusok tanításánál. Hoang és szerzőtársai (2018) szintén felhasználják a koszinusz távolságot *W2V* modelljük súlyozásához, de ők sem a jelen dolgozatban használt módon, egy *ML* modell tanításához. Továbbá disszertációmban nem csak a koszinusz-távolságot, de számos más hasonlóságmetrikát is felhasználok.

További megkülönböztető sajátossága disszertációmnak, hogy a fenti módszereket egy tágabb kontextusban, egy munkaerőpiaci adattárház fejlesztéséhez kívánja használni, mellyel a későbbiekben a felsőoktatási kompetenciakínálat és kereslet jobb megfeleltethetőségét igyekszik elősegíteni.

3 A kutatás elméleti háttere

Jelen tézis, illetve a kapcsolódó kutatás három nagy elméleti területre épít, melyek az adattárolási technológiák, az ontológiák és a szövegbányászat. A gyakorlathoz szorosan kapcsolódó elméleti koncepciókat a konkrét feladat és az általam implementált, illetve tervezett megoldás mellett a kutatási keretrendszert ismertető fejezetekben írom le. Az egyes területek alapjait azonban a jelen elméleti háttérrel foglalkozó fejezetben mutatom be.

Az adattárolási technológiák rövid bemutatásával – a 3.1. alfejezetben – elméleti megalapozást kívánok adni a szempontrendszernek, ami alapján később a konkrét megvalósításra javaslatot teszek.

A kutatási kérdések vizsgálata során több feladat támogatásához különböző ontológiákat használtam fel. Külső ontológiákban tárolt adatok alapján építek fel egy kompetenciaszótárat, amely az álláshirdetésekben található készség, tudás és képesség stb. elemek beazonosításának alapeszközéül szolgál. Egy ontológiát használok fel továbbá arra is, hogy az álláshirdetésekben explicit nem megjelenő, de logikailag kapcsolódó, látens kompetenciákat feltárjam. Ezért a 3.2. alfejezetben bemutatom a technológia mögötti elméleti alapokat.

A szövegbányászat és a természetesnyelv-feldolgozás témaköréhez kapcsolódó megoldásokat és algoritmusokat elsősorban a kompetenciák beazonosítása során alkalmazok. A kompetenciaszótár használatának feltétele, hogy az álláshirdetések szövege megfelelően elő legyen készítve a feldolgozáshoz. Annak érdekében továbbá, hogy az egyszerű szövegegyezésen alapuló egyezőség felismerésénél hatékonyabb módszert tudjak alkalmazni, megvizsgálom a kulcsszóhasonlóság és -távolság mérésére használt algoritmusokat is a 3.3. alfejezetben.

3.1 Adattárházak és adattavak

Ebben az alfejezetben a kapcsolódó adattárolási technológiák egy magas szintű áttekintése olvasható, amelyben elsősorban nem matematikai, technológiai stb. alapokra fókuszálva adok áttekintést a területről, hanem gyakorlati irányból közelítve az egyes megoldások azon szempontjai mentén kívánom őket bemutatni, amelyekre a későbbiekben a javaslataimat építeni fogom.

A tranzakciós (működési) adatbázisok célja a gyakran ismétlődő tranzakciók minél gyorsabb kezelése és a konzisztencia biztosítása által a vállalatok zavartalan napi ügymenetének megalapozása. Egy tranzakciós adatbázisban az esetek túlnyomó többségében az adatokat relációs adatmodellben, táblázatos, általában normalizált formában tárolják. A táblázatban az egyedeket a sorok (rekord, reláció), míg azok tulajdonságait az oszlopok reprezentálják. A normalizálás az adatok táblákba szervezésében és a redundancia minimalizálásában, a tároláshoz szükséges tárhely optimalizálásában játszik fontos szerepet.

A 90-es években a korai üzleti intelligencia alkalmazások és az adatalapú döntéstámogatási rendszerek megszületése megteremtette az igényt a tranzakciós adatok stratégiai szempontú feldolgozására és elemzésére. Azonban az új, gyakran nagy komplexitású kérdések megválaszolására a tranzakciós adatbáziskezelő rendszerek nem voltak felkészítve. Egyrészt az egyes üzleti egységek adataikat a folyamatok mentén szervezett struktúrákban, gyakran egymástól elkülönülten működő, nem kapcsolódó adatbázis rendszerekben tárolták; másrészt a nagy erőforrásigényű analitikus lekérdezéseket legtöbbször csak munkaidőn kívül lehetett futtatni, hogy ne akadályozzák a rendszereket az üzlet operatív igényeinek kielégítésében. További problémát jelentett, hogy a tranzakciós rendszerekben a napi működéshez nem szükséges historikus adatkörök, melyek meglétét szabályozó nem írta elő, nem álltak rendelkezésre. Ezen problémákra adott megoldásként hozták létre az első adattárházakat, ahova az adatokat, egy adattisztítási folyamat után a vállalat összes operatív adatbázisából rendszeresen betöltötték az úgynevezett *ETL* alkalmazások segítségével, központosított formában, egyes üzleti entitások szerinti struktúrában, és azokat a stratégiai döntések támogatásához szükséges, akár hosszabb ideig (5-10 év) is tárolták.

3.1.1 Adattárház

Az adattárház, Inmon (2005) által adott definíciója alapján, adatok olyan döntéstámogatást szolgáló gyűjteménye, amely téma- vagy tárgyorientált, integrált, nem volatilis vagy tartós és időfüggő. A témaorientáltság (*subject oriented*) azt jelenti, hogy a tranzakciós, napi működést támogató adatbázisokkal ellentétben, melyeket a folyamatok mentén szervezik, az adattárházakat a szervezet számára fontos fő entitások, illetve témák köré építik fel. Integráltság (*integrated*) alatt Inmon azt érti, hogy az adattárházak több folyamat adataiból táplálkoznak, bemenetüket számos

operatív adatbázis adja, így a vállalatban az igazság központi forrásaként szolgálhatnak. További különbség, hogy míg egy tranzakciós adatbázis tartalmát naponta számos tranzakció módosítja, addig általában az adattárházakba került adatok tartósak (*nonvolatile*), azaz bár napi használatban vannak, különböző jelentések, analízisek alapjául szolgálnak, de nem jellemző, hogy változna a tartalmuk (*load-and-access processing*). Végül az adattárházak negyedik jellemzője, az időfüggőség (*time variancy*) azt jelzi, hogy az adattárház objektumaira vonatkozó értékek a bekerülés pillanatában, vagy egy bizonyos tranzakció idejében igazak, időbélyeget, a tranzakció idejét jelző értéket kapnak, és az adattárház egyik célja, hogy ilyen módon a tényadatok változását az időben követhetővé tegye (Inmon, 2005).

Az adattárházak másik elterjedt definíciója kifejezetten az üzleti igényekből indul ki. Kimball és Ross (2013) a működési adatbázisok (*database management system*, *DBMS*) és az adattárház illetve a hozzájuk kapcsolódó üzleti intelligencia rendszerek (*DW/BI*) megkülönböztetéséből indul ki. Az adattárházakat úgy definiálják, mint azok a rendszerek, ahol a működési adatokat a döntéstámogatást segítő lekérdezéseket és elemzéseket megkönnyítő struktúrában, több dimenziós modellekben, izolált környezetben tárolják. Pusztán céljait, és az Inmon által definiált alapvető attribútumait tekintve egy adattárház-megoldás tehát kézenfekvő lenne a munkaerőpiaci adataink tárolására, hiszen azok változását hosszú időn keresztül kívánjuk nyomon követni, és a riportok, elemzések stb., melyeket rájuk építve készíthetünk, stratégiai döntések alapjául szolgálhatnak.

Az adattárházak azon képessége, hogy jobban tudják támogatni a különböző szempontok szerinti lekérdezéseket és riportokat, abból az adottságból fakad, hogy a beillesztés és frissítés műveletek egy tranzakciós adatbázishoz képest relatíve ritkák, így több indexelést lehet hatékonyan megvalósítani bennük. Az adattárházak általában támogatják a multidimenziós indexelést is, és számos megoldás fejlett gyorsítótárazást kínál a gyakran használt adatok még gyorsabb elérésének biztosítására. További fontos különbség, hogy míg a működési adatbázisok esetében az adatok normalizált tárolása fontos cél, addig az adattárházak általában többdimenziós adatmodellt használnak.

Az adattárházak kialakítása általában a Codd és munkatársai (1993) által lefektetett elvek mentén történik, akik az adatoknak a multidimenzionális elemzését lehetővé tevő ábrázolási modelljét *OLAP* (*online analytical processing*) néven definiálták. „A multidimenzionális adatmodellben […] a multidimenzionalitás arra utal, hogy itt az

elemi adatokat nemcsak egy kulcs függvényében lehet elérni […], hanem több kulcstól való függése is nyilvántartott az adatbázisban" (Fajszi és Cser, 2004, p. 32). Az adatok multidimenziós tárolását általában három dimenzióban egy adatkockával szokták szemléltetni, ahol a dimenziók azok az attribútumok, melyek mentén a tényadatokat ábrázolni szeretnénk. Az *OLAP* rendszerek célja a felhasználói lekérdezések széles skálájának minél gyorsabb megválaszolása, ezért a klasszikus adattárházak esetében az adatokat sokszor denormalizált formában, a redundanciát tudatosan használva tárolják. Az adattárházakban a válaszidő csökkentése érdekében használnak a gyakran használt lekérdezéseknek megfelelő dimenziók mentén előre kiszámolt értékeket, aggregátumokat is.

Az *OLAP* elvek megvalósítására több modellt is kidolgoztak az idők során. A *MOLAP* (*multidimensional OLAP*) modell áll implementáció szempontjából a legközelebb az adatkocka koncepciójához, mivel az egy speciális célstruktúrában többdimenziós tömbök segítségével tárolja a mért adatokat. A *ROLAP* (*relational OLAP*) megoldások esetében a többdimenziós modellt egy relációs adatbázisban valósítják meg, egy speciális sémában, mint amilyen a csillag- vagy a hópehelyséma. A fő entitást (tárgyat, témát) azonosító, azt megkülönböztető, illetve a numerikus tényadatokat az úgynevezett ténytábla (*fact table*) tartalmazza, ami a séma központi eleme. A ténytáblában az előzőek mellett idegen kulcsok is helyet kapnak, melyek olyan attribútumokra mutatnak, melyeken az entitások osztozhatnak, illetve melyek alapján a tényadatokat csoportosítani lehet. Ezeket az attribútumokat az úgynevezett dimenziótáblákban (*dimension table*) tárolják. A hópehelyséma esetében a dimenziótáblákat normalizálják, a redundancia csökkentése érdekében (Inmon, 2005; Kimball és Ross, 2013).

A relációs adatmodellre épülő *ROLAP* architektúrák előnye, hogy jobban skálázhatóak, ugyanakkor a bonyolultabb lekérdezések végrehajtásához sokszor számos, költséges illesztési (*join*) művelet használatára van szükség. A gyakorlatban előfordul a sémák denormalizálása és a redundancia tudatos alkalmazása, például optimalizációs szempontok alapján egyes, gyakran használt dimenzionális adatok is a ténytáblában kaphatnak helyet, a „drága" illesztési műveletek elkerülése érdekében. A *MOLAP* modell esetében gyakran említik hátrányként, hogy szükséges az aggregátumok előkalkulálása, illetve amennyiben a tárolt adatmátrix ritka (*sparse*), úgy gyakran többszintű adatreprezentációra és fejlett tömörítési algoritmusok

használatára van szükség a hatékony helykihasználás érdekében (Han *et al.*, 2011). A tervezett munkaerőpiaci adattárház esetében a mért, numerikus adatok hiánya miatt (a fizetés, fizetési kategória az egyetlen numerikus adat, amely egyes hirdetések esetében rendelkezésre állhat) nem indokolt egy *MOLAP* rendszer használata. A zömében szöveges (*string*) típusú adatot, a jobb skálázhatóság, illetve a könnyebb átláthatóság és modellezés érdekében egy denormalizált relációs adatmodellre épülő rendszerben gyűjteném.

A klasszikus adattárházak vertikálisan, azaz egy adott szervergép határain belül, jól skálázhatók (*scale up*), egyszerűen lehet például szükség esetén egy új háttértárat, vagy memóriaegységet hozzáadni a rendszerhez. Azonban bizonyos adatmennyiség fölött a vertikális skálázás nem gazdaságos, vagy akár megvalósíthatatlan, így természetesen adódik a terhelés horizontális, olcsóbb számítógépek hálózatba kötött klasztere közötti szétosztása (*scale out*) iránti igény. Egy elosztott architektúrára építő adattárház a relációs lekérdezéseket a hálózat egyes tagjain párhuzamosan futtatja, és a válaszokat is párhuzamosan dolgozza fel (Inmon és Linstedt, 2014), azonban a relációs adatbázisok horizontális kiterjesztése, az adatok elosztása az adatmodell sajátosságai miatt nem triviális. A probléma megoldására tett kísérletek során létrehozott, vagy újra felfedezett megoldásokból nőtte ki magát a *NoSQL* mozgalom.

3.1.2 NoSQL adatbázisok

Az első *NoSQL* megoldásokat tehát elsősorban a horizontális skálázás iránti igény hívta életre, melyet az ezredforduló relációs adatbázis-kezelő rendszerei nem tudtak megfelelően támogatni. Az ekkor létrehozott, új koncepciók alapján működő alkalmazások támogatják az adatok és a feldolgozás több olcsó számítógépből álló klaszteren történő hatékony, párhuzamos szétosztását (Presser, 2017). Ezt a modellt sokszor masszívan párhuzamos (*massively parallel architecture, MPP*) architektúraként hivatkozzák. Inmon és Linstedt (2014) alapján az *MPP* architektúra a semmit nem megosztani (*share nothing*) elvére épül, azaz a hálózat összes tagja (*node, shard*) csak a saját erőforrásaival gazdálkodik, és csak azok felett rendelkezik kontrollal. A feldolgozás az egyes nodeokon teljesen független a hálózat többi tagjától. A koordinációt gyakran egy dedikált központi egység (*master*) irányítja, ami felügyeleti funkciókat szintén ellát.

A párhuzamosíthatóság követelménye miatt az új típusú adatbázisok, relációk helyett új módokon, például aggregátumok vagy oszlopok alapján szervezik az adatokat. Az aggregátum tulajdonképpen az objektumorientált paradigmákból jól ismert objektum adatbázisszintű reprezentációja, azaz adatoknak olyan atomi csoportja, melyeket egy egységként kezelnek, általában együtt mozgatnak, változtatnak. Jellemzően a relációs adatbázisok tranzakcióit is ezek mentén az adatkörök mentén szervezik, a konzisztencia biztosítása érdekében. Több, a *NoSQL* családba sorolt adatbázis-kezelő típus fejlődött ki az ezredforduló első évtizedében, melyek közül a dokumentumtárak, a kulcs-értékpár és az oszlopalapú rendszerek is az aggregátumorientált koncepciót követik (Sadalage és Fowler, 2012). A legfőbb *NoSQL* adatbázis típusok között tartják számon még a gráf adatbázisokat és néhány keresésre, illetve idősoros adatok tárolására optimalizált megoldást is (Tudorica és Bucur, 2011).

Az új paradigma közelítette egymáshoz az objektumoknak – az alkalmazások által – a memóriában tárolt reprezentációját az adatbázisban használt ábrázolásához. Ebben az új, webalkalmazásokra fókuszáló időszakban, a fejlesztőknek legtöbbször nem volt szükségük arra, hogy az adatokat az SQL által biztosított rugalmas módon, tetszőleges kompozícióban kérdezzék le (Seeger, 2009). Az adatbázist egyszerű perzisztenciarétegnek tekintették az objektumok számára, és üdvözölték az új megoldásokat, melyek megkímélték őket az objektum-relációs konverzió „nyűgjétől". A legegyszerűbb kulcs-értékpár- és dokumentumtárak a memóriaobjektumok egyszerű szöveges, jellemzően *JSON* formátumú reprezentációinak tárolásához nyújtanak támogatást. Ezek a megoldások általában jól skálázhatók, és nem kényszerítik az adatokat egy előzetesen meghatározott sémába. Természetesen nem csak *JSON* objektumok tárolhatók ezekben a *NoSQL* adatbázis típusokban, számos adattípust, multimédia objektumokat stb. támogatnak. Például a 4.1.2 alfejezetben bemutatásra kerülő STUDIO ontológia modellje (10. ábra) is a *Tokyo Cabinet*[15] nevű kulcs-értékpár adatbázisban lett implementálva (Vas *et al.*, 2009).

A kulcs-értéktárak előnyei között tartják számon, hogy a kulcsok mentén az adatok rendkívül gyorsan elérhetőek, és rendkívül rugalmasak a tárolható értékek típusának tekintetében. A dokumentumtárak további hozzáadott értéke, hogy a dokumentumok különböző attribútumok szerinti indexelésére és lekérdezésére is támogatást adnak.

[15] https://dbmx.net/tokyocabinet/

Az oszlopalapú adatbázisokban az adatokat nem rekordok szerint, hanem attribútumok alapján csoportosítva tárolják. „Ezzel a szervezéssel a kevés oszlopot és sok sort érintő elemzések hatékonyabban elvégezhetők, ezért előszeretettel alkalmazzák analitikus adatbázisokban, adattárházakban” (Gajdos, 2019, p. 289). Ez a tárolási mód azért hatékonyabb az ilyen típusú lekérdezéseknél, mert a soralapú tárolással ellentétben, ahol az adatbázis-kezelő rendszernek balról jobbra az összes rekord minden attribútumát be kell olvasni ahhoz, hogy a kívánt értékeket megtalálja, az oszlopalapú rendszereknek csak a kívánt attribútumokat tartalmazó blokkokat kell átvizsgálnia. Így, főleg sok oszlop tárolása esetén, a lekérdezések válaszideje jelentősen csökkenthető. Ezzel a szervezési megoldással továbbá az adatok hatékonyan eloszthatók az adatbázis-klaszter elemei között, illetve az aggregáció is gyorsabban, kevesebb I/O művelettel megvalósítható (Nayak *et al.*, 2013).

Az idősoros adatokra optimalizált adatbázisokat (*time series database, TSDB*) egy periódus alatt, szabályos (esemény) vagy szabálytalan (állapotváltozás) időközönként beérkezett, egymást követő, időbélyeggel rendelkező adatpontok (jellemzően mérési eredmények) rögzítésére tervezték. Ilyenek például a szenzor- illetve a szerver és egyéb rendszerek terhelését és teljesítményét rögzítő logadatok stb., ezért például termelésirányításban, üzemeltetésben és *IOT* rendszerekkel kapcsolatban gyakran használják ezeket a megoldásokat. „A *TSDB* megoldásokat a változások időbeli nyomon követésére és elemzésük megkönnyítésére optimalizálták” (Naqvi *et al.*, 2017, p. 4). Az idősor adatbázisok esetében az elsődleges prioritás az írási műveletek végrehajtása, azaz az adatok mentése, még annak árán is, ha egyes olvasási műveleteket éppen nem fognak tudni kiszolgálni (Pelkonen *et al.*, 2015).

3.1.3 A CAP-tétel

A CAP-tétel néven azt a jó közelítésnek számító megfigyelést hívják, mely arra hívja fel a figyelmet, hogy a konzisztencia (*consistency*), a rendelkezésre állás (*availability*) és a partíció tolerancia (*partition tolerance*) tulajdonságai közül egy elosztott adatbázisrendszer esetében legfeljebb kettő garantálható egyidejűleg (Fox és Brewer, 1999). Fox és Brewer erős konzisztencia alatt a tranzakciók kezelésénél értelmezett *ACID* konzisztenciát értik, azonban Gajdos (2019) definíciója alapján ez egy hibás értelmezés. Az ő megfogalmazásában az *ACID* konzisztencia azt jelenti, hogy „csak a sikeresen (teljes egészében) lefutott tranzakcióknak van hatása az adatbázis tartalmára”, míg egy elosztott rendszert akkor nevezhetünk konzisztensnek, ha

„bármely időpillanatban egy adategység értékét bármely csomóponttól lekérdezve ugyanazt az értéket kapjuk” (Gajdos, 2019, p. 163 és 278). Egy elosztott rendszert továbbá akkor nevezünk magas rendelkezésre állásúnak, ha redundánsan és replikálva tárolja az adatokat, és a hálózat elérhető elemeihez (*nonfailing node*) beérkező kéréseket mindig megválaszolja (Gilbert és Lynch, 2002, p. 53). Továbbá a partíció tolerancia (reziliencia) azt jelenti, hogy az adatbázis egy adott lekérdezésre a hálózat egyes csomópontjai (adatbázisreplikák) közötti kapcsolat megszakadása esetén is helyes választ tud adni (Fox és Brewer, 1999; Gajdos, 2019).

Jelen munka szempontjából az egyes adatbázismegoldások között választás során fontos a CAP-tétel figyelembe vétele, amiről bővebben az 5.3.1 alfejezetben lesz szó.

3.2 Ontológia

Az ontológia eredetileg filozófiai fogalom, mely tudományterületen a létezés szisztematikus magyarázatára tesz kísérletet (Corcho *et al.*, 2003). Az eredeti filozófiai kontextusban az ontológia a tudományelmélet „létezőt, a létet és alapjait, tulajdonságait vizsgáló ága, a hagyományos értelemben vett metafizika egyik része, egészen az ókori görög filozófiáig nyúlik vissza, bár maga a kifejezés csak a XVII. század elején jelent meg” (Vas, 2007, p. 9). Jellemzően korunkban az ontológia kifejezés leginkább az információrendszerek területén betöltött szerepe alapján ismert, a tudásmenedzsment terminológia része.

Az információrendszerek kontextusában az ontológia egy adott szakterületen felhalmozott tudás egyszerű és hatékony megosztását lehetővé tevő reprezentációs technika és eszköz (Neusch, 2014). „Az ontológia a létező minden formájának és módjának logikáját szisztematikusan, formálisan és axiomatikusan képezi le” (Cocchiarella, 1991). Ellentétben egy egyszerű szószedettel vagy egy taxonómiával, ahol a szakterület kifejezései általában hierarchiába vannak rendezve, az ontológiák nem csak az objektumokat, hanem a köztük lévő kapcsolatokat, a kapcsolatok irányát, a rájuk vonatkozó megkötéseket, illetve az egyes fogalmak tulajdonságait is tartalmazzák. Egy ontológia segítségével egy tudományterület, szakma, stb. komplex tudástérképe, modellje építhető fel. Tehát egy egyszerű szószedettel ellentétben, ami egy kontextus és definiált kapcsolatok nélküli szólista, egy ontológia a kontextust is tartalmazza (Burleson, 2016; Sneftel, 2013).

Praktikus szempontból az ontológia egy adatstruktúra, egy adatmodellező eszköz, amit gyakran használnak például szemantikus web alkalmazásokban. Mivel az ontológiákra következtető motorok (*inference engine*) építhetők, azaz logikai szabályok alapján az elemekről és azok kapcsolatairól következtetések vonhatók le, ezért az ontológiákat gyakran felhasználják mesterséges intelligencia alkalmazások és szakértői rendszerek komponenseként.

Az ontológia Gruber definíciója alapján „a fogalomalkotás explicit specifikációja” (“*explicit specification of a conceptualization*”) (Gruber, 1993; Sántáné-Tóth, 2001, p. 2). Az ontológia egy közös ábrázolási alap *(representational primitives*), melynek segítségével egy szakterületen felhalmozott tudás, vagy például egy diskurzus formalizálható, részletesen leírható (*specification of a conceptualization*) és tárolható, oly módon, hogy az mind emberek, mind gépek által feldolgozható (Gruber, 2009). Ezek alapján elmondható, hogy egy jól használható ontológia formális struktúrát használ, mely követi a modellezés legjobb gyakorlatait és a terület sztenderdjeit, és tartalma közös megegyezésen alapul. A „*representational primitives*” kifejezést Gruber (2009) adattípus értelemben is használja, egy modellezési eszközkészletet értve alatta, azokat az osztályokat, attribútumokat és kapcsolatokat, melyek a metamodell részét képezik. A „*conceptualization*” kifejezés alatt Guarino és Giaretta (1995, p. 6) egy olyan tudatosan létrehozott szemantikus konstrukciót ért, amely a valóság egy szeletének struktúráját korlátozó (irányító) implicit szabályokat kódolja (fogja rendszerbe).

Az ontológia tehát referenciapontként szolgálhat egy adott szakterület számára. A szakirodalomban Gruber definíciója alapként szolgál más szerzők számára is, így lényegében a különböző egyéb meghatározások csak kiegészítéseket tartalmaznak, általában az ontológiák következő tulajdonságait hangsúlyozva (Studer *et al.*, 1998):

- formális (formalizált) – számítógéppel olvasható és feldolgozható, ami mellett természetesen nem árt, ha emberek technológia felhasználása nélkül is megértik,
- az adott szakterület specialistáinak (*domain expert*) közös megegyezése alapján jön létre,
- modellezési keretet ad (struktúra, metamodell),

- a tartalmat az adott szakterület fogalmi váza adja.

A szakterület, „domain, vagy értelmezési tartomány egy specifikus tárgyterület vagy tudásterület, mint pl. orvostudomány stb.” (Gottdank, 2006, p. 78). A szakterületi tudás (*domain knowledge*) kifejezés pedig az alkalmazási terület tudásobjektumaira és a területet leíró statikus információkra utal (Corcho és Gómez-Pérez, 2000). Az ontológiaépítés nem korlátozódik azonban valamilyen szakterületre, ontológia építhető akár egy könyvből, előadásból vagy diskurzusból is.

Guarino és Giaretta (1995) az ontológiát az adott fogalomalkotás részleges leírásaként fogalmazza meg, azaz a fogalomalkotásra, mint emberek egy csoportja által a világról alkotott ideára tekint (Corcho *et al.*, 2003), ami megszorítás arra vonatkozóan, hogy az ontológia közös megegyezésen alapul, de azt is jelzi, hogy a „közös” szó emberek egy szűkebb körének közösségét is jelölheti.

Az ontológia tehát a világ, illetve annak egy részének tudásáról készült modell, absztrakció, melyben a modellező meghatározza a területre leginkább jellemző ontológia osztályokat (*class, set*), azok struktúráját, hierarchiáját, az osztályok közötti kapcsolatokat (*relations*), és az azokra vonatkozó axiómákat, szabályokat is (Neusch, 2014). Az ontológiaosztályoknak tulajdonságai (*attributes, properties*) is lehetnek (Gruber, 2009). Ez a metamodell mintegy keretként szolgál az ontológia tartalommal való feltöltéséhez. Adatmodellezési szempontból ez nagyon hasonló például az adatbázis séma koncepciójához a relációs adatmodellezés esetében, illetve az osztályok struktúrájához az objektumorientált tervezés esetében (Burleson, 2016).

A szakterület szakértői és a felhasználók számára az ontológia értékét azonban a tartalom jelenti, amivel a felépített metamodellt, a struktúrát megtöltik úgy, hogy rögzítik az adott *domain* alapvető fogalmait, besorolják azokat az osztálystruktúrába, és megadják a közöttük lévő kapcsolatokat (Neusch, 2014). A modellbe felvett tudáselemek tehát az ontológiaosztályok instanciái, az osztálystruktúra pedig a tervrajz, dizájn sablon. A relációs adatmodellezésben, egy rekord, az objektumorientált programozás esetében pedig egy objektum (futásidejű koncepció) hasonlítható például egy instanciához egy ontológiában. A modellezési eszköztár tehát a struktúrán (osztályok, kapcsolatok, tulajdonságok) kívül magában foglalja a tartalmat (példányok) is. „Az ontológiák tehát az értelmezési tartomány alapvető fogalmainak számítógép által használható definícióit és a köztük lévő kapcsolatokat tartalmazzák"

(Gottdank, 2006), más megfogalmazásban az ontológia megadja egy adott témakör szókincsét, terminológiáját alkotó alapfogalmakat (tartalom) és a köztük lévő kapcsolatokat, valamint a kifejezések kapcsolatok segítségével történő ötvözésének szabályait (értelmezési keret, hierarchia) (Neches *et al.*, 1991).

A metamodellhez általában kapcsolódnak a felépítési logikára vonatkozó szabályok is, melyek kidolgozása szintén jelentős mértékű együttműködést követel meg a szakértőktől, mivel nagyban befolyásolják, hogy a modellezés során létrehozott ontológia érvényes és értelmes lesz-e. Ilyen szabályok például a következők:

- Mely ontológiaosztályok között létesíthető kapcsolat és
- az adott ontológiaosztályba tartozó elemek között milyen kapcsolatok (relációk) megengedettek?
- Milyen kötelező, vagy opcionális attribútumai lehetnek egy egyednek stb.?

Ezek mellett az explicit módon, a metamodellben megfogalmazott szabályok mellett olyan implicit, a modellben nem rögzített megkötések is lehetnek, mint:

- Névkonvenció, azaz hogyan kell megnevezni az egyes példányokat?
- Mekkora részt reprezentálhat egy ontológiaelem a teljes szakterületből, azaz milyen szemcsézettségen (*granularity*) kell a tudáselemeket ábrázolni? (Szintén kapcsolódik a névkonvenció kérdésköréhez, például minősítő jelzős kapcsolatokkal lehet specifikálni az egyes tudáselemeket, például osztály, ontológiaosztály, java osztály stb.)

Ezekre és a hasonló kérdésekre nem feltétlenül van egzakt válasz, illetve azt jelentősen meghatározza a modellezés célja. Megválaszolásuk jelentős megfontolást, együttműködést és szakértői döntést igényel.

„Az ontológiákat általában egy logikán alapuló, az adatmodelltől és implementációs stratégiától független nyelven fogalmazzák meg úgy, hogy részletes, pontos, egyértelmű, megbízható és értelmes megkülönböztetéseket tehessünk osztályok, tulajdonságok és viszonyok között” (Heflin, 2005, p. n.a.). Ez a függetlenség az alsóbb szintű ábrázolási módoktól nagyfokú rugalmasságot biztosít és lehetővé teszi ontológiák alkalmazását heterogén adatforrások integrálásában, illetve alkalmazások együttműködésének biztosításában is (Gruber, 2009).

3.2.1 Ontológiák típusai

A szakirodalomban az alkalmazás kötöttségétől, az újrahasznosíthatóságtól, a fogalmi rendszer típusától, illetve a szemcsézettségétől függően általában a következő ontológiatípusokat különböztetik meg (Gómez-Pérez, 1999).

- Általános ontológia – általános, absztrakt dolgokat, generalizációkat, eseményeket, oksági viszonyokat ábrázol. Mivel nagyon primer terminológiát használ, ezért szakterületek között egyszerűen újrahasznosítható.
- Szakterületi ontológia – az adott szakterület specifikus tudását modellezi, egyszerűen csak az adott területen belül használható fel újra.
- Folyamat ontológia – az adott folyamat egyes elemeit, tevékenységeit és az azokhoz kapcsolódó egyéb erőforrásokat modellezi. Nem, illetve nehezen újrahasznosítható.
- Alkalmazási ontológia – egy specifikus alkalmazáshoz kapcsolódó ismereteket modellez. Általában nem újrahasznosítható az adott alkalmazás keretein kívül.

Egy másik definíció alapján az általános ontológiát gyakran hivatkozzák felsőbb szintű (*upper, top level, basic formal ontology*) ontológiának, mert olyan alap szókészletet képez le, amely könnyen megosztható szakterületi ontológiák széles skálája között (Neusch, 2014), azaz a generalizáció és az absztrakció szintje maximális (Spear *et al.*, 2016).

3.2.2 Fogalmak rendszere

Az ontológia tehát egy tudástérkép, fogalmak (*concept*) szemantikusan felépített rendszere. Egy ontológiában a legkisebb entitás egy fogalom. Egy fogalom lehet absztrakt vagy konkrét, egyszerű vagy összetett, valós vagy vélt (kitalált), tehát bármi, ami az univerzum egy objektuma, és így le tudjuk írni (Gomez-Perez és Corcho, 2002). Mivel egy ontológiafogalom ebből következően bármi lehet, „amit le lehet írni, így fogalom lehet egy feladatleírás, szerep, akció, stratégia vagy érvelési folyamat stb.” (Corcho és Gómez-Pérez, 2000).

Angolul a *Concept* kifejezést gyakran használják az osztályok (metaszint) és az instanciák megnevezésére is, ugyanúgy, ahogy az *Object* kifejezést is. Jelen dolgozatban, összhangban az objektumorientált paradigmákban használt terminussal, amikor a metaszintre utalok, az osztály vagy a kategória kifejezést fogom használni,

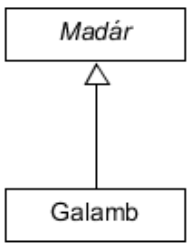
míg a fogalom és az objektum kifejezések az instancia szinonimájaként jelenhetnek meg.

3.2.3 Fogalmi kapcsolatok modellezése

A minél részletgazdagabb modellezési lehetőségek biztosítása érdekében számos kapcsolattípust különböztetünk meg egymástól, annak megfelelően, ahogy a modellezett objektumok között, azok egymáshoz való viszonyának megfelelően, a valós világban is számtalan típusú, egymástól különböző kapcsolattípus lehet. Az ontológiakapcsolatok az egyes fogalmak (instanciák) közötti logikai kapcsolatokat és függőségeket testesítik meg (Gulla és Brasethvik, 2008). Metaszinten a tudásreprezentáció területe osztozik a kapcsolattípus-definíciókon az objektumorientált paradigmákkal. Gulla és Brasethvik (2008) alapján a következő típusokat különböztetjük meg.

3.2.3.1 Hierarchikus reláció

A hierarchikus reláció (*hyperonymy*) taxonómikus, vertikális, *is_a* típusú reláció (Gulla és Brasethvik, 2008), ami hierarchikus alá-fölérendeltséget fejez ki, azaz azt, hogy egy alosztály (*hyponym*), illetve az osztály egy bizonyos egyede, egy adott szuperosztályból (*hypernym*) származik. Gyakran nevezik ezt a relációtípust általánosításnak is. Például, ha vesszük a *Madár* absztrakt osztályt és a *Galamb* osztályt, akkor ebben az esetben elmondható, hogy a *Galamb Madár* típusú, azaz *Galamb is_a Madár*. Ennek UML reprezentációját mutatja a 2. ábra.



2. ábra: Alárendelő reláció

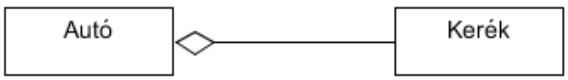
Az objektumorientált programozási nyelvekben, például a JAVA nyelvben egy osztálynak csak egy szülőosztálya lehet. Ez a megkötés követi a hagyományos taxonómiák, mint például a (biológiai) rendszertan szabályait. Azonban az ontológiák esetében az ilyen jellegű megszorítás legtöbbször nagyon megkötné a modellezést végző kezét, így az ontológiákkal kapcsolatban elterjedt a polihierarchikus relációk használata. A polihierarchia engedélyezése azt jelenti, hogy bizonyos elemek esetében

megengedjük, hogy azok akár több kategóriába is tartozzanak egyszerre (Morville és Rosenfeld, 2006).

3.2.3.2 Rész-egész viszonyt kifejező relációk

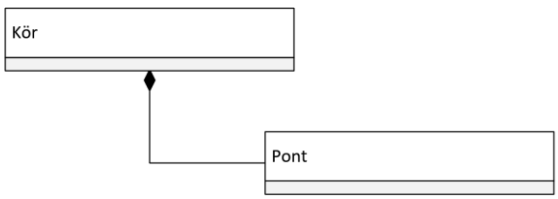
Ezek a relációk nem fejeznek ki hierarchikus alá-fölérendeltséget, sokkal inkább azt modellezik, hogy valamely fogalom számos másik fogalom összességéből áll, azaz a tartalmazás, a rész-egész viszony ábrázolására szolgálnak. Az előzőeket úgy is mondhatjuk, hogy valamely fogalom része egy másik fogalomnak (*part_of*), vagy valamely fogalom rendelkezik egy másik fogalommal (*has_a*). A része (építőeleme, *part of*) kapcsolattípust partonómiai (*partonomy*) kapcsolattípusnak is nevezzük (Sántáné-Tóth *et al.*, 2007, p. 356). Gulla és Brasethvik (2008) alapján ebbe a csoportba tartozik az UML szakkifejezésekkel kompozíciónak és aggregációnak nevezett kapcsolattípusok.

- Az **aggregáció** fejezi ki, hogy egy fogalom valamely nagyobb egész részét képezi (*is_part_of*). Az aggregáció tipikus példájaként szokták említeni az autó és alkatrészei például kormány, kerék közötti kapcsolatot (Fowler, 2003). Az aggregáció UML ábrázolását mutatja a 3. ábra.



3. ábra: Aggregáció

- A **kompozíció** azt fejezi ki, hogy „az egyik osztály objektumai a másik osztály objektumait *fizikailag tartalmazzák*” (*contains*) (Sike és Varga, 2003, p. 124). A kompozícióval szembeni megkötés, hogy a komponens egy instanciáját csak egy aggregációs objektum tartalmazhatja (Fowler, 2003; Sike és Varga, 2003).

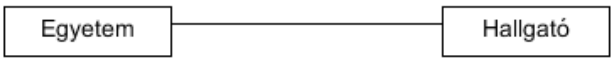


4. ábra: Kompozíció (Fowler, 2003)

3.2.3.3 Asszociáció

Az asszociációs kapcsolat azt fejezi ki, hogy az egyik fogalom kapcsolatban áll egy másik fogalommal (Gulla és Brasethvik, 2008). Az asszociáció egy nagyon általános

relációtípus, mellyel egyszerű társítást fejezünk ki objektumosztályok között (Sike és Varga, 2003).



5. ábra: Asszociáció (Sike és Varga, 2003)

3.2.3.4 Egyezőség, azonosság

Az ontológia fogalmához kapcsolódóan fontos lehet két fogalom egyezőségének jelölése. Szemantikai szinten ez arra utal, hogy két kifejezés ugyanazt a fogalmat azonosítja. Az EuroVoc (*Multilingual Thesaurus of the European Union*) megfogalmazásában az egyezőségi reláció (*equivalence relationship*) egy fogalom preferált és nem preferált megnevezései között áll fent, például „idős ember" és „öreg ember" kifejezések (EuroVoc, n.d.). Ilyen módon az azonossági relációnak a jellemző tulajdonsága, hogy lehetővé teszi a kifejezések helyettesítését logikai formulákban (Guarino és Welty, 2000).

Jelen tézishez kapcsolódóan, praktikusan ez a reláció az ESCO objektumok esetében jelenik meg. Minden ESCO objektumnak van egy elsődleges címkéje (*preferredLabel*) és opcionálisan egy vagy több alternatív címkéje (*alternativeLabel*). Például a „*laboratory techniques*" elsődleges címkével ellátott tudáselem alternatív címkéi többek között a „*techniques used in laboratory*", „*lab techniques*" vagy éppen a „*gas chromatography*" (European Commission, 2018).

Természetesnyelv-feldolgozás esetében, ha két kifejezés ugyanazt a fogalmat jelöli, akkor szinonimákról beszélünk. Az egyes szinonimák által jelölt fogalmak egyértelmű beazonosításával a névelem normalizálás területe foglalkozik (3.3.4 alfejezet).

3.2.4 Ontológiák felhasználása

Az ontológiák leggyakoribb felhasználási területe a tudásmenedzsment. Sokszor használják ezt a modellezési módszert szakterületi tudás leképezésére, melynek szemléltetésére jó példa a STUDIO rendszer (4.1.2 alfejezet). Egy-egy szakterület így felépített „tudástérképe" e-learning rendszerekben sokszor a tudás formalizálásának alapvető modulja, míg a kapcsolódó, rá épülő algoritmusokon keresztül a tudáshiányok feltárásának, és a tanításnak is fontos eszköze lehet. A STUDIO-ban például az egyes ontológiaelemekhez attribútumként kérdések és tananyag kapcsolható, és a rendszer ezek alapján képes a hallgatók tudáshiányainak pontos meghatározására, és a

hiányosságok pótlását elősegítendő célzott „tanulási utak” ajánlására (Gábor *et al.*, 2016; Weber és Vas, 2015).

A szakterület modellezésére gyakran ontológiát használnak továbbá szakértői-, döntéstámogató-, illetve mesterséges intelligenciára épülő rendszerekben és szemantikus web alkalmazásokban is. Az oktatáson és az egyszerű információkategorizáláson kívül széleskörűen használnak ontológiákat összetett dokumentumok klasszifikációjára is, például információkinyerő rendszerekben vagy elektronikus könyvtárakban, így azok könnyebben kereshetővé, számítógéppel egyszerűbben feldolgozhatóvá válhatnak. Mivel az ontológiák leképezik, modellezik a valós világot, annak osztályait, kapcsolatait és egyedeit, újrafelhasználható alapot jelentek a rendszerfejlesztés, adatbázis-tervezés stb. számára is (Neusch, 2014).

Ontológiák szemantikus web alkalmazásokban való felhasználásának klasszikus példája a keresés. Az egyszerű szövegegyezésen alapuló internetes keresések pontossága, például az egyes objektumok, entitások, erőforrások összekapcsolásának segítségével, nagymértékben javítható. Ilyen – tudástárra alapuló – kereső létrehozására irányuló első kezdeményezés volt a Wolfram Alpha, melynek célja, hogy egzakt válaszokat adjon a természetes nyelven megfogalmazott kérdésekre. Ilyen továbbá a Google Knowledge Graph is, mely évek óta része a Google keresőmotorjának. Régen, ha például rákerestünk arra, hogy „Mátyás király felesége” vagy „Mona Lisa festője”, számos weboldalra mutató hivatkozás listáját kaptuk, melyekből jobb esetben megtudhattuk a kívánt adatot. Ma ugyanezekre a lekérdezésekre a kereső, a háttérben található tudásgráf segítségével, pontos választ tud adni anélkül, hogy el kellene navigálnunk az oldalról.

Ontológia szemantikus web alkalmazásokban való felhasználásának egy másik példáját vázolja föl Heflin 2004-ben íródott munkájában. Egy akkoriban még csak vízióként létező szabadidő-tervező szemantikus web alkalmazást ismertet, mely a felhasználó preferenciái, például kedvelt filmek, ételek, könyvek stb. alapján meg tudja tervezi annak ideális estéjét. Kiválasztja a filmet, megvásárolja a mozijegyet. Az étterem jellege és az online foglalási adatok alapján még a konkrét, elérhető vendéglőre is javaslatot tud tenni, vagy akár asztalt is foglalni (Heflin, 2005). Manapság az ilyen jellegű intelligens alkalmazások, többek között az ontológiáknak is köszönhetően, egyre inkábbá a mindennapi valóság részévé válnak.

Szövegbányászati alkalmazásokat is kiegészíthetnek ontológiai komponenssel, annak érdekében, hogy a természetesnyelv-feldolgozást megtámogassák az adott szakterületet reprezentáló fogalmi rendszer nyújtotta többlet szemantikával (Neusch, 2014). Jelen tézisben is felhasználok ontológiákat ahhoz a szövegbányászati feladathoz, hogy az álláshirdetések szövegében beazonosítsam a lehetséges kompetenciákat, illetve az egyes hirdetésekhez foglalkozást társítsak.

3.2.5 Következtetés ontológiákon

Ahogy azt az előző alfejezetben már említettem, ontológiákat széles körűen felhasználnak egyes szemantikus web és mesterséges intelligencia alkalmazásokban, mert azokra építve automatikus következtetések vonhatók le az adott, modellezett szakterületre vonatkozóan. Az ontológia, definíciója alapján, nem csak azokat a kifejezéseket tartalmazza, melyeket explicit módon definiál, hanem azt a tudást is, amire ezek alapján következtetni lehet (Corcho *et al.*, 2003; Neches *et al.*, 1991). Az emberi aggyal ellentétben, amely képes az absztrakcióra, elvont fogalmak interpretációjára, és ez alapján objektumok közötti kapcsolatok felfedezésére, egy számítógépnek szüksége van kontextusra, kiegészítő információkra (például metaadatok) az adat szinten reprezentált fogalmak értelmezéséhez. Vegyünk példának egy egyszerű dolgot, mint egy szék. Az emberi agy képes a szék koncepcióját absztrahálni, és úgy értelmezni ezt az objektumot, mint valamit, ami alkalmas arra, hogy üljünk rajta. Ez alapján a koncepció alapján az ember – érzékei által továbbított információk (például látás) segítségével – képes beazonosítani egyéb objektumokat is, melyek alkalmasak lehetnek arra az adott célra, hogy üljenek rajtuk (például lépcső, illetve olyan egyéb szilárd tárgyak, melyek magassága és felülete megfelelő). A szék, a lépcső és azon egyéb objektumok közötti kapcsolat, melyek a célnak megfelelnek, természetesen alakul az emberi agy számára az ülés kontextusában. Azonban egy számítógép nem képes ilyen jellegű absztrakcióra, például az ülőalkalmatosságok közötti kapcsolat automatikus kikövetkeztetésére, így egy algoritmusnak szüksége van a kontextus explicit leírására, hogy egyes objektumok közötti kapcsolatokat felismerjen.

Itt lépnek be a képbe az ontológiák mint a kontextusinformációk forrásai egy adott diskurzus esetében, és így ontológiák alkalmazásának segítségével az egyes mesterséges intelligencia alkalmazások képesekké válnak az adott diskurzus hatékonyabb értelmezésére (Deshpande és Kumar, 2018). Az ontológiákon való

érvelés, illetve következtetés (*inferencing, reasoning*) lehetőségének segítségével a számítógépek számára lehetővé válik új összefüggések feltárása, új információk kinyerése egy adott szakterület adataiból, azokon kívül, melyeket a modell explicit módon tartalmaz, és mindez automatikusan végrehajtható. Tehát egy számítógépprogram, például egy szakértői rendszer, amely rendelkezik következtetések levonására alkalmas modullal, képes új konklúziókra vagy akár önálló döntéshozatalra jutni (DuCharme, 2013). Burleson (2016) felsorolja néhány előnyét annak, hogy az ontológiák segítségével képesek vagyunk következtetéseket levonni.

- Kapcsolatok automatikus feltárása, illetve levezetése. Például, amennyiben egy ontológiában explicit meg van adva a kapcsolat, hogy 'Galamb Máténak van egy Bogáncs nevű kutyája (*has_a*)' – és egy axiómában rögzítjük az adott relációtípus inverzét –, egy következtető motor le tudja vonni a konklúziót, hogy Bogáncs gazdája Galamb Máté.

- Automatikus klasszifikáció (*autoclassification*). Egy adott instancia valamilyen megkülönböztető tulajdonsága alapján a következtető motor automatikusan be tudja azonosítani azt az ontológiaosztályt, amelybe az adott egyed tartozik. Például ha egy magyar állampolgár személyi száma egyessel kezdődik, akkor ha létezik erre vonatkozó következtetési szabály egy rendszerben, az automatikusan be tudja sorolni az adott állampolgárt a Férfiak osztályába. Vagy egy másik példával élve, ha egy ontológiában személyekről tartunk nyilván adatokat, és azoknak lehet olyan tulajdonsága, hogy hangszeren játszik (*"playInstrument"*), mely tulajdonságon keresztül egy adott személyt összekötünk egy hangszerrel, azaz a tulajdonság beazonosít egy hangszert, amin az adott személy játszik, úgy ez alapján a tulajdonság alapján egy személy instancia automatikusan besorolható a "*Zenész*" osztályba. További feltétel, hogy a Zenész osztály is tartalmazza az adott tulajdonságot (DuCharme, 2013).

- Az ontológiákon való érvelés, illetve következtetés arra is használható, hogy összekapcsoljunk egymásnak megfelelő koncepciókat, melyek különböző, különálló adatforrásból, tudásbázisból vagy alkalmazásból stb. származhatnak. Ez nagy segítséget jelenthet például *linked data* alkalmazások építésénél, illetve lehetővé teszi alkalmazásokon átívelő, komplex szemantikus lekérdezések megfogalmazását.

Strukturális szempontból következtetések vonhatók le osztály és osztály közötti, osztály és osztálypéldány közötti, illetve egyed és egyed közötti kapcsolatok alapján (Neusch, 2014).

3.3 Szövegbányászat

A szövegbányászat (*text mining*) az adatbányászat alterülete, olyan módszereket és eszközöket összefogó tudományág, melyek segítségével strukturálatlan, szövegesen adott adatokból kinyerhetjük az elemzési cél szempontjából fontos információkat. Az adatbányászat és a szövegbányászat közötti fő különbséget Witten és munkatársai úgy fogalmazzák meg, hogy míg az első lényegében mintázatok keresése adatokban, addig a második esetében a különbség csak annyi, hogy ezek az adatok szövegesen adottak (Witten *et al.*, 2011). A „mintázatkeresés" azt jelenti, hogy olyan információkra vagyunk kíváncsiak, melyek explicit módon nem szerepelnek az elemzett szövegek összességében (Fajszi és Cser, 2004), amit a szövegbányászatban használt terminus technicusszal korpusznak is neveznek.

A szövegbányászati elemzés különlegessége tehát, hogy a bemeneti adatok rosszul strukturáltak és szövegesek, továbbá következtetéseinket olyan dokumentumokból próbáljuk levonni, melyeket előzőleg nem készítettek fel elemzésre, ami megnehezíti a szövegek automatizált analízisét. Sok esetben nem csak az input adatok, de az elemzési feladat is rosszul strukturált, sem a célállapotok, sem az oda vezető utak nem vázolhatók fel egyértelműen, egzakt formalizmusokkal, illetve sokszor a kapott eredményeket is értelmezni szükséges, így a következtetések során helyet kaphat a szakemberek szubjektivitása is. Legtöbbször azonban a problémák jellege miatt nincs lehetőség, illetve idő arra, hogy humán erőforrásokat használjunk fel a megoldás, illetve annak értelmezése során, például a lekérdezések nagy száma, vagy a kritikus válaszidő miatt (Neusch, 2014). Ezért Witten és munkatársai (2011) szerint a szövegbányászati megoldásoktól azt várjuk, hogy *megfelelően pontos*, nem szakértő felhasználó, vagy más számítógépes algoritmus számára is könnyen értelmezhető választ adjanak, mely alapját képezheti döntésnek vagy automatikus beavatkozásnak.

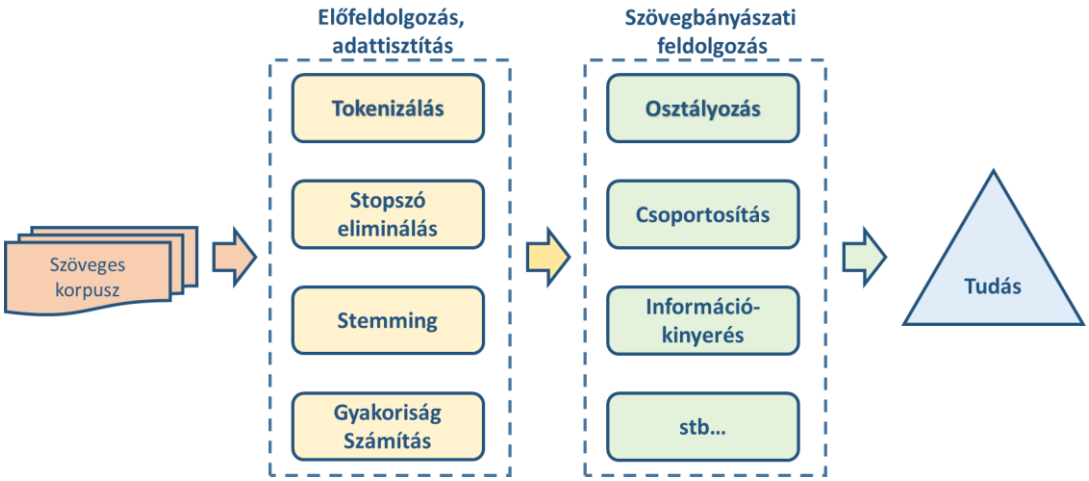
A szövegbányászat multidiszciplináris tudományág, mely nagymértékben támaszkodik az adatbányászat, a statisztika, a számítógépes nyelvészet, a nyelvtechnológia, a könyvtártudomány, az adatbázis rendszerek, a mesterséges intelligencia stb. területek eredményeire (Kő, 2013). A szakterület relevanciáját az

adja, hogy „az internet korának egyik jelentős trendje az elektronikus adatok rohamosan növekvő mennyisége, melyek nagy része szöveges” (Tikk *et al.*, 2007, p. 14). Ezek az adatok sokszor a szervezetek számára felbecsülhetetlen értékű információkat tartalmaznak például az ügyfelek preferenciáiról, a választók véleménypolaritásáról vagy éppen általánosságban arról, hogy milyen témák érdeklik az embereket (Evangelopoulos és Visinescu, 2012).

A szövegbányászati problémák közé sorolunk gyakran olyanokat is, – melyek megoldása bonyolultabb, komplex nyelvészeti ismeretek felhasználását is igényli – mint amilyen a már említett véleménypolaritás-vizsgálat (*opinion mining, sentiment analysis*), vagy szövegek automatikus fordítása stb. Azokat a technológiákat, melyek a nyelvi objektumok felismerésére, ezáltal a szöveg kvázi értelmezésére, szintaktikai és morfológiai elemzésére tesznek kísérletet, számítógépes nyelvészet („természetesnyelv-feldolgozás”, *Natural Language Processing, NLP*) összefoglaló néven is hivatkozzák. A két fogalom közötti határvonal rendkívül homályos, és gyakran, bizonyos problémák esetében akár szinonimaként is használják őket. Bár olyan megoldásokat is besorolnak az *NLP* ernyője alá, melyeket egyébként nem tartok a szövegbányászat körébe tartozónak (például chatbotok), jelen dolgozatban a természetesnyelv-feldolgozás alkalmazásaira, mint a szövegbányászat eszközkészletének részeire tekintek.

A szövegbányászat egy feldolgozási folyamatként is felfogható, amely több lépésből áll. Először általában a vizsgálandó szöveget az elemzéshez megfelelő formába alakítják át (előfeldolgozás), majd az így előállt részlegesen strukturált (*semi structured*) adathalmazon alkalmazzák az elemzés céljának megfelelő modellt. Szintén gyakori feladat a modellek parametrizálása. A megfelelő paraméterértékek megtalálása érdekében gyakran kísérleteket kell végezni, és az adatokat újra és újra átfuttatni a modelleken.

A szövegbányászat klasszikus, általános, folyamatszemléletű modelljét mutatja be a 6. ábra. Az előfeldolgozás során általában a tokenizálás, azaz a szöveg szegmentációja, a stopszavak – az elemzés szempontjából irreleváns vagy jelentéssel nem bíró tokenek – eltávolítása, és a szavak szótári alakra hozása történik meg. Utóbbi végrehajtható algoritmikusan (*stemming*) vagy szótár segítségével (*lemmatization*). Általában az egyes szövegbányászati algoritmusokat, megoldásokat az ilyen módon előkészített szövegen alkalmazzák.



6. ábra: A szövegbányászat általános modellje Fajszi és mtsai. (2010) és Gillani (2015) alapján

A következő alfejezetben részletesen leírom az előfeldolgozás lépéseit, azonban mindenképpen meg kell itt említeni, hogy a szöveg előkészítésére használt módszerek, ”komplexebb modellek, főleg *deep learning* esetében ronthatják a modell teljesítményét, mivel ezekkel az eljárásokkal információt veszítünk” (Thomas, 2019a). Azaz bizonyos esetekben jobb eredményeket lehet elérni akkor, ha a szöveg előfeldolgozását elhagyjuk és a korpuszt eredeti formájában használjuk. A véleményanalízis esetében például nagyon sokat számíthat bizonyos kifejezések negatív vagy pozitív töltésének megállapításakor a szavak környezete, a jelzők, a ragok stb. Tehát vannak olyan problémák, melyek esetében a szöveg hagyományos értelemben vett előfeldolgozása nem visz közelebb a megoldáshoz, míg mások esetében jobb eredményeket érhetünk el általa. Így minden konkrét felhasználási esetben mérlegelni kell alkalmazásának szükségességét.

3.3.1 A szöveg előkészítése

A szövegbányászati elemzések során feldolgozott szövegek változatos forrásokból származhatnak, és nagyon sokféle formában állhatnak rendelkezésre. A formának ez a változatossága megnyilvánulhat a feldolgozandó szöveg nyelvében, a karakterkódolásban vagy éppen a formátumban. Így a szövegbányászat folyamatának első lépése az, hogy a sokszor kaotikus módon rendelkezésre álló input adatokat beolvassuk az elemzést végző eszközünkbe, és egy sztenderdizáltabb, jobban kezelhető formába hozzuk (Neusch, 2014).

A gyakori szövegelőkészítési lépések közé tartozik a szöveg „uniformizálása”, azaz minden kis- vagy nagybetűssé alakítása, a tokenizálás vagy szegmentáció, azaz a

korpusz elemi nyelvi objektumokra (pl. szavak, írásjelek stb.) bontása, és a jelentést nem hordozó, a feldolgozás szempontjából irreleváns, úgynevezett stopszavak kiszűrése. Amikor a stopszavak kiszűrésére szükség van, akkor érdemes lehet az írásjeleket és vélhetően hibás karaktersorozatokat is eltávolítani.

A tokenizálás, az elemzés elemi objektumainak előállítása, legtöbbször pusztán azt jelenti, hogy a szöveget szavak szekvenciájaként ábrázoljuk. De ez egyáltalán nem triviális feladat. Például a magyar nyelvben a szótagszámlálás szabálya azt írja elő, hogy a hatnál több szótagból és a kettőnél több elemből álló összetételeket tagolnunk kell. Ekkor például az „adatbázis-kezelő" szóösszetétel esetében egyáltalán nem mindegy, hogy azt egy szerves egységként, vagy két külön objektumként kezelem-e (Tikk *et al.*, 2007), hiszen ennek alapján megváltozik a szavak jelentése, illetve a második esetben információt is veszítünk. A tokenizálás folyamán tehát sokféle kérdés és probléma merülhet fel, így az idők során ennek megfelelően számos megközelítést és algoritmust fejlesztettek ki. Azokban az esetekben, amikor nagy pontosság elérésére van szükség, a tokenizáláshoz gyakran szótárat is használnak. Ugyanez a helyzet az olyan nyelvek esetében – mint például a kínai –, melyekben nincsen a szavakat egyértelműen elválasztó, azaz a szöveget szegmentáló karakter (Guo, 1997).

Az adott probléma függvényében, az előfeldolgozás során sokszor kiszűrik az úgynevezett stopszavakat, melyek olyan speciális nyelvi elemek, amik önálló jelentéssel általában nem rendelkeznek, illetve eltávolításuk a szövegbányászati feladatok egy részének esetében nem jelent információveszteséget (Neusch, 2014). A stopszavak listáját definiálhatjuk a megoldandó probléma függvényében mi magunk is, de általában a szövegbányászati programcsomagok és könyvtárak tartalmazzák a legelterjedtebb ilyen kifejezéseket. Speciálisan egy adott elemzési problémához tartozó legpontosabb stopszó-lista előállítása úgy történhet, hogy a korpusz szavait gyakoriság szerint rendezzük, majd a leggyakoribb, és a legritkább kifejezésekből kiválogatjuk a kontextus függvényében irreleváns, és a későbbi feldolgozás során eldobni kívánt elemeket (Manning *et al.*, 2009). A leggyakoribb szavak között általában névelőket, kötőszavakat stb. találunk, míg a legritkább szavak között előfordulhatnak például elírások.

A legtöbb probléma esetében, ahol a szöveg statisztikai elemzése által levonható következtetésekre vagyunk kíváncsiak, érdemes az egyes szavakat visszavezetni a közös ősre, azaz kanonikus, szótári alakra hozni (Tikk *et al.*, 2007), hogy az elemek

egyezőségét az alakjukból fakadó különbségek ellenére fel tudjuk tárni (Manning *et al.*, 2009). Kivételt képeznek az olyan esetek, ahol fontos egy-egy szó pontos, árnyalt jelentése a szövegkörnyezetben, amikor érdemes inkább megtartani a toldalékokat, névutókat stb., melyek módosíthatják a szavak jelentését, ezzel befolyásolva a kontextust. Tehát ugyanúgy, ahogy a többi szövegbányászati előfeldolgozási feladat esetében is, érdemes a konkrét probléma függvényében dönteni ennek a lépésnek az alkalmazásáról.

A szavak „közös alakra hozásának" elterjedt két módszere az algoritmikus alapon működő szótőképzés (csonkolás vagy angolul *stemming*), és a morfológiai alapokon nyugvó, szótárra épülő lemmatizálás (Sebők *et al.*, 2016). „A két eljárás között az a különbség, hogy míg a nyelvészeti motivációjú lemmatizálás mindig értelmes szóalakot állít elő, addig a szótövezés során jellemzően a szó csonkolása történik, amely gyakran nem értelmes szótári alakot ad eredményül" (Tikk *et al.*, 2007, p. 41). A szótövezés egy „nyers heurisztika (*crude heuristic*)" (Manning *et al.*, 2009, p. 32). A lemmatizálás tehát pontosabb, azonban számításigényesebb eljárás, míg a szótőképzés gyors, mivel relatív kisszámú, kódba írt szabály alapján dolgozik, viszont ebből kifolyólag pontatlanabb is (Neusch, 2014). Smith (2011) ezt úgy fogalmazza meg, hogy a „csonkolás a szegény ember lemmatizációja".

Ha a toldaléklevágást technikai oldalról vizsgáljuk, számos szoftveres megoldást találunk az ilyen jellegű feladatok elvégzésének támogatására. Például a jMorph egy „Java alapú morfológiai elemző" szoftver a magyar nyelvhez, mely tartalmaz lemmatizáló modult is (Incze *et al.*, 2005), míg a Snowball egy a csonkoló alkalmazás szintén a magyar nyelvhez (Tordai és de Rijke, 2005). Az angol nyelvhez az első és legismertebb csonkoló algoritmus a „*Porter stemmer*" (Porter, 1980). Mesterképzésen írt szakdolgozatomhoz végeztem teszteket a jMorph és a Snowball programokkal. A 1. táblázat a jMorph és a Snowball programokkal végzett tesztek néhány eredményét tartalmazza, melyekből jól látható a két eljárás közötti különbség (Neusch, 2014).

Szó	**Szótári alak**	**Stem**	**Lemma**
babakocsijáért	babakocsi	Babakocs	babakocsi
bajlódni	bajlódik	Bajlódn	bajlódik
munkámmal	munka	Munka	munka
lebélyegez	lebélyegez	Lebélyegez	bélyegez

1. táblázat: Lemmatizáló és szótőképző megoldások nem reprezentatív összehasonlítása

Fontos továbbá megjegyezni, hogy mind a szótőképző mind a lemmatizáló megoldások implementációfüggők, azaz a különböző megvalósítások ugyanarra az inputra más-más eredményt adhatnak, ahogy azt Thomas (2019b) bemutatja az *Nltk* és a *Spacy* programcsomagokon keresztül. Példájában a `['feet', 'foot', 'foots', 'footing']` bemeneti vektor esetén, míg az előbbi alkalmazásával a lemmatizálás eredménye a `['foot', 'foot', 'foot', 'footing']` vektor, addig az utóbbi használatával a `['feet', 'foot', 'foots', 'footing']` (Thomas, 2019b). A szótövezés esetében ugyanerről Manning és munkatársai (2009, p. 34) adnak áttekintést a Lovins, a Porter és a Paice csonkoló algoritmusok ugyanazon szövegen való összehasonítása által, melyből szintén jól látszik, hogy az eredmény mennyire függ az adott implementációtól. Például a *'reveal'* szót – az előbbi felsorolás sorrendjében – az algoritmusok a következőképen csonkolták: *'reve'*, *'reveal'* és *'rev'*.

3.3.2 A szöveg ábrázolása

A korpusz előzőekben részletezett előfeldolgozásával a szövegbányászati algoritmusok számára egy jobban kezelhető, strukturáltabb adathalmazt hozunk létre. A szöveget azonban az előkészítésen felül valamilyen módon ábrázolni, kvantifikálni is szükséges, annak érdekében, hogy algoritmikusan kezelni tudjuk. Az elemzési probléma függvényében „a dokumentumok reprezentálására három megközelítés terjedt el: a halmazelmélet alapú, az algebrai és a valószínűségi modell" (Tikk *et al.*, 2007, p. 25). Ezekben a modellekben a dokumentumokat az előkészítés után egyszerű listákkal, vagy komplex objektumokként például vektor vagy mátrix alakban ábrázolják, vagy valószínűségi eseményként kezelik.

A leggyakoribb ilyen ábrázolási mód, amennyiben az elemzéshez nincsen szükség a szavak sorrendjének megtartására, a szózsák (*bag of words*) modell. A szózsák modell segítségével egy dokumentumban található *token*eket és azok előfordulási gyakoriságát ábrázolhatjuk egy listában, melybe az elemeket annyiszor tesszük bele, ahányszor az eredeti dokumentumban is előfordultak (*multiset*).

Azt a modellt, melyben az unigramok szógyakoriságát (*tf*, *term frequency*) egész számokat tartalmazó vektorként ábrázoljuk, vektortér modellnek nevezzük (Russell és Norvig, 2005, p. 748). A modellben minden vektor a korpusz egy dokumentumát reprezentálja. Egy-egy vektor lehet bináris is, amennyiben csak annyit szeretnénk megjeleníteni benne, hogy a dokumentum tartalmaz-e egy adott tokent vagy sem, de

más súlyokat is rendelhet az egyes kifejezésekhez, például a relatív gyakoriságot, vagy a *tf-idf* értéket. Nasir és munkatársai megemlítik a kifejezésgyakoriság logaritmusát is, mint a vektortér modellben gyakran használt értéket (2020, p. 4). Ez utóbbit Manning és munkatársai a *tf* súlyok szublineáris skálázásának nevezik, melyre azért lehet szükség, mert „valószínűtlen, hogy egy dokumentumban 20 alkalommal előforduló kifejezés hússzor szignifikánsabb lenne az adott dokumentum szempontjából, mint egy egyszer előforduló" (2009, p. 126). Általánosan megfogalmazva tehát azt feltételezzük, hogy egy kifejezés relevanciája annál magasabb, minél nagyobb *tf* érték tartozik hozzá, de a relevancia nem lineárisan nő a kifejezésgyakoriság emelkedésével. Manning és szerzőtársai idézett könyvükben kiemelik továbbá a maximum *tf* normalizálás technikáját, melynek során a dokumentumban előforduló kifejezésgyakoriság-értékeket a maximális *tf* értékkel normáljuk, annak érdekében, hogy „azon anomália hatását csökkentsük, hogy a hosszabb dokumentumokban, pusztán hosszuk miatt, a *tf* értékek általában magasabbak" (2009, p. 127).

Az előzőekre építve számítják egy szó vagy szókapcsolat *tf-idf* (*term frequency – inverse document frequency*) értékét, amely azt számszerűsíti, hogy mennyire fontos az adott kifejezés az adott dokumentum szempontjából a dokumentumok teljes kollekciójához (korpusz) viszonyítva, azaz a kifejezés mennyire jellemzi az adott szöveget (Lane *et al.*, 2019). Az első faktor általában az adott token normalizált gyakorisága a dokumentumban, míg „a második faktort arra használják, hogy nagyobb súlyt adjanak azoknak a szavaknak, melyek csak pár dokumentumban találhatók meg" (Jurafsky és Martin, 2018, p. 113), azaz kisebb súlyt kapjanak a nagyon általános kifejezések, melyekről azt feltételezzük, hogy nem annyira relevánsak (például a korpusz specifikus stopszavak). Tehát az első egy dokumentumspecifikus „lokális" mérőszám, míg utóbbi egy korpuszspecifikus, „globális" metrika. A dokumentumgyakoriság (df_t), azon dokumentumok száma, melyek tartalmazzák *t* kifejezést, míg az inverz dokumentumgyakoriságot az $idf_t = log\frac{N}{df_t}$ képlettel határozzák meg, ahol N az összes dokumentum számát jelöli. A képletben annak, hogy a logaritmusnak milyen alapot választunk, a rangsorolásra nincs hatása (Manning *et al.*, 2009, p. 118).

„Ezen a ponton a dokumentumainkat egy-egy vektorként tudjuk ábrázolni, ahol az egyes dimenziók az egyes kifejezések súlyai, vagy 0 a szótár azon elemei esetében, melyek nem szerepelnek egy adott dokumentumban" (Manning *et al.*, 2009, p. 119).

A dokumentum-kifejezés mátrix (*document-term matrix*) az előzőek mátrix formában való ábrázolása, ahol a mátrix sorai a dokumentumokat, míg oszlopai az egyes kifejezéseket reprezentálják.

3.3.3 Az N-gram modell

A szózsák modellben az egyes tokenek sorrendjéből származó információtartalmat elveszítjük. A probléma megoldásának egyfajta megközelítése az, ha a különálló tokenek helyett az azokból képzett n hosszú szekvenciákat, *n-gram*okat vizsgáljuk. Klasszikus értelemben az *n-gram* kifejezést valószínűségi modellekben használják, ahol a cél annak megbecsülése, hogy egy természetes nyelvi jelekből álló szekvencia esetén mennyi a valószínűsége egy adott n. elemnek, az azt megelőző $n-1$ elem alapján (Brown *et al.*, 1992). Ilyen értelemben az *n-gram* kifejezést Shannonnak tulajdonítják, aki azt 1948-as, A kommunikáció matematikai elmélete című művében írta le (Shannon, 1948). A gyakorlatban egyes karakterszekvenciák megjelölésére, melyek hossza q, a *q-gram* vagy *karakter n-gram* elnevezés használatos inkább (Ukkonen, 1992; Voronov, 2020), míg leggyakrabban az *n-gram* kifejezésen együttesen elforduló, egymást követő nyelvi objektumokat (szavak) értenek (Szirmai, 2005). Én is ezt a konvenciót fogom használni jelen dolgozatban.

3.3.4 Szövegbányászati alkalmazások

A szöveg előzőekben részletezett előkészítése általában azért történik, hogy olyan formába alakítsuk azt, amit a konkrét elemzési kérdésre választ adó algoritmus fel tud használni. A 6. ábrán láthatóakkal összhangban a leggyakoribb szövegbányászati problémák az információkinyerés és kivonatolás, illetve az osztályozás és a csoportosítás. Ezek mellett a legfontosabb alkalmazási területek közé tartozik a tartalomkeresés és a véleményanalízis is (Kő, 2013). Ide sorolható még szófaji egyértelműsítés (*part-of-speech tagging*) és a névelem-felismerés (*named-entity recognition, NER*) is, melyek tulajdonképpen feldolgozási részfeladatnak is tekinthetők.

A szófaji egyértelműsítés egy számítógépes nyelvészeti feladat, ami „a szövegkorpuszban található szavakat általános lexikai jelentésük és kontextusuk alapján megjelöli és felcímkézi” (Sebők *et al.*, 2016, p. 78). Egy adott szó szófaja és mondatban elfoglalt helye alapján következtetni tudunk a valószínűsíthető szomszédokra, illetve a kontextusra is (Jurafsky és Martin, 2018).

Névelem-felismerés alatt a szövegben található tulajdonnevek (személyek, földrajzi helyek stb.) és egyéb speciális objektumok, például telefonszámok algoritmikus beazonosítását és annotálását értjük (Sebők *et al.*, 2016). A szófajok feltárására és a névelemek felismerésére nagymértékben támaszkodnak az információkinyerési és kivonatolási feladatok során is. A névelemek normalizációja (*named-entity normalization, NEN*) során, Khalid és szerzőtársai (2008) alapján, két olyan problémára próbálnak megoldást találni, amely az egyes kifejezéseknek egy jól beazonosítható objektumhoz rendelését nehezítik meg. Az első, amikor az egymástól különböző, de azonos nevű fogalmakat próbálják meg egymástól megkülönböztetni (névelem-egyértelműsítés, *named-entity disambiguation*). A második a szinonimák problémája, amikor ugyanarra az entitásra több néven is hivatkoznak.

3.3.4.1 Információkinyerés és kivonatolás

Az információkinyerő és kivonatoló eljárások célja, hogy az elemzett dokumentumokban megtalálják és összegyűjtsék a felhasználó információigényét kielégítő, vagy általában a korpuszt jellemző releváns tartalmakat, azaz strukturált információt állítsanak elő a szövegből. Az információkinyerés célja a válasz megtalálása egy konkrét felhasználói lekérdezésre, azaz „az adott feladat szempontjából fontos szövegrészek (információk, tények) kigyűjtése" (Fajszi *et al.*, 2010, p. 271). Az információkinyeréssel szemben, kivonatolás esetén a cél a dokumentumot leginkább meghatározó, legjobban leíró részek megtalálása, azaz tulajdonképpen a szöveg összefoglalása.

A kivonatolás és az információkinyerés során többféle megközelítést is használnak, sokszor kombinálva is az egyes módszereket. Ezek a megközelítések általában statisztikai, illetve szemantikai alapokra vagy a szöveg szerkezetének elemzésére építenek. A statisztikai alapú eljárások – mint például a szavak egyszerű, vagy relatív gyakoriságának, illetve *tf-idf* értékének számítása – segítségével megállapítható, hogy mely kifejezések a legjellemzőbbek az adott dokumentumra, azaz melyek az úgynevezett kulcsszavak. Ezek ismeretében például a kivonatolás során meghatározhatók azok a mondatok, illetve a szövegnek azon pontjai, melyek a legvalószínűbben jellemzik az adott dokumentumot (Neusch, 2014).

A statisztikai módszereken kívül Tikk és szerzőtársai (2007) alapján elterjedten használnak különböző szerkezeti elven működő módszereket is, mint amilyen a címbeli szavak kulcsszóként kezelése, a kifejezések egyes meghatározott előfordulási

helyeinek, például kivonat (*abstract*), vagy konklúzió kiemelt módon kezelése stb. Manning és munkatársai (2009) szintén leírják, hogy egyes algoritmusok nagyobb súllyal kezelnek egyes szövegpozíciókat, mint például az első vagy az utolsó bekezdés.

A szemantikai elven működő módszerek sokszor olyan nyelvi elemeket keresnek, mint az utaló frázisok, az idézésre utaló szavak vagy a névelemek, és ezek alapján próbálnak következtetni az egyes szövegrészek relatív fontosságára. Az információ-visszakeresésben elterjedtek továbbá a mintaillesztésen alapuló módszerek, például a reguláris kifejezések használata (Tikk *et al.*, 2007).

3.3.4.2 Osztályozás és csoportosítás

Osztályozás és a csoportosítás alatt olyan modellek és eljárások összességét értjük, melyeket dokumentumok rendszerezésére, kategóriákba sorolására használnak. A különbség a két feladattípus között az, hogy ismerjük-e előre az egyes osztályokat, melyekbe a dokumentumainkat be szeretnénk sorolni.

Egy osztályozási (*classification*) feladat esetén ezek a kategóriák előre ismertek. Az osztályozandó dokumentumokat két csoportra osztják. Az egyik az úgynevezett tanítódokumentumok csoportja, melynek elemeit manuálisan besorolják a megfelelő osztályokba. A tanítódokumentumok attribútumai alapján az osztályozást végző algoritmus fel tudja térképezni, meg tudja tanulni az egyes osztályok tulajdonságait, például a jellemző szavakat, kifejezéseket stb. Általánosan az adatbányászatban a tanítópontok egy részét nem tanításra, hanem arra használják, hogy teszteljék a modell hasznosságát (Bodon, 2010). A kukucskálás jelenségének elkerülésére úgynevezett validáló halmazt is szokás alkalmazni, amely a modell paraméterbeállításainak finomhangolására szolgál. A tanítás során megalkotott szabályokat a későbbiekben az ismeretlen kategóriájú dokumentumokon alkalmazva, az osztályozást végző algoritmus megpróbálja azokat a helyes kategóriába sorolni. A gépi tanulásnak ezt a módját a tanulóadatokon keresztül felügyelt tanulásnak (*supervised learning*) nevezzük (Tikk *et al.*, 2007).

A dokumentum-osztályozás során leggyakrabban használt algoritmusok az összetettebb statisztikákon alapuló dokumentumtávolság-mátrixok, a neurális hálók, a döntési fa alapú módszerek és a legközelebbi szomszédokon alapuló eljárások. Egyes esetekben az osztályozás során taxonómiák, vagy ontológiák felhasználására is sor

kerülhet, mintegy kiegészítendő az előzőekben említett módszereket szemantikán alapuló eljárásokkal is (Neusch, 2014).

A csoportosítás (*clustering*) szövegek kategorizálásának módja arra az esetre, mikor „nem rendelkezünk semmilyen a priori kategóriarendszerrel, címkékkel az adatok struktúráját vagy jellemzőit illetően, és a létrehozandó csoportok számáról sincsen előzetes tudásunk" (Sebők *et al.*, 2016, p. 123). A klaszterezés során a cél „a dokumentumokból olyan elkülönülő csoportokat alkotni, hogy az egy csoportba kerülők minél hasonlóbbak, az eltérő csoportokban lévők pedig minél különbözőbbek legyenek" (Fajszi *et al.*, 2010, p. 279). A dokumentumok klaszterezésére használt módszereket annak függvényében, hogy egy adott elem esetében megengedjük-e, hogy több csoportba is beletartozzon, vagy sem, kétféleképpen osztályozhatjuk. A szigorú módszerek esetében egy dokumentum egy csoportba tartozhat, míg a lágy eljárások esetén akár többe is (Tikk *et al.*, 2007, p. 146). Mivel csoportosítási feladatok esetében a modelleket nem tanítódokumentumokon keresztül „tanítjuk" be, ezért a kapcsolódó módszereket felügyelet nélküli tanulás (*unsupervised learning*) néven hivatkozzák.

Egy tipikus csoportosítási feladatot ismertet Evangelopoulos és Visinescu (2012). A szerzők Barack Obama 2009-es afrikai látogatása kapcsán emberek a Fehér Háznak – több afrikai országból – küldött SMS üzeneteit elemezték. Az üzenetekből azt próbálták meghatározni, hogy az egyes országokban melyek azok a témák, amik leginkább foglalkoztatják az embereket, és ez alapján próbálták meg felkészíteni az elnököt, illetve kiválasztani a megfelelő kommunikációs stratégiát.

3.3.4.3 Véleményanalízis

A véleményanalízis (*sentiment analysis*) tulajdonképpen egy „speciális osztályozási feladat" (Sebők *et al.*, 2016, p. 73). Inputját a különböző internetes médiákban, fórumokon, tematikus és közösségi oldalakon fellelhető hozzászólások, ügyfélszolgálati adatbázisok, SMS-ek és egyéb, jellemzően felhasználók által generált tartalmak képezik (Neusch, 2014). Célja egy adott téma (termék, közéleti szereplő, politikai entitás stb.) megítélésének, illetve az emberek hozzá való érzelmi viszonyulásának feltárása (Sebők *et al.*, 2016), azaz algoritmusok segítségével „számszerűsíteni bizonyos szövegek vélemény-polaritását valamilyen pozitív-negatív skálán" (Szekeres, 2013).

A véleménybányászatot gyakran használják a többségi vélemény meghatározására, illetve a felvetett problémák, érvek, megoldási javaslatok intelligens feldolgozásával bizonyos vállalati és politikai döntések előkészítésére is (Neusch, 2014). Evangelopoulos és szerzőtársa szerint ez megalapozhatja az e-demokráciát, ami a szövegbányászat hozzájárulása a társadalmi innovációhoz, az állampolgárok és a kormányzat közötti kommunikáció elősegítése az „emberek hangjának" összegzése által, és a politikusok felruházása azzal a képességgel, hogy hatékonyan értelmezzék a polgárok visszajelzéseit (Evangelopoulos és Visinescu, 2012). Ugyanakkor természetesen, mint minden technológiát, a véleménybányászatot is fel lehet használni erkölcsileg kifogásolható módon, így az a jelenlegi, „igazság utáni (*post truth*) korban" a populista politika és a tömegmanipuláció eszközkészletének is szerves részévé vált (Krekó, 2018).

3.3.5 Webbányászat

A webbányászatot a szövegbányászat egy speciális ágának is tekinthetjük, ahol a cél a weben található hiperszöveges dokumentumok feldolgozása és elemzése. A webbányászatot a vizsgálat céljának függvényében három további területre szokták bontani, melyek:

- webes tartalom bányászata (*web content mining*),
- a web struktúrájának feltérképezése (*web structure mining*) és
- a felhasználáshoz kapcsolódó adatok elemzése (*web usage mining*) (Khalil és Fakir, 2017).

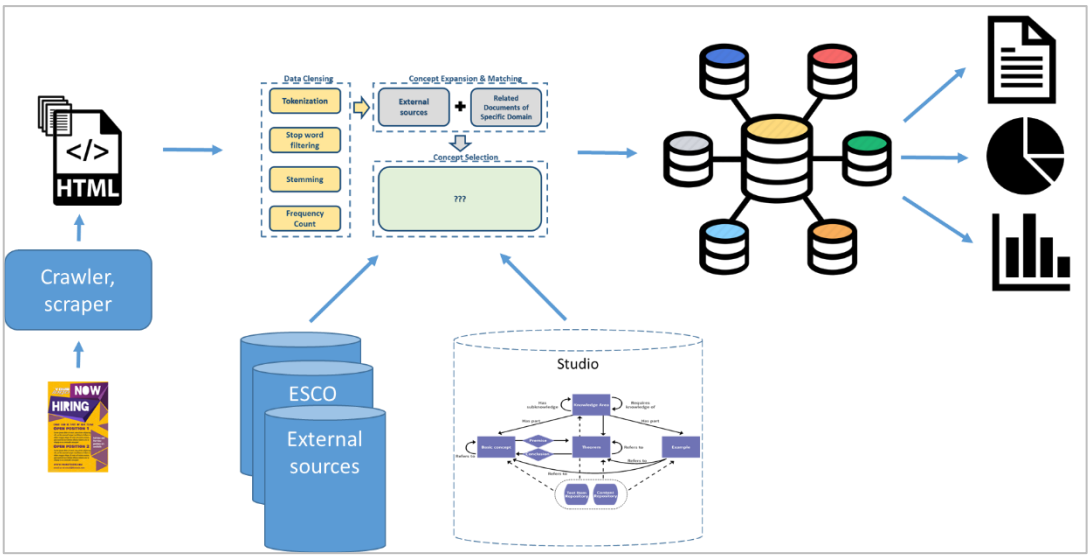
Jelen tézisben álláskereső portálok adatait dolgozom fel, így az input adatok összegyűjtése egy webes tartalom bányászati feladat. Az webbányászati adatgyűjtés legfontosabb eszköze az úgynevezett *crawler*, ami egy webes tartalmak letöltésére és feldolgozására szolgáló alkalmazástípus neve. Szinonimaként használják még a „*spider, robot, worm* illetve *walker"* megnevezést is (Ceri *et al.*, 2013). A szakmai terminológiában megkülönböztetik a webes tartalmakon dolgozó robotokat az alapján, hogy végeznek-e valamilyen feldolgozást az adott oldal tartalmán, illetve valamilyen céllal rögzítik-e azt. Ennek megfelelően a leggyakrabban a *spider* megnevezést a *HTML* oldalakon található linkek feltérképezésére használatos alkalmazásokra, a *crawler* kifejezést pedig azokra a robotokra használják, melyek az egyes oldalak tartalmával nem törődnek, azokat csak bejárják és céljuk elsősorban az indexálás,

például valamilyen keresőszolgáltatás támogatása érdekében (Khalil és Fakir, 2017). Az előzőekkel szemben a *scraper* célja, hogy kinyerje az adatokat a weboldalakról és azokat valamilyen strukturált formában rögzítse. Erre a célra általában valamilyen előzetesen felépített összerendelési sémát használnak, ami alapján meghatározott *DOM* (*document object model*) elemeket valamilyen adatstruktúrában (*JSON*, *XML*, adatbázis stb.) tudnak tárolni (Montalenti, 2012). Általában a *DOM* elemekben tárolt információ beazonosításra *CSS Selector*okat vagy *XPATH*-t használnak. Lawson (2015) alapján a különbség az egyes robottípusok között az, hogy míg a *crawler* letöltés után elveti a weboldalakat, addig a *scraper* archiválja őket, és adatot próbál kinyerni belőlük. A *crawler* egy generikus alkalmazás, míg a *scraper* célzottan, specifikus információk kigyűjtésére szolgál (Jarmul és Lawson, 2017).

4 Kutatási keretrendszer

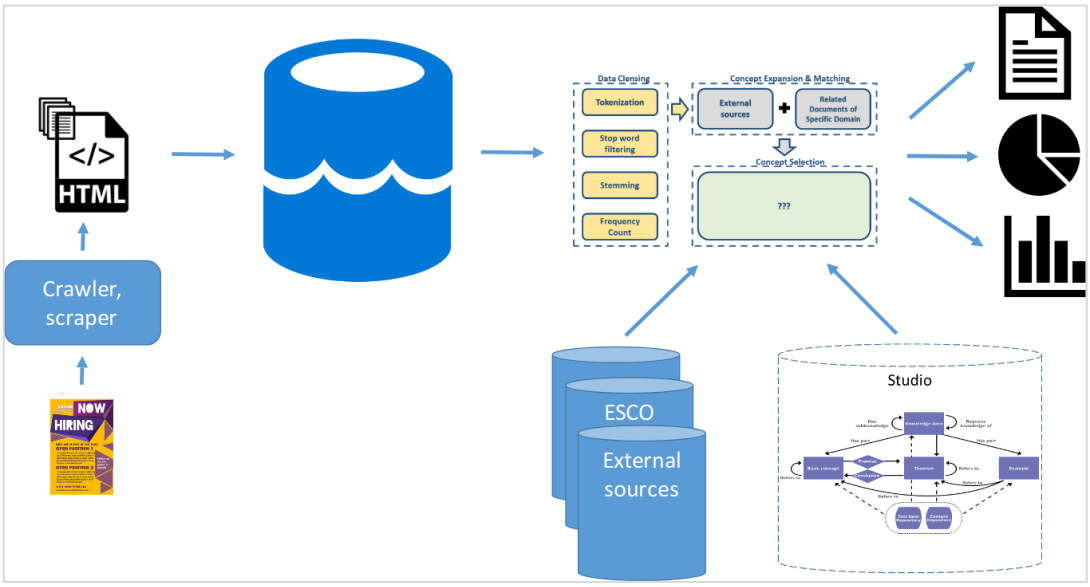
A 3. fejezetben ismertettem a kutatás elméleti hátterét, jelen fejezetben pedig a célom, hogy kontextusba helyezzem ezeket az ismereteket, és bemutassam, hogy hogyan kapcsolódnak a gyakorlati kutatáshoz. A 7. ábra a dolgozatban felvázolt keretrendszer architektúráját szemlélteti. További célom jelen fejezettel, hogy az ábrán látható összkép felől közelítve rámutassak azokra a pontokra, amelyeket a következő fejezetekben részletesen kifejtek. Technikai szempontból a rendszer főbb moduljai a következők.

1. Az input adatokat begyűjtő *scraper* (5.1. fejezet).
2. A kompetenciahalmazok beazonosítását végző szövegbányászati modul, ami az ábrán vázolt megközelítésben egy ETL (*extract, transform, load*) folyamat része. A szövegbányászati modul felső szintű áttekintését a 4.2. alfejezet tartalmazza, a 6. fejezet az explicit, míg 7. fejezet az implicit kompetenciák beazonosításának lehetőségeit tárgyalja kísérleti példák bemutatásával.
3. A külső (szemantikus és egyéb) források feldolgozásba kapcsolását lehetővé tevő interfész. A kísérletek során felhasznált ontológiákat a 4.1 fejezetben ismertetem, a kapcsolatot lehetővé tevő interfész implementációjára jelen dolgozat nem tér ki.
4. Az adattárolási eszköz (adattárház (7. ábra) vagy adattó (8. ábra)). A tárolni kívánt adatok körét az 5.2. fejezet mutatja be, míg az implementálni javasolt architektúrát és a kiválasztás szempontjait az 5.3. és 5.4. alfejezetek tartalmazzák.
5. És a 4.-re épülő analitikai megoldás, melynek bemutatása túlmutat a dolgozat keretein.



7. ábra: A kutatás adattárházra épülő architektúrája (saját szerkesztés Gábor et al.*, 2016; Gillani és Kő, 2014 felhasználásával)*

Amennyiben a keretrendszert *big data* megközelítéssel valósítjuk meg, úgy ez, a 7. ábrán szemléltetett architektúra modell némiképpen megváltozik. Mivel egy adattó esetében az adatokat általában előfeldolgozás nélkül töltjük be egy elosztott fájlrendszerre, és a logikát a szükséges információk kinyerése során alkalmazzuk azokon (*schema on read*), így a középső komponensek sorrendje felcserélődik, illetve természetesen az adattárház komponenst ez esetben felváltja az adattó (8. ábra).



8. ábra: A kutatás adattóra épülő architektúra modellje (saját szerkesztés Gábor et al.*, 2016; Gillani és Kő, 2014 felhasználásával)*

Ahogy arról már volt szó, a dolgozatban felvázolt fő felhasználási cél esetében az inputokat álláshirdetések adják, de ugyanígy elképzelhető olyan használati eset is, ahol folyamatmodellekkel dolgozunk. Akár egyiket, akár másikat használjuk, elmondható,

hogy a feldolgozási folyamat bemeneti oldalán rosszul vagy félig strukturált, jellemzően szabad szöveges formában adott dokumentumok jelennek meg. A bemeneti adatállományok közös jellemzője, hogy olyan vállalati objektumokat írnak le, reprezentálnak, melyekhez munkavégzés kötődik, azaz egy munkakört vagy pozíciót. Ezek jellemzője egyrészt, hogy végrehajtó is társul hozzájuk, továbbá hogy leírásaikban beazonosíthatók olyan kifejezések, melyek a kapcsolódó feladatok elvégzéséhez szükséges kompetenciaelemeket jelölnek. A kontextus függvényében ezeket a kompetenciákat azok különböző attribútumai is kiegészíthetik.

Az álláshirdetésekben általában pozícióra (állás, beosztás) keresnek embert, ami egy alkalmazott szempontjából egyedi, részletes leírása annak, amit el kell végeznie, azaz a hozzá rendelt feladatok (*task*) összessége. Ezzel szemben a munkakör vagy szerepkör, (*job, job role*) feladatok, kötelességek (*duty*) és felelősségek (*responsibility*) összessége, melyek hasonló feladat- és felelősségi körrel rendelkező pozíciókra érvényesek. A felvázolt megoldás szempontjából munkakörökkel, azok leírásával általában a folyamatmodellekben találkozunk. Jelen kontextusban érdemes még megemlíteni a „foglalkozás" (*occupation*) fogalmát, ami egy általánosabb megfogalmazás, például szoftverfejlesztő. A foglalkozások jegyzékét az egyes országok és nemzetközi szervezetek legtöbbször nomenklatúrákban (ISCO, FEOR stb.) rögzítik, melyek sokszor statisztikai célokat szolgálnak és nehezen változnak. Az oktatás vonatkozásában a foglalkozások végzettséghez, míg az állások képzettséghez vannak kötve (Gábor, 2019).

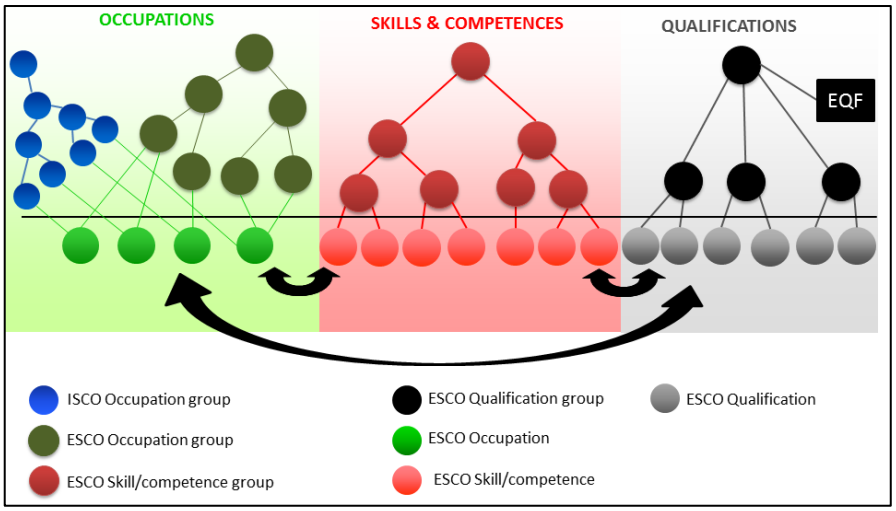
A felvázolt keretrendszerben tehát a folyamatmodellek és az álláshirdetések reprezentálják a munkaerőpiaci keresletet, és feltételezzük, hogy kinyerhetőek belőlük azok a kompetenciaelemek, melyekre az azokat meghirdető vállalatnak szüksége van. Az álláshirdetésekben explicit megtalálható, és azokhoz implicit köthető kompetenciaelemek beazonosításának támogatásához külső ontológiákat is felhasználtam, melyeket a következő alfejezetben röviden bemutatok

4.1 Felhasznált külső ontológiák

4.1.1 ESCO

Az ESCO (*European Skill, Competences, Qualifications and Occupations*) ontológia – ami fontos felhasznált eszköz jelen kutatásban, hiszen a felépített kompetenciaszótár elemeinek jelentős része innen származik – szemantikus osztályozása az európai

térségben jellemző végzettségeknek és foglalkozásoknak, melyek között a kapcsolatot készség- és kompetenciaelemek segítségével teremti meg a keretrendszer (Smedt *et al.*, 2015), ahogy az a 9. ábrán látható (Boomgaert, 2013). Az ESCO jelentős űrt tölt be számos területen, hiszen kidolgozása előtt nem volt széles körben elfogadott, országok közötti összehasonlítást lehetővé tevő, foglalkozások kompetenciakövetelményeit kódoló séma (Handel, 2012; Wowczko, 2015).



9. ábra: Az ESCO 3 pillére és a köztük lévő kapcsolatok (Boomgaert, 2013)

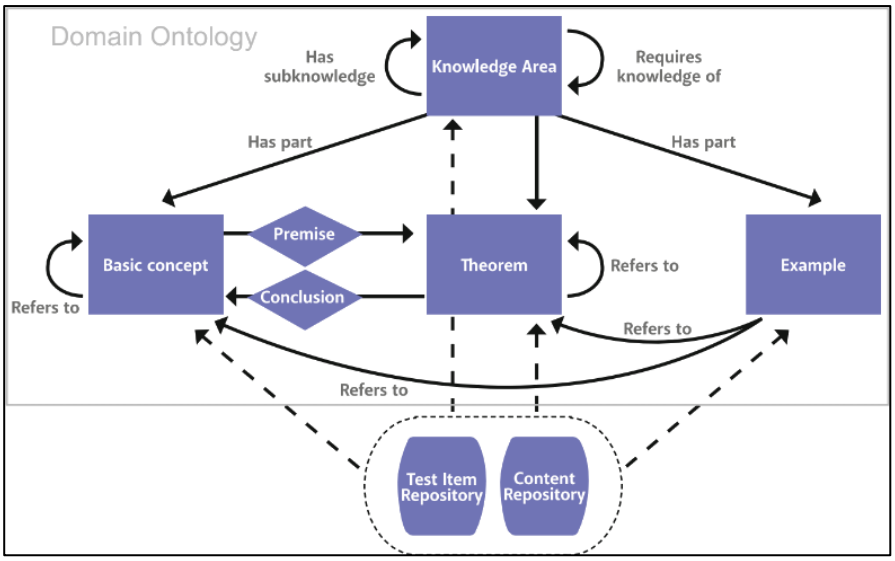
Az ESCO ontológia létrehozásának elsődleges célja, hogy az európai munkaerőpiacot támogassa mind a kínálati (munkavállalói), mind a keresleti (vállalati) oldalon. Ahogy Smedt és szerzőtársai (2015) megfogalmazzák, erre a megkülönböztetett támogatásra a munkaerőpiac sajátosságai miatt van szükség. Ezek az egyedi tulajdonságok a szemantikus internetes technológiákat nagymértékben alkalmassá teszik arra, hogy hozzáadott értéket termeljenek a területen. A legfontosabb karakterisztikája a munkaerőpiaci „áruknak", hogy minden egyes kínált és keresett „termék" egyedi, mivel nincs két egyforma képességű munkavállaló, és elhanyagolható azon nyitott pozíciók száma, melyek egyforma erőforrás-szükséglettel rendelkeznek. Így amennyiben az egyes foglalkozások és végzettségek kompetenciatartalmának részletes leírására képesek vagyunk, úgy közelebb kerülünk ahhoz, hogy egyszerűen megtalálhassuk a „megfelelő embert a megfelelő pozícióra" (Smedt *et al.*, 2015, p. 1). Vrang és munkatársai (2014) az ESCO-t egy olyan központi eszközként (*exchange hub*) írják le, amely megteremti a kapcsolatot az egyes országok nemzeti foglalkoztatási szolgálatai és foglalkozási nómenklatúrái között, és mindezt többnyelvű módon. Az előzőekre a szerzők egy olyan példát hoznak, melyben egy

munkaerő-allokációt támogató alkalmazás, az ESCO segítségével, lengyel sebészeti ápolók elhelyezkedését segíti Franciaországban (Vrang *et al.*, 2014).

Jelen tézis szempontjából az ESCO ontológia elsősorban kompetenciaszótárként funkcionál, illetve a foglalkozáskapcsolatokon keresztül a látens kompetenciák beazonosításának eszköze lehet (32. ábra).

4.1.2 STUDIO

A STUDIO egy ontológia alapú e-learning metodológia, és adaptív tudástesztelő keretrendszer (Vas, 2007). A STUDIO központi modulja az ontológia (10. ábra), mely önmagában használható, ontológiaszerkesztővel rendelkező alkalmazás, de a rendszerben kiegészül egy tudáshiányokat feltáró adaptív tesztelési modullal és az ahhoz kapcsolódó e-learning részrendszerrel (Weber és Vas, 2015).



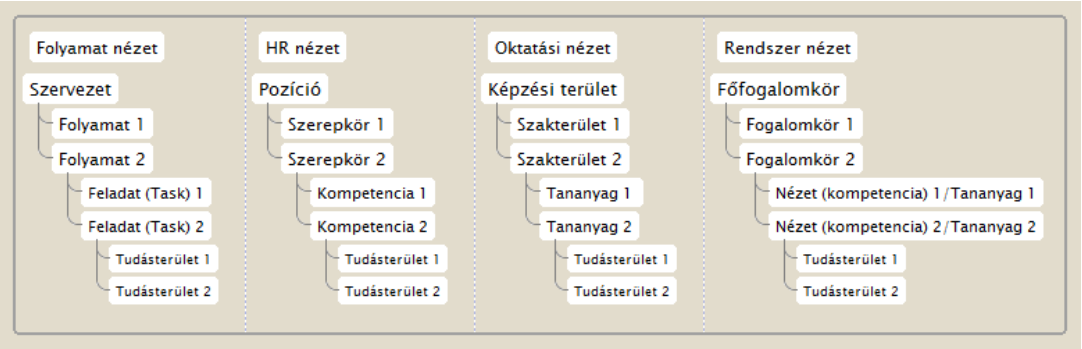
10. ábra: A STUDIO ontológia modellje (Gábor et al.*, 2016, p. 88)*

A STUDIO ontológiája számos – egyes tudásterületeken keresztül akár kapcsolódó – szakterületi „részontológiát[16]" fog össze (Szabó és Neusch, 2015). A szakterületi ontológia és a tudástesztelés közötti logikai kapcsolatot az úgynevezett *fogalomkör* (*concept group*) objektum hivatott megteremteni, ami tulajdonképpen az ontológiának, egy gyakorlati szempontú, az adott felhasználásnak megfelelő leszabása.

Míg tehát a STUDIO-ban az ontológia az „univerzum" fogalmainak rendszerezésére használható, addig a fogalomkörök az ontológia gyakorlati szempontú reprezentációját

[16] STUDIO rendszer szempontjából hívjuk csak ezeket a struktúrákat rész- vagy szakterületi ontológiáknak, azonban mindegyik részontológia önmagában egész, értelmes reprezentációja lehet egy adott tudományterület tudásának.

teszik lehetővé számos alkalmazási területen, amilyen például az oktatás, az emberierőforrás-menedzsment vagy a vállalatok intellektuális tőkéjének folyamatalapú reprezentációja (11. ábra).



11. ábra: Fogalomkör struktúra a STUDIO rendszerben (Neusch, 2014)

A fogalomkörök rendszere rugalmasan építhető fel, mintegy építőkockákként használva és összekapcsolva a felhasználási esetet leíró tudásterületeket az ontológia akár egymástól távoli, közvetlenül nem kapcsolódó részeiről. Azonban bár struktúrájában a rendszer jelentős szabadságot ad a modellező számára, az egyes ontológiarészek belépési pontját követően mégis megőrzi az adott „részhálózat" hierarchikus felépítését és eredeti kapcsolatait. Mivel az eredeti ontológia logikáját tekintve az általános elemektől az egyre inkább specifikus tudásterületek felé építkezik, így a felépített fogalomkör tartalmazza azokat a tudáselemeket is, melyek ismeretére szükség van ahhoz, hogy a közvetlenül a leszabott struktúrába kapcsolt elem ismeretét elfogadhassuk. Bár az ismertetett logika elsősorban tudásteszteléshez lett kitalálva, azonban jelen tézis céljait is támogatja, mivel az állásajánlatokban beazonosított kompetenciaelemekhez kapcsolódó egyéb, az adott hirdetések (pozíciók) szempontjából látens tudáselemek beazonosítására és ezáltal a kontextus gazdagítására tesz képessé. Emellett a STUDIO ontológiába felvett egyes tudásterületek szintén a kompetenciaszótár részét képezik a kísérletek során.

4.1.3 O*NET

Az O*NET[17] (*Occupational Information Network*) az Egyesült Államok Munkaügyi Minisztériumának támogatásával megvalósult projekt. Kifejlesztésének egyik célja az volt, hogy kiváltsa az USA-ban 70 évig elődjeként szolgáló taxonómiát (Peterson *et al.*, 2001). Az O*NET tulajdonképpen nem egy ontológia, adatbázisa azonban sokkal

[17] https://onetonline.org

több egyszerű taxonómiánál, hiszen a foglalkozások listája és annak hierarchiája mellett, azokhoz kapcsolva tartalmazza az ellátásukhoz szükséges tudás-, készség- és kompetenciaelemeket, az általános végzettségi elvárásokat, statisztikákat és számos egyéb, a munkaerőpiacot egy átfogó modellben leíró információt. Jelen dolgozatban az O*NET rendszert elsősorban az álláshirdetések foglalkozáshoz rendelésének folyamatában használom fel a látens kompetenciák feltárásának érdekében.

4.2 Szövegbányászati modul

A dolgozatban felvázolt feldolgozási folyamat egy állásajánlatból, mint input dokumentum, indul ki, majd az ebben megjelenő kompetenciák beazonosítása után, azok szemantikus kiegészítésére törekszik, és az így kapott eredményt a felhasznált tárolási stratégia függvényében vagy közvetlenül, vagy az adattárházon keresztül visszacsatolja az felhasználóknak (képzőintézmény). Technikailag azonban a folyamat első lépése, azaz a szabadszöveges korpuszban a kompetenciákat, tudáselemeket és készségeket reprezentáló n-gramok (szavak és tetszőleges (n) elemű kifejezések, nyelvi szerkezetek) beazonosítása nem triviális. A feladat megoldásának lehetséges módjaival foglalkozik a 2. kutatási kérdés, melynek vizsgálata során a 6. fejezetben arra keresek megoldást, hogy miként lehetséges az eredetei korpuszban a releváns kompetenciákat reprezentáló n-gramok beazonosítása. Milyen módszerek és eszközök állnak rendelkezésre, amelyek alkalmazhatók a problémára, és segítséget nyújthatnak a cél eléréséhez?

A kompetenciaelemek kinyeréséhez olyan megoldások felhasználása szükséges, melyek segítségével képesek lehetünk a feladat szempontjából releváns információk megtalálására a nagy mennyiségű, rosszul strukturált, szövegesen, az élő nyelv segítségével leírt állásajánlatok halmazából. Ahogy azt az elméleti hátteret ismertető fejezetben is leírtam, a szövegbányászatnak, mint az adatbányászat alterületének célja mintázatok beazonosítása az ilyen jellegű adatokban, azaz az adott vizsgálandó szöveges korpuszban (Witten *et al.*, 2011). A korpuszt jelen probléma esetében tehát az állásajánlatok halmaza adja, míg a beazonosítandó mintázatok, „minőségi kifejezések” (Liu *et al.*, 2015) az értelmes és érvényes kompetenciaelemeket reprezentáló szavak és szókapcsolatok. Ez olyan információ, amely strukturáltan – azaz automatizáltan és könnyen beazonosítható, felismerhető módon – nem szerepel az elemzett szövegek összességében, így a feladat pontosan megfeleltethető a szövegbányászati információkinyerés definíciójának, amely Fajszi és munkatársai

(2010, p. 271) alapján "az adott feladat szempontjából fontos szövegrészek (információk, tények) kigyűjtése […] azaz strukturált információ előállítása".

A második kutatási kérdésben megfogalmazott probléma megoldását a 6. ábrán látható klasszikus szövegbányászati modell alapján képzelem el. Tehát az első lépésében az álláshirdetések korpuszának szövegbányászati előfeldolgozása a cél. Az előfeldolgozás utolsó lépésében a korpusz szavaiból – a szavak sorrendjét és a mondathatárokat figyelembe véve, azokból szópárokat, szóhármasokat stb. képezve – n-gramokat állítunk elő.

A 6. ábrán felvázolt modellben, az adattisztítás után a létrehozott n-gramok szűrése történik meg, annak érdekében, hogy beazonosíthatók legyenek azok, melyek a kiindulási probléma kontextusában érvényes tudáselemeket reprezentálnak. Az előállított n-gramok közül a releváns kompetenciák beazonosításának több útja is létezik. A 6. fejezetben kísérleteket végzek ezen kompetenciaelemek feltárására, egyszerű szógyakoriság-, illetve *tf-idf* alapú modellekkel és szótár felhasználásával is. Bemutatok továbbá egy logisztikus regressziót használó felügyelt tanulási módszert is, melyben magyarázó változókként az álláshirdetésekben található *n-gram*ok és a kompetenciaszótár egyes elemei között számolt hasonlósági metrikák értékeit használom.

Az explicit megjelenő kompetenciakifejezéseken kívül a külső rendszerek segítségével feltárhatóak olyan kapcsolódó tudáselemek, készségek stb., melyek direkt módon nem jelentek meg ugyan a hirdetés szövegében, azonban feltételezhető, hogy ennek ellenére relevánsak az adott pozíció ellátásához. A látens kompetenciák feltárásának lehetőségeivel foglalkozik a 7. fejezet, melyben az álláshirdetések címében található foglalkozások beazonosítására teszek kísérletet reguláris kifejezések és egyszerű szabályok kombinálásával, illetve egy döntési fára alapuló felügyelt tanulási módszer alkalmazásával, hogy annak segítségével a kapcsolódó ontológiákból vissza tudjam csatolni a foglalkozáshoz társított kompetenciaelemeket.

A keretrendszer általános ismertetése után a következőkben kitérek az egyes kutatási kérdéseimhez tartozó empirikus kutatásom bemutatására, amely az előbb vázolt keretrendszer elemeinek kialakítására, fejlesztésére irányul. Az adattárolási architektúra kiválasztásával (1. kutatási kérdés) az 5. fejezet foglalkozik részletesen. A 6. fejezetben olyan módszereket, illetve olyan kísérleteket mutatok be, melyek

segítségével az explicit (2. kutatási kérdés), míg a 7. fejezetben az implicit (3. kutatási kérdés) kompetenciák feltárására törekedtem.

5 Az adattárolási architektúra kiválasztása

A munkaerőpiaci adatokat első lépésben össze kell gyűjtenünk, fel kell dolgoznunk, tárolnunk kell, végül a szükséges információt ki kell belőlük nyerni. A kutatási keretrendszert bemutató fejezetben egy adattárházra és egy adattóra épülő magas szintű architektúrát vázoltam fel, mint a megvalósítás lehetséges irányait. Jelen fejezet célja, hogy a kapcsolódó alapfogalmak (3.1. fejezet) ismeretére alapozva bemutassa az egyes technológiák és implementációk közötti választás alapjául szolgáló szempontrendszert, illetve a kiépíteni javasolt megoldást.

5.1 Adatgyűjtés

Ahogy arról a kutatás kereteit ismertető alfejezetben már szó esett, az elvégzett kísérleteink során használt álláshirdetések elsődleges forrása az internet, pontosabban az Indeed állásportál. Az Egyesült Királyság munkaerőpiacára szánt hirdetések esetében az adatgyűjtés kiindulópontja a következő *URL*.

https://www.indeed.co.uk/jobs?q=information+technology&fromage=1&sort=date

A kidolgozott *scraper* alkalmazás ezen a ponton kezdi meg az adatgyűjtést, rögzíti az adott oldalon lévő álláshirdetéseket, majd a következő oldalra navigál automatikusan, és megismétli a kért adatok rögzítését, egészen addig, ameddig ez a szolgáltató oldalán lehetséges, azaz maximum a századik oldalig. Ez számszerűleg azt jelenti, hogy egy adott futás során az adatgyűjtő alkalmazás összesen 1000 hirdetést tud legyűjteni, még akkor is, ha a szolgáltató adatbázisában esetleg több található az adott időszakra. Általánosan elmondható tehát, hogy a *scraper* ezen korlátozás miatt általában némi veszteséggel dolgozik.

Az indeed.co.uk oldalról az adatgyűjtést az alkalmazás naponta végzi 2019 január 16-i kezdettel. Az *URL fromage* paraméterével szabályozható, hogy a szolgáltató *API* csak azokat a hirdetéseket adja vissza, melyek a megadott egész számban kifejezett napnál újabbak, mint látható ez a szám itt 1.

Az adatgyűjtés eszközéül a *Scrapy* keresőrobot-rendszert választottam, ami egy *Python*ban íródott, nyílt forráskódú projekt. A *Scrapy* egy olyan „integrált rendszer, mely egy ütemezőből, egy letöltést segítő modulból, egy – az adatfolyamot kontrolláló – központi motorból, illetve *spider*eknek nevezett egyedi osztályokból áll, melyeket a felhasználónak kell megírnia, és azt a logikát tartalmazzák, ami a válaszban kapott

dokumentumokat feldolgozza (*parsing*)" (Myers és McGuffee, 2015, p. 85). Tehát a keretrendszer biztosít minden infrastruktúrát az adatgyűjtéshez, a felhasználónak csak az „üzleti logikát" kell leírnia *CSS Selector*ok vagy *XPATH* segítségével a *spider* osztályokban, azaz hogy a legyűjtött dokumentumok mely elemei mit jelentenek, és hogy milyen formában szeretné azokat tárolni. Jelen probléma esetében definiálnom kellett egy JobPosting osztályt, melynek adattagjai megfelelnek az általam gyűjteni kívánt adatoknak, de alkalmazkodnak az állásportál által adott reprezentációhoz. Ezt követően *selectorok* segítségével a *spider* osztályban meg kellett adnom, hogy az egyes *HTML* oldalakon, mely *DOM* elemek, a JobPosting osztály mely adattagjának felelnek meg[18].

A Scrapy számtalan kimeneti formátumot támogat, mivel jelen esetben a legyűjtött álláshirdetések önmagukban teljes entitások, így elsődleges tárolásukra a *JSON Lines* formátumot választottam, ahol minden sor egy-egy *JSON* objektum, melyeket új sor karakterek szeparálnak (*newline-delimited JSON*). Az álláshirdetésekből legyűjtött adatok körét a 2. táblázat szemlélteti.

Azonosító	**Leírás**
title_result_page	A hirdetés címe a keresési eredmények között.
title_posting	A hirdetés címe a belső oldalon.
posting_id	Az álláshirdetés azonosítója az adott portálon, ami alapján a hirdetés később újra megtalálható.
Company	Az állást hirdető vállalat.
company_rating_value	Az állást hirdető vállalat opcionális értékelése.
company_rating_count	Opcionális adat, mely azt jelzi, hogy amennyiben az adott vállalat rendelkezik értékeléssel, az hány ember véleményét tükrözi.
crawling_date	A gyűjtés dátuma.
job_location	A pozíció földrajzi helye.
job_description	A hirdetés szöveges leírása.
posting_time	A hirdetés feladásának relatív ideje, nálunk maximum 1 nap.

2. táblázat: A scraping eszközzel gyűjtött adatok

A 2. táblázatból a hirdetés szöveges leírása legtöbbször *HTML* formában adott és rosszul strukturált, azonban ez az adattag az, mely feltételezésem alapján további

[18] A scraper forráskódja megtalálható a https://github.com/gneusch/JobPostingScraper címen elérhető GitHub repozitóriumban.

elemzéssel számos új információt adhat, mint például a pozíció szempontjából legfontosabb kompetenciák, a szükséges végzettség és tapasztalat (szenioritási szint), a kínált juttatások köre stb.

5.2 A tárolni kívánt adatok köre

Függetlenül a választott adattárolási megközelítéstől és technológiától, a későbbi elemzésekhez szükséges információk köre nem változik. A következőkben az információtartalmat ismertetem, amit az álláshirdetésekből kinyerni szeretnék, az implementációtól függetlenül, illetve az adatok közötti kapcsolatok szemléltetésére bemutatom a relációs sémát, amit a szükséges adatköröknek megfelelően alakítottam ki (12. ábra).

Az adatgyűjtésről szóló alfejezetben bemutatott 2. táblázat szemlélteti a *Scrapy* keretrendszerben létrehozott adatgyűjtő robot által generált kimenet tartalmát. Az 3. táblázatban azok az információk láthatók, melyeket az egyes álláshirdetésekből kinyerni remélünk és tárolni szeretnénk. A táblázat tartalmazza az adatok megnevezését, leírását, annak jelzését, hogy az adat opcionális-e, illetve azt, hogy az egyes elemek hogyan kapcsolódnak a *spider* által generált kimenethez.

3. táblázat: Az álláshirdetésekből tárolni kívánt adatkörök

Mező megnevezés (azonosító)	**Leírás**	**JSON objektum**	**Kötelező/Opcionális (K/O)**
pozíció[19] url (*url*)	Az álláshirdetésre mutató hivatkozás címe	posting_id	K
állásportál (*job_portal*)	A JSON Lines fájl nevében tárolt tulajdonság	-	K
pozíció megnevezése (*job_title*)	Az álláshirdetés címe az állásportálon	title_posting	K
pozíció leírása (*job_description*)	Az álláshirdetés leírása az állásportálon. A feldolgozás alapjául szolgáló szöveg.	job_description	K
pozíció típusa (*job_type*)	A pozíció leírása alapján[20]. Munkaidő	-	O

[19]Az álláshirdetés és pozíció kifejezéseket jelen kontextusban szinonimaként használom.

[20]Azt jelenti, hogy az adott információ valamilyen szövegfeldolgozási folyamat eredményeképpen áll elő.

Mező megnevezés (azonosító)	**Leírás**	**JSON objektum**	**Kötelező/Opcionális (K/O)**
	hossza, vagy teljes-, rész- stb. munkaidő.		
fizetés (*salary*)	A pozíció leírása alapján.	-	O
juttatások (*benefits*)	A pozíció leírása alapján.	-	O
foglalkoztatás típusa (*employment_type*)	Főállás, vállalkozó, konzultáns (külső cégen keresztül), gyakornok stb. A pozíció leírása alapján.	-	O
tapasztalat (*experience_level*)	Szenioritási szint. A pozíció leírása alapján.	-	O
aktivitás kezdete (*from_date*)	A meghirdetés időpontja.	posting_time	K
Aktivitás vége (*to_date*)	Utolsó aktív nap, vagy az aktuális, ha az álláshirdetés jelenleg aktív.	-	K
vállalat név (*legal_name*)	Vállalat neve az állásportálon, vagy null.	company	O
vállalat szektor (*sector*)	A pozíció leírásából, a cég neve alapján egyéb külső forrásból. (szinonima: *business_stream*)	-	O
vállalat profil (*profil*)	A pozíció leírásából, a cég neve alapján egyéb külső forrásból.	-	O
vállalat leírás (*description*)	A pozíció leírásából, a cég neve alapján egyéb külső forrásból.	-	O
vállalat értékelés (rating)	Az álláshirdetés megfelelő adata alapján.	company_rating_value	O
vállalat értékelők száma (*review_count*)	Az álláshirdetés megfelelő adata alapján.	company_rating_count	O
lokáció (*location*)	Lokáció az állásportálon, ha ez	job_location	K

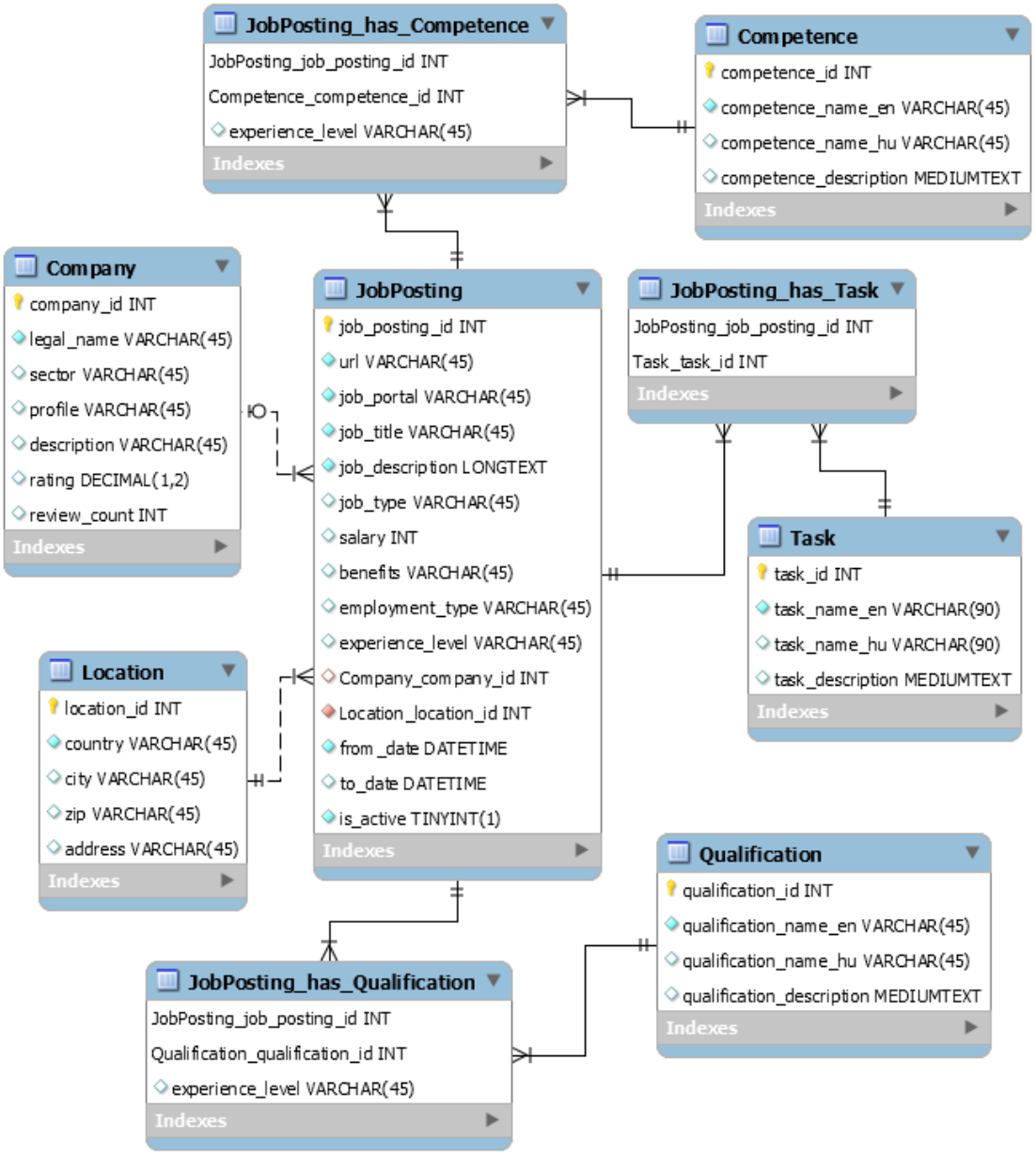
Mező megnevezés (azonosító)	Leírás	JSON objektum	Kötelező/Opcionális (K/O)
	üres, akkor a pozíció leírása, vagy a keresési szűrő alapján.		
feladatok (*task*)	Mapping táblán keresztül kapcsolt lista. A pozíció leírása vagy külső forrás alapján.	-	O
kompetenciák (*competence*)	Mapping táblán keresztül kapcsolt lista. A pozíció leírása vagy külső forrás alapján.	-	O
végzettségek (*qualification*)	Mapping táblán keresztül kapcsolt lista. A pozíció leírása vagy külső forrás alapján.	-	O

Kher (2016, 2017) bemutat egy hipotetikus online álláskereső portál adatbázis oldali támogatásához létrehozott sémát. Jelen dolgozatban felvázolt sémával ellentétben Kher lényegesen nagyobb hangsúlyt fektet munkájában a felhasználók (álláskeresők), illetve a cégek adatainak sokkal részletesebb tárolására, továbbá sok információt rögzít a hirdetések aktivitásával kapcsolatban. A 12. ábrán látható struktúra hangsúlyai ezzel szemben az álláshirdetésben leírt pozícióhoz kapcsolódó adatokon vannak.

A ténytáblában a közvetlenül a pozícióhoz tartozó, azt leíró adatok kaptak helyet, míg a dimenziós táblákban az állást meghirdető cégre, a helyszínre, a feladattartalomra, a szükséges kompetenciákra és végzettségekre vonatkozó információk találhatók. Mivel egyes dimenziók és az álláshirdetések között *n:n* kapcsolat áll fenn – egy pozícióhoz több kompetencia stb. is szükséges lehet, míg egy adott kompetencia stb. több pozíció esetében is megjelenhet igényként – így ezen dimenzionális- és a ténytáblákat kapcsolótáblák segítségével szükséges illeszteni.

A vállalat és a helyszín tekintetében azok részletes adatait szintén dimenzionális táblákban, míg az azokra mutató idegen kulcsokat a ténytáblában tároljuk, azonban ezek az álláshirdetés azonosításában nem játszanak szerepet. Mivel a vállalat nem

biztos, hogy explicit módon szerepel a hirdetésben, ezért ez az adat a ténytábla szempontjából opcionális, míg a munkavégzés helye esetében az egyetlen információ, ami biztos, hogy rendelkezésre áll, az a régióra, illetve a célországra vonatkozik, hiszen azt a kezdeti keresési feltételünk alapján tudjuk. Az 12. ábra jelmagyarázata a 1. mellékletben található.



12. ábra: Az adattárház lehetséges logikai adatmodellje

5.3 Adattárolás megfontolandó aspektusai

A gyűjtött adatok heterogén volta, és nagy mennyisége miatt a választott adattárolási megoldásnak a nyilvánvaló gyorsaság, olcsó tranzakciókezelés stb. alapelvárásokon túl számos egyéb követelménynek is eleget kell tennie. Ennek megfelelően a

következőkben sorra veszem azokat a szempontokat, melyek az adatok tárolására szolgáló eszközök összehasonlításának alapjául szolgálhatnak. Az architektúraválasztási szempontrendszert szekunder kutatás, illetve szakirodalmi áttekintés segítségével dolgoztam ki.

5.3.1 Technológiák összehasonlítása

Természetesen a klasszikus szempontok, mint a sebesség, megbízhatóság, skálázhatóság, a megoldás költségei vagy éppen az adattömörítés lehetőségei stb. is nagyon fontosak, így az egyes adattárolási megoldásokat ezek szerint a szempontok szerint is meg kell vizsgálni. Az alap szempontok közül a két talán legfontosabb; a megvalósíthatóság – azaz hogy igényel-e az implementáció speciális szakértelmet vagy addicionális erőfeszítést – és a támogatás. Például az SQL egy univerzálisan használt lekérdezőnyelv; támogatja-e azt az adott megoldás, vagy specifikus tudás illetve készségek szükségesek? Szükséges-e bármilyen speciális hardverelem a telepítéshez? Rendelkezésre áll-e megfelelő kereskedelmi vagy közösségi támogatás stb.?

5.3.1.1 Sebesség

A sebesség szempontjából az itt javasolt keretrendszerben a választott adatbázisnak elsősorban a lekérdezéseket kell tudnia minél hatékonyabban és gyorsabban kielégítenie. Az előzőek alapján láthattuk, hogy ebből a szempontból mind a *ROLAP*, mind a *NoSQL* rendszereknek vannak hátrányai.

A relációs modell esetében a komplex *join* műveletek nagyon költségesek. Ha rendelkezünk a megfelelő erőforrásokkal, akkor ezek a költségek a mérlegben, egyébként pedig a lekérdezések válaszidejében jelennek meg. Az adattavak esetében, az elosztott tárolás ellenére, a párhuzamosítható feldolgozás a hálózat elemei közötti szétosztásának köszönhetően hatékonyan és relatív gyorsan kinyerhető az információ. Ugyanakkor, ha az nincs előre feldolgozva, akkor minden egyedi lekérdezéskor le kell futtatni a kompetenciakifejezések feltárásához szükséges természetesnyelv-feldolgozást támogató- és gépi tanuló algoritmusokat, ami nem hatékony. Tehát ha az adatokat egy adattóba „borítva”, előfeldolgozás nélkül tároljuk, úgy például minden alkalommal, amikor egy bizonyos foglalkozáshoz kapcsolódóan szeretnénk képet kapni a szükséges kompetenciákról, illetve azok alakulásáról, akkor fel kellene dolgozni az összes, a felhasználó által kért időintervallumba eső álláshirdetést ahhoz, hogy a relevánsakat megtaláljuk; melyek esetében további feldolgozást igényelne a

kompetenciák kinyerése a szabadszöveges leírásokból. Ez, a komplex és költséges feldolgozási lépések miatt, még tetszőlegesen kis időintervallum lekérdezése esetében is rengeteg időbe és erőforrásba kerülne.

Az előzőekkel ellentétben, abban az esetben, ha a nyers adatokat legyűjtjük egy elosztott fájlrendszerre vagy egy dokumentumtárba, ami jól integrálható a feldolgozást végző eszközzel (például *MapReduce, Spark* stb.), majd rögtön ezután fel is dolgozzuk azokat, végül letároljuk az eredményeket a relációs adatmodellnek megfelelő struktúrába, és definiáljuk/frissítjük a szükséges indexeket, adatkockákat stb., a felhasználók lekérdezéseinek válaszideje elhanyagolható lesz. Ezen megoldás esetén a feldolgozás eredményét pont a kiindulási *NoSQL* adatbázis aggregátum-orientáltsága miatt hatékonyabb egy relációs adatbázisban tárolni. Az aggregátum-orientált adatbázisok ugyanis hatékonyságukat és erejüket pont abból merítik, hogy az elosztott architektúrán az adatokat az objektumok mentén tárolják. Ez azt jelenti, hogy ha egy aggregátum mentén kérdezünk le adatokat, azaz jelen esetben legtöbbször hirdetéseket kezelünk, a rendszer nagyon gyorsan és hatékonyan tud válaszolni, ismerve az egyes objektumok határait, illetve elhelyezkedésüket az elosztott klaszterben. De amennyiben az adatoknak már egy más jellegű reprezentációja érdekel minket, például az, hogy az egyes kompetenciák iránti kereslet hogyan változott az időben, át kell lépnünk az aggregátum határokat, ami a *NoSQL* adatbázisok esetében nem triviális, hiszen nem erre lettek optimalizálva. Természetesen technikailag a probléma megoldható, például egy, már említett *MapReduce* folyamatban, azonban ehhez arra lenne szükség, hogy a hálózaton keresztül a klaszter számos *node*ja működjön együtt, illetve a folyamat koordinálásának is további költsége van. Egy ilyen jellegű lekérdezés az előzőekkel ellentétben egy relációs adatbázisból triviális. Azaz a dokumentumtárak az adatokból különböző dimenziók mentén történő riportolásra (*slice and dice* stb.) és analitikák futtatására kisebb hatékonysággal alkalmasak (Sadalage és Fowler, 2012), ezért sebesség szempontjából optimálisabb a számos szükséges dimenzióadatot egy relációs modellben tárolnunk.

5.3.1.2 Skálázhatóság, megvalósíthatóság, támogatás és költségek

A skálázhatóság kérdésköréről az előzőekben esett szó, de összefoglalóan elmondható, hogy a *NoSQL* adatbázisok mind vertikálisan, mind horizontálisan jól skálázhatók, míg az *in-memory* adatbázisok egyetlen számítógép határai közé vannak szorítva. A két véglet között helyezkednek el a relációs modellt implementáló adattárházak. Az

utóbbiak horizontális kiterjesztése, a relációs adattárolás sajátosságai miatt nem triviális, de vannak gyártók jól működő megoldásokkal. Viszont cserébe ezek a termékek jellemzően rendkívül drágák. Ezzel ellentétben a *NoSQL* adatbázisok legtöbbször nyílt forráskódúak, vagy valamilyen fajta *freemium* modellben elérhetőek, így saját infrastruktúrára, a megfelelő szaktudás és hardver birtokában, ingyenesen feltelepíthetőek. Sok gyártó üzleti modellje a terméktámogatás köré épül. Továbbá számos *NoSQL* megoldás felhő szolgáltatóknál szoftver-mint-szolgáltatás (*SaaS*) formában is igénybe vehető, használatarányos fizetési feltételek mellett.

Megvalósíthatóság szempontjából érdemes kitérni a szükséges szaktudás kérdésére is. Az adattárházbevezetési-projekteket általában a gyártó, vagy külső tanácsadócég szakemberei támogatják. Rengeteg esettanulmány igazolja vissza, hogy erre általában szükség van, hiszen a bevezetés, a modellezés, az adatok tisztítása és betöltése, a folyamatok kiépítése stb. rengeteg szaktudást igényel, ami jellemzően a cégeknél nem áll rendelkezésre. Ugyanakkor a bevezetés után a rendszer karbantartása és üzemeltetése már talán kisebb feladat, hiszen a relációs *OLAP* rendszerek esetében is a kommunikáció *de facto* szabványos nyelve legtöbbször a strukturált lekérdezőnyelv, vagy *SQL,* ami általánosan is az egyik legelterjedtebb és legnépszerűbb programozási nyelv[21]. Emiatt viszonylag könnyű relációs adatbázisok üzemeltetéséhez értő szakembert találni a piacon. Támogatás szempontjából a *NoSQL* adatbázisok a nyílt forráskódnak, a kezdeti talán túlzó várakozásoknak, ugyanakkor a technológia érettségi szakaszában is bizonyított teljesítménynek és megbízhatóságnak köszönhetően óriási felhasználói közösséggel, illetve általában rendkívül átfogó és közérthető dokumentációval rendelkeznek. A gyártók egyes esetekben *SQL* interfészt is fejlesztettek termékeikhez, habár azok jellemzően inkább saját, az adatstruktúrának megfelelően tervezett lekérdezőnyelvet használnak. Továbbá minden nagyobb felhőszolgáltató kínál a legelterjedtebb típusú *NoSQL* adatbázisok közül pár kattintással, szinte azonnal üzembe helyezhető klasztert.

5.3.1.3 A CAP-tétel következményei

A 3.1.3 alfejezetben röviden ismertettem a CAP-tételt és utaltam rá, hogy az egyes adatbázismegoldások közötti választás során fontos figyelembe venni a közvetített sejtést, azaz hogy egy elosztott adatbázisrendszerben a konzisztencia, a rendelkezésre

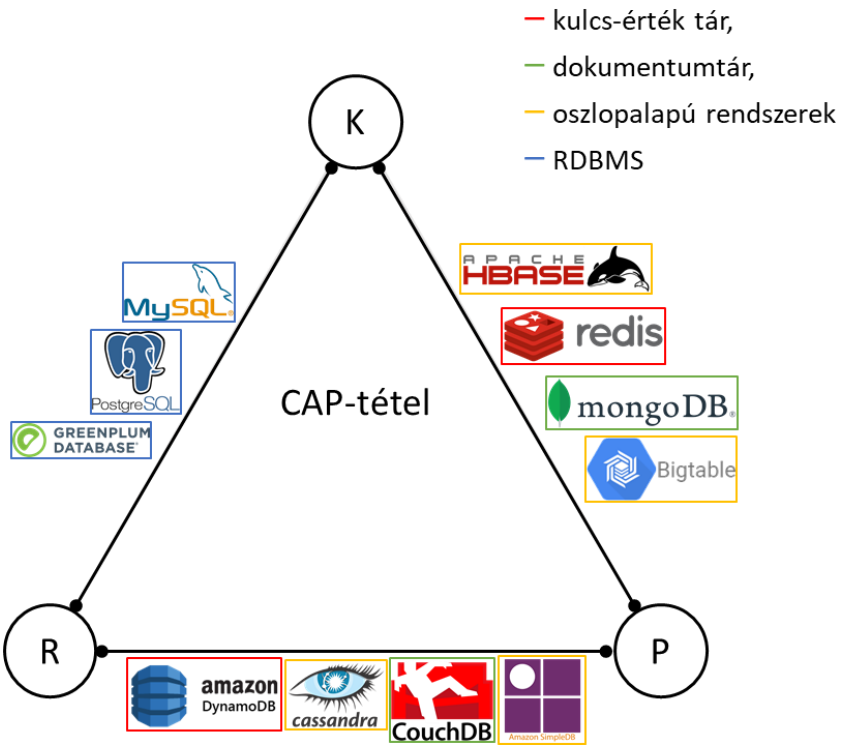
[21] A StackOverflow, fejlesztőket támogató weboldal 2020-as éves felmérése alapján a legkedveltebb programozási nyelvek között a 11. helyen végzett az *SQL* (“Stack Overflow Developer Survey 2020”, 2020).

állás és a partíciótolerancia tulajdonságai közül egyszerre maximum kettő biztosítható. Gajdos (2019) és Fox és Brewer (1999) alapján tehát három esetet különböztethetünk meg, melyek gyakorlati szempontból a következőket jelentik:

- *Konzisztencia – Rendelkezésre állás (KR)*: Azok a rendszerek, melyek nem partíciótoleránsak, csak korlátozottan skálázhatóak.
- *Partíciótolerancia – Rendelkezésre állás (PR)*: Azon rendszerek esetében, melyek nem képesek erős konzisztenciát biztosítani, előfordulhat például, hogy *A* felhasználó módosít egy adott értéket az adatbázisban, melyet *B* felhasználó röviddel ezután lekérdez. Ha ezt a lekérdezést egy olyan replika szolgálja ki, amely még nem kapta meg *A* felhasználó módosításait, akkor *B* még a módosítás előtti értéket kapja vissza. Ezek a rendszerek általában fokozatos (*eventual*) konzisztenciát biztosítanak, azaz azt garantálják, hogy „előbb-utóbb minden olvasás a legutóbbi írás értékét éri el" (Gajdos, 2019, p. 284).
- *Konzisztencia – Partíciótolerancia (KP)*: Végül azon elosztott rendszerek esetében, melyek nem tudnak magas rendelkezésre állást biztosítani, „hálózati partíció fellépése esetén azok [az] adategységek elérhetetlenné válnak, amelyekre […] a rendszer nem tudja biztosítani a műveletek atomi végrehajtását" (Gajdos, 2019, p. 281). Ha egy rendszer folyamatosan konzisztenciára törekszik, nem lehet mindig elérhető (Kleppmann, 2015).

Sadalage és Fowler (2012) ezt úgy fogalmazza meg, hogy minden elosztott adatbáziskezelő-rendszer életciklusa alatt fellép hálózati hiba (partíció). A rendszer kiválasztása során azt az (üzleti) döntést kell meghozni, hogy ebben az esetben mi a fontosabb; a rendelkezésre állás, vagy a konzisztencia? Későbbi cikkében Brewer szintén ezt a gondolatmenetet használja, illetve kifejti, hogy a modern értelemben vett „*CAP* cél a konzisztencia és a rendelkezésre állás maximalizálása, amennyire és amilyen kombinációban az az adott alkalmazás szempontjából értelmes" (Brewer, 2012). A modern rendszerek legtöbbje hálózati hiba esetére rendelkezik valamiféle működési és helyreállítási tervvel, de azért a legtöbb egyértelműen elhelyezhető a konzisztencia, partíciótolerancia és rendelkezésre állás két dimenziója mentén (lásd 13. ábra). Kutatók és szakemberek körében az elmúlt években megjelent az a nézet, hogy az egyes adatbázis-megoldások elhelyezése ebben a térben elavult, illetve hogy

a CAP-tétel segítségével nem lehet érvelni egyes rendszerek ellen vagy mellett, mivel az túlegyszerűsíti a problémát (Kleppmann, 2015).



13. ábra: Gyakoribb adatbázis termékek a CAP térben (saját szerkesztés Singh és Kumar (2019) és Khazaei és mtsai (2016) alapján)

Jelen dolgozat céljainak esetében – mivel a tárolni és feldolgozni kívánt adatok mennyisége miatt igényeinket egy idő után egyetlen hardver eszköz nem fogja tudni kiszolgálni – a legfontosabb, hogy a választott megoldás megfelelően skálázható legyen. Mivel a rendszerben nem folyamatosan, valós időben változó (pl. tőzsdei- vagy szenzoradatok), vagy az operatív működés szempontjából kritikus (pl. üzleti tranzakciók) adatokkal kívánunk dolgozni, az adatgyűjtés és az ETL folyamat is ütemezve történik, továbbá egyelőre nincs olyan felhasználási eset, melyben a végfelhasználó az adatokat módosítaná, ezért az erős konzisztencia nem elsőrendű szempont. Fontos azonban a magas rendelkezésre állás garantálása, az inputadat-feldolgozás folyamatosságának biztosítása, illetve a felhasználók bizalmának megnyerése és megtartása érdekében. Néhány konkrét, gyakrabban használt termék *CAP* (*KRP*) terében való elhelyezkedését mutatja be a 13. ábra Singh és Kumar (2019) és Khazaei és mtsai (2016) alapján.

5.3.1.4 Az adatok struktúrájából eredő igények

Az állásajánlatok adatainak egy része repetitív jellegű és jól strukturált, oly módon, ahogy azt az állásportál biztosítja, és olyan mértékben, amennyire az a *scraping*

folyamat során megőrizhető. De ugyanakkor az adatok legfontosabb köre, az álláshirdetések leírása, ami a legtöbb információt tartalmazza, sem nem strukturált, sem nem repetitív, kettő közülük csak véletlenül hasonlíthat egymásra felépítésében vagy tartalmilag. A pozícióleírásokban vannak persze kontextuális adatok – melyek meglétére jelentős mértékben támaszkodunk is –, például a foglalkozás, az elvárt tapasztalati szint, a szükséges kompetenciák köre és végzettségek stb., de a legtöbb esetben rejtettek, amiket így csak egyedi, kifejezetten erre a problémára szabott módon lehet beazonosítani a szöveges adathalmazban. Amennyiben *NoSQL* megoldásokat használunk, a nem repetitív adatokban rejlő információ feltárására általában az adatbáziskezelő rendszeren kívül kerül sor (Inmon és Linstedt, 2014).

Tehát a bemeneti oldalon egy félig strukturált, hibrid jellegű adathalmazzal dolgozunk, ami technikailag *JSON* formában áll rendelkezésre. Ezen objektumok tárolására és kezelésére egy aggregátum-orientált dokumentumtár lenne a legalkalmasabb (Sadalage és Fowler, 2012). Annál is inkább, hiszen a hirdetések struktúrája azok forrásától függően, de akár az időben is változhat. A dokumentumtárak megengedik, hogy az adatoknak legyen egy kvázi struktúrája, sémája, de nem kényszerítik ki azt, azaz olyan adatkörök is letárolhatók, amelyeknek kategóriája előre nem definiált, illetve a lekérdezőnyelvek is fel lettek készítve ennek támogatására. Tehát egy dokumentumtár használata esetében nem kényszerülünk minden feldolgozást azonnal elvégezni, illetve a sémába nem illő adatok eldobására sincs szükség. Üzleti, használhatósági szempontból azonban a legfontosabb, hogy a végfelhasználóink egyedi lekérdezéseire gyors és pontos választ tudjunk adni. Főleg a valósidejűség garantálása érdekében kritikus, hogy a lekérdezéseket és elemzéseket egy jól strukturált és „letisztázott” adathalmazból szolgáljuk ki. Ez utóbbi egy erős érv amellett, hogy érdemes a hirdetéseket előzetesen feldolgozni, és a beazonosítható információkat (például a kapcsolódó kompetenciákat, a foglalkozást stb.) pedig strukturált formában tárolni.

Az állásajánlatokon kívül – melyek önmagukban számtalan portálról, sokféle heterogén struktúrában állhatnak elő – az előzőekben részletezetteknek megfelelően, egyéb külső forrásból származó adatokat is gyűjteni fogunk. Ezeket, bár valószínűleg az elemzés idejében is elérhetőek lennének eredeti külső forrásukból, azért szeretném elmenteni, hogy a későbbi feldolgozást gyorsabbá, gördülékenyebbé tegyem. Ilyen dimenziók például az ESCO ontológia kapcsolódó pillérei, az O*NET kapcsolódó

koncepciói és egyéb egyszerű nomenklatúrák, mint az ISCO, FEOR stb. A kutatás későbbi szakaszaiban gyűjteni szeretnénk ezeken felül iparági, regionális, technológiai trendekre vagy adott munkaerőpiaci szegmensre vonatkozó adatokat is. A választott adattárolási megoldásnak tehát támogatnia kell az adatok struktúrájának és formájának e sokféleségét, ideális esetben natív módon, mindenféle specifikus, egyedi implementáció igénye nélkül.

5.3.1.5 Adattisztítás, adatbetöltés, információfeltárás

Az előbb részletesen felsorolt adatforrásokból az adatokat nem elég változatlan formában lementeni. A választott adattárolási megközelítés függvényében – vagy egy relációs sémában való tárolás előtt, vagy közvetlenül az elemzések végrehajtásakor – szükséges azokat a megfelelő üzleti objektumokkal megfeleltetni (*mappelni*). Ezt egy „klasszikus" adattárház megoldás esetében valamilyen *ETL* folyamat során szokás megtenni. Jellemzően ebben a feldolgozási lépésben az adatokkal kapcsolatos minőségi problémákat is kezelni kell, mielőtt betöltenénk azokat a relációs sémába. A megfelelő alkalmazás egyik kiválasztási szempontja lehet tehát a megfelelő *ETL* támogatás rendelkezésre állása. Az erre a feladatra kifejlesztett, piacon elérhető eszközök azonban jellemzően általános célokat szolgálnak, tehát a konkrét feladatnak megfelelően szükséges azokat testre szabni, ami nem feltétlenül triviális, hasonlóan az *ETL* alkalmazás zöldmezős, belső fejlesztéséhez, ami szintén hosszadalmas és fáradságos feladat lehet. Az alkalmazások karbantartására, továbbfejlesztésére, az adatok vagy az adatbázis esetleges változásaihoz adaptálására stb. szintén szükséges erőforrásokat allokálni. Amennyiben egy „*big data*" megoldást vizsgálunk – ahol az adatokat jellemzően nyers formájukban tároljuk – ugyanezt az erőfeszítést az elemzések idejében kell megtennünk, hiszen ebben az esetben a logika az alkalmazott *MapReduce*, *Spark* vagy más hasonló implementációba kell, hogy kerüljön.

A nyers adatokat jelen feladat esetében nem egyszerűen tisztítani szükséges, hanem az információk feltárása érdekében szövegbányászati megoldásokat és gépi tanulási algoritmusokat is alkalmazni kell. Az ehhez szükséges alkalmazáslogika pedig annyira egyedi a probléma szempontjából, hogy elkerülhetetlenül „belső" fejlesztést igényel. A kérdés önmagában tehát – hogy ezt a logikát az adatbetöltés előtt, vagy után alkalmazzuk – marginális, hiszen fejlesztésre mindkét esetben szükség van. Az egyetlen gyakorlati, lényeges különbség, hogy egy adattárház-megoldás választása

esetén szükséges egy előkészítő (*staging*) tár kialakítása is, ahol a legyűjtött álláshirdetések a feldolgozás előtt átmenetileg tárolhatóak.

5.3.1.6 Idősor-adatbázisok használhatósága a keretrendszerben

Mint azt a problémafelvetésben kifejtettem, a dolgozatban felvázolt koncepció alapján később megvalósítani kívánt keretrendszer egyik legfontosabb célja az lesz, hogy a tanterveket, illetve oktatási stratégiákat kidolgozó szakemberek adatvezérelt döntéseit támogassa azzal, hogy segítségével rálátást kapnak a munkaerőpiacon keresett kompetenciák időbeli alakulásáról. Mikor a kutatási kérdéssel elkezdtem foglalkozni, teljesen nyilvánvalónak gondoltam, hogy egy ilyen problémára legegyszerűbben egy idősor-adatbázis segítségével lehet megfelelő választ adni, hiszen azok pont arra lettek optimalizálva, hogy nagy mennyiségű, időbélyeggel ellátott, ismétlődő adatot tároljanak és kiszolgálják az azokat elemző eszközöket (3.1.2 alfejezet).

Ezeket az adatbázisrendszereket azonban elsősorban tetszőlegesen rövid periódusonként ismétlődő adatpontok (mérések) eredményeinek kezelésére használják, ezért is rendkívül elterjedtek például a nagyfrekvenciás-kereskedés, a gyártásautomatizáció vagy éppen az *IOT* eszközrendszerek támogatásában. Jelen kutatás alapján elkészítendő keretrendszerben azonban, ahogy az a 12. ábra alapján látható, egy relatíve nagy komplexitású struktúra kezelését szeretném megvalósítani, nem egy-egy jól meghatározható mérés eredményét tárolni és feldolgozni. A kifejezetten idősor-adatbázisok nincsenek felkészítve ilyen komplex adatstruktúrák kezelésére, azok általában egy mérést, egy időbélyeget és a méréshez kapcsolódó metaadatok halmazát fogadják, sokszor specializált formában, mint például a *DB-Engines*[22] független rangsora alapján 2021 januárjában legnépszerűbb *InfluxDB* (Naqvi *et al.*, 2017).

A dolgozatban felvázolt probléma esetében továbbá a „mérések” viszonylag ritkán, naponta ismétlődnek, azaz nincs annyi mérési pont, ami miatt indokolt lenne az idődimenzió alapján indexeket létrehozni és fenntartani, illetve egy adott mérési pillanathoz több megfigyelés is kapcsolódik, hiszen egy adott napon több ezer álláshirdetést is legyűjthet a *scraper* a megfigyelt oldalakról. Tehát az idősor-adatbázisok által kínált előnyöket a javasolt rendszerben nem tudnánk kihasználni,

[22] https://db-engines.com/en/ranking/time+series+dbms

illetve a szükséges idősoros elemzéseket akár egy relációs adatbáziskezelő-rendszer is hatékonyan támogatni tudja.

5.3.1.7 Elemzések és kimutatások

Mivel az adatok tárolásának célja az, hogy végül azok alapján olyan információkhoz jussunk, melyekkel aztán az érintetteknek a problémafelvetésben jelzett céljai támogathatók; az implementáció megválasztásakor érdemes megvizsgálni azt is, hogy az egyes adattárolási megoldások kínálnak-e valamilyen megoldást, támogatást az elemzések elvégésére és az eredmények vizuális reprezentációjára. Kínál-e egy adott termék beépített analitikai eszközöket például statisztikai modellek, rendezés és ad-hoc hierarchiák, multidimenzionális megjelenítés, diagramok stb. Általánosságban elmondható, hogy mindegyik nagyobb adatbázis rendszerhez (legyen az relációs vagy *NoSQL*) rendelkezésre áll a gyártó vagy harmadik fél által fejlesztett elemzési megoldás, illetve számos felhőszolgáltató is biztosít ilyen termékeket. Továbbá elérhető szinte minden programozási nyelvhez, számtalan vizualizációs és elemzést segítő programkönyvtár (például *JavaScript*hez a D3.js[23] stb.), melyek jelentősen megkönnyítik és meggyorsítják az adatok elemzését. Azonban ezeknek a lehetőségeknek a feltárása és elemzése túlmutat jelen dolgozat keretein.

Általánosságban a teljes kutatás és annak esetleges későbbi alkalmazása, illetve kommercializációja szempontjából fontos vizsgálandó szempont, hogy az egyes megoldások milyen mértékben támogatják a teljes folyamat automatizációját, az adatok legyűjtésétől és tárolásától kezdve az analitikai folyamatokon át, az elemzések elkészítéséig.

5.4 A javasolt adattárolási architektúra

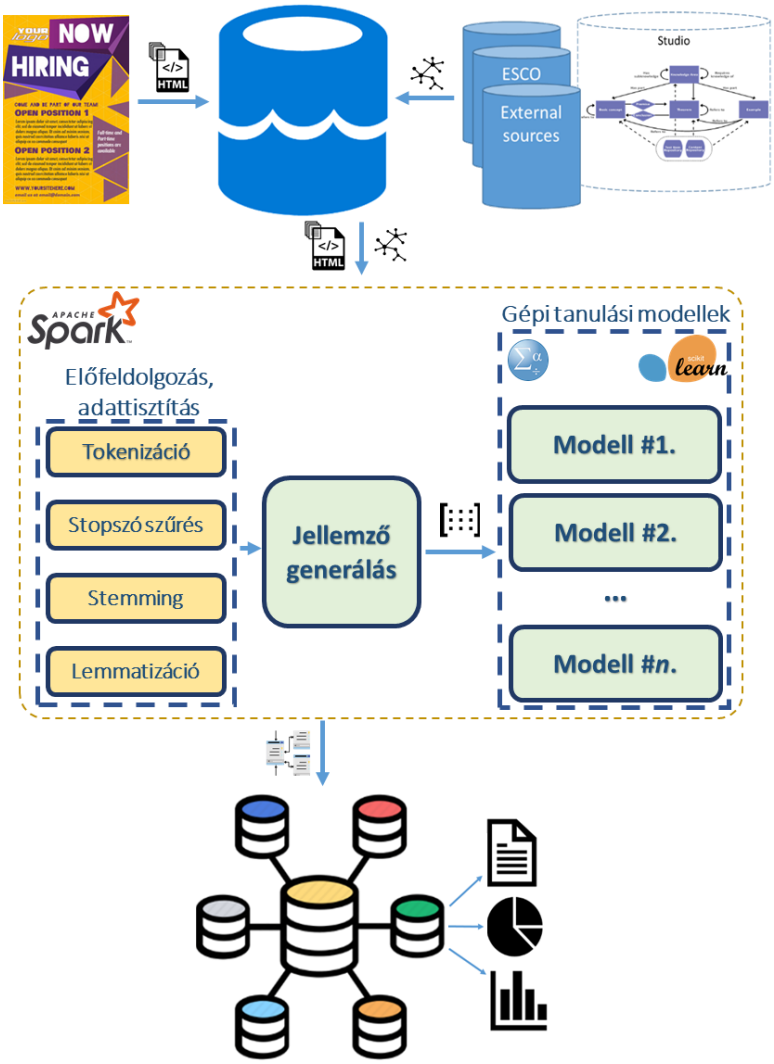
Összefoglalóan tehát a legfontosabb kiválasztási és értékelési szempontok a következők.

- A *scraping* folyamat eredményeként az adatok *JSON* formában állnak rendelkezésre, egyes attribútumok mentén strukturálva.
- Számos fontos, információértékkel bíró tartalmat azonban az álláshirdetések leírásaiból további feldolgozással szükséges kinyerni. Ezen feladat során mindenképpen szükséges az adatok tárolása egy előkészítő tárban.

[23] https://d3js.org/

- Az álláshirdetéseket, azok leírását feldolgozatlan formában továbbra is tárolni kívánom, hogy szükség esetén a jövőben is rendelkezésre álljanak. Továbbá az egyéb külső rendszerek felhasznált adatait is szeretném, hogy a feldolgozás egyszerűsítése és gyorsítása, illetve a folyamatosság biztosítása érdekében az adatbázisban is rendelkezésre álljanak.
- A rendszer sebességének szempontjából támasztott legfontosabb követelmény, hogy a felhasználók lekérdezései valós időben megválaszolhatóak legyenek, ezért szükséges a kompetenciák, foglalkozások stb. előzetes feltárása és tárolása, mely utóbbira jelen esetben egy relációs adatbázis alkalmasabb.
- Fontos, hogy a rendszer jól skálázható legyen és magas rendelkezésre állást garantáljon. Az erős konzisztencia azonban nem elvárás.

Az előzőek fényében, figyelembe véve az ismertetett technológiák előnyeit és hátrányait, illetve a legfontosabb célt, a felhasználók információigényének lehető leggyorsabb kiszolgálását, egy hibrid architektúrára épülő megoldást hoznék létre, melynek fő alkotóelemeit a 14. ábra szemlélteti. Ezt a fajta hibrid megközelítést, amikor az adott probléma fényében a legmegfelelőbb, akár különböző megoldásokat egyszerre használó adatbázis-architektúrát építünk fel, Sadalage és Fowler (2012) *polyglot* perzisztencia néven hivatkozzák.



14. ábra: Hibrid tárolási megoldás sematikus architektúrája

Ebben az architektúrában az összegyűjtött álláshirdetések és a külső rendszerek adatai először egy dokumentumtárba kerülnek. Nincs szükség tehát a *scraper* által gyűjtött adatok előzetes átalakítására, azonnal (a letöltés pillanatában) letárolhatók abban a részlegesen strukturált formában, amibe a *scraping* folyamat során kerülnek. Az előkészítő (*staging*) tár biztosításán kívül további előnye ennek a kialakításnak, hogy nagymértékű rugalmasság jellemzi, azaz ha például a kapott adatok struktúrája változik, vagy ha új adatköröket kívánunk bevonni a feldolgozásba stb. a keretrendszer ezen pontján nem szükséges változásokat eszközölnünk. Megőrizhetjük itt továbbá a külső rendszerekből származó- és az olyan adatokat is, melyeket jelenleg nem kívánunk használni, vagy a létrehozandó sémába nem illeszkednek, de adott esetben később hasznosak lehetnek.

A dokumentumtár azonban nem alkalmas arra, hogy a lehetséges felhasználói igényeket a lehető leggyorsabban kielégítsük, többek között azért sem, mert

architekturálisan aggregátumok kezelésére, és nem az adatok különböző dimenziók mentén való szeletelésére és forgatására vannak felkészítve, illetve a szöveges formában tárolt hirdetésleírások feldolgozása hosszadalmas és költséges, így érdemes annak eredményét eltárolni.

Ebből kifolyólag egy „integrációs" rétegben, az adattóra épülő *Spark* alkalmazásban tervezem a hirdetésleírások nyelvi feldolgozását, és gépi tanulási algoritmusok segítségével a szükséges információ kinyerését megvalósítani. Azért előnyös az architektúra ilyen kialakítása, mert további rugalmasságot biztosít. Az adatfeldolgozás így bármikor elvégezhető, illetve adott esetben újra végrehajtható, amennyiben a felhasznált adatok köre (például egy új kompetenciaszótárt vonunk be a folyamatba, vagy a kompetenciaként beazonosított kifejezéseket csatoljuk vissza), vagy a modellek változnak. Nemcsak az információk kinyeréséhez használt algoritmus változtatható így könnyűszerrel, de a felhasznált implementáció is. A dolgozat későbbi részében ismertetett módon, a gépi tanulási modelljeimet például két különböző eszközben, az *IBM SPSS Statistics*ben és a *Scikit-learn* programcsomagban építettem fel és teszteltem (6.1.4 és 7.1.1.2 alfejezetek). Mind a két kapott modell könnyűszerrel integrálható egy elosztott dokumentumtár adatain dolgozó párhuzamosított *Spark* alkalmazásba.

Ezután a feltárt információk, immár strukturált formájukban kerülhetnek a 12. ábrán bemutatott relációs sémába egyszerű *SQL insert* és *update* utasítások segítségével. Azonban a legnagyobb méretű adattagot, a pozíció leírását (*job_description*), mivel a javasolt *NoSQL* adatbázisban hosszútávon megőrizni kívánom, a redundáns tárolást és a relációs adattárház horizontális skálázásának szükségességét elkerülendő, a felvázolt sémából kihagynám (ellentétben a 12. ábrával).

Ebben az architektúrában tehát biztosítható, hogy az egyes lekérdezések nagyon gyorsan, csak *SQL* utasításokra építve kiszolgálhatóak legyenek, ugyanakkor a nyers dokumentumok tovább tárolhatóak a dokumentumtárban. Így az is biztosított, hogy ha a jövőben változtatni kell a feldolgozási folyamaton, vagy a modelleken, vagy új információra van szükség a már archivált hirdetésleírásokból, minden adat rendelkezésre álljon.

5.5 Összefoglalás és további kutatási lépések

Az első kutatási kérdés esetében a cél az adattárolási architektúra kiválasztását megalapozó szempontrendszer-, majd segítségével a legmegfelelőbb architektúramodell kidolgozása volt. Kutatásomnak ez a szakasza feltáró módszertant használt, és célja a megvalósíthatóság vizsgálata és a későbbi kutatás megalapozása volt.

Egy kisebb szoftverfejlesztési projekt keretében megvizsgáltam az adatgyűjtéshez elérhető eszközök kínálatát, és a *Scrapy* keresőrobot-rendszert választottam, ami egy Pythonban íródott, nyílt forráskódú alkalmazás, amely biztosít minden infrastruktúrát az feladathoz.

Szintén az első kutatási kérdés vizsgálata során meghatároztam a tárolni kívánt adatok körét, melyekre a végső célok eléréséhez elengedhetetlenül szükség van, illetve felvázoltam egy lehetséges relációs sémát, amely egy relációs adattárolási megoldás választása esetén implementálható.

Az architektúraválasztási szempontrendszert szekunder kutatás, illetve szakirodalmi áttekintés segítségével dolgoztam ki. A legfontosabb megvizsgált szempontok között helyet kaptak klasszikus tulajdonságok, mint a sebesség, megbízhatóság, skálázhatóság, a megoldás költségei, a megvalósíthatóság és az elérhető támogatás. Ezeken kívül a megoldásokat összehasonlítottam a CAP-tétel következményei, az adatok struktúrájának és rendelkezésre állásának sajátosságaiból eredő igények alapján is. Megvizsgáltam továbbá az idősor-adatbázisok alkalmazhatóságának lehetőségeit is. Összefoglalóan a disszertációhoz kapcsolódó munka során elkezdtem az automatizált adatgyűjtést, kidolgoztam az adatoknak egy lehetséges modelljét, továbbá a kiválasztási szempontok részletes vizsgálatával ajánlást tettem a megvalósításra javasolt hibrid, *polyglot* adattárolási architektúrára. A kutatás következő lépése a konkrét termékek és az implementáció környezetének kiválasztása, illetve maga a megvalósítás lesz, amennyiben ennek anyagi és egyéb erőforrás-szükségletei megteremthetőek.

6 A hirdetésekben explicit megjelenő kompetenciaelemek beazonosítása

Jelen fejezetben olyan módszereket és kísérleteket ismertetek, melyekkel második kutatási kérdésem megválaszolására törekedtem, azaz arra, hogy az álláshirdetések szövegében explicit megjelenő kompetenciákat kibányásszam. A releváns készség és tudáselemek feltárásának legegyszerűbb, legkézenfekvőbb megoldása azok manuális beazonosítása. Ennek a módszernek az előnye, hogy az elérhető pontosságot csak a besorolást végző szakember hozzáértése és figyelme befolyásolja, hátránya, hogy rendkívül idő és energiaigényes művelet.

6.1.1 Út a szótárral támogatott kompetenciakereséshez

Egy szofisztikáltabb és viszonylag intuitív megoldás, amely segíthet beazonosítanunk a minőségi kifejezéseket, ha a korpuszban előforduló szóenneseket annak megfelelően ábrázoljuk, hogy szerepelnek-e egy adott dokumentumban vagy sem. Ilyen módon minden dokumentumhoz egy-egy, a szóennesek számának megfelelő dimenziószámú vektort kapunk, amely minden pozícióban 1-et értéket vesz fel, amennyiben az adott kifejezés szerepel a dokumentumban, és 0-t amennyiben nem. A szimpla tartalmazás helyett a vektorokban a gyakoriságot is szerepeltethetjük. Ezzel a módszerrel a gyakoribb kifejezések könnyen beazonosíthatók.

Annak vizsgálatára, hogy ez a módszer mennyire segít a minőségi kifejezések (*keyphrase candidate*) feltárásában, melyek aztán a kompetenciák beazonosítását végző szakember munkáját megalapozhatják, kísérleteket végeztem két egymástól távol eső, véletlenszerűen kiválasztott hónapban, 2019 márciusában és októberében begyűjtött álláshirdetéseken. Kezdeti vizsgálataimat, a komplexitás csökkentésének érdekében a trigramokra korlátoztam, azzal a feltevéssel élve, hogy a legalább 3 szót tartalmazó kifejezések elég hosszúak ahhoz, hogy beazonosíthassanak vagy tartalmazhassanak komplexebb kompetenciákat is, de még nem annyira hosszúak, hogy túlságosan zajos legyen az eredmény (az optimális n-gram hossz megállapításának kérdésével a 6.1.2.2-es alfejezetben bővebben foglalkozom). A kísérlethez az álláshirdetések szövegéből a stopszavakat eltávolítottam. Mivel az így létrejött mátrixok mindkét hónapra több, mint húszezer sort és hárommillió oszlopot tartalmaztak (az oszlopokban a trigramokkal), így annak vizsgálatára fókuszáltam, hogy a leggyakoribb 1000 kifejezés mennyire jelöl az IT szektorban releváns

kompetenciákat. A vizsgálat további folytatását is annak eredményétől tettem függővé, hogy a legnagyobb gyakoriságú kifejezések között milyen információkat találok, tekintve a szükséges manuális energiabefektetés nagyságát.

A kísérlet eredménye azt mutatta, hogy még ez – a leggyakoribb ezer szóhármast tartalmazó halmaz – is rendkívüli mértékben tartalmazott zajnak tekinthető elemeket. A zajon kívül – az álláshirdetések szöveg-előfeldolgozásának hiányában – rengeteg, a munka jellegére, a tapasztalati elvárásokra, illetve a munkáltatóra, annak erkölcsi viszonyulásaira és értékpreferenciáira, elvárásaira és a kínált előnyökre utaló kifejezést is kaptam, mint például „*full time job*", „*3 years experience*", „*without regard race*", „*private medical insurance*" stb. Ez az eredmény, önmagában vizsgálva, megerősíteni látszik a 2. fejezetben idézett Nasir et al. által leírt tapasztalatot, miszerint a vizsgált hirdetések inkább hangsúlyozzák a kínált pozíció sajátosságait, mint az elvárt kompetenciákat (Nasir *et al.*, 2020).

A 2019 márciusi eredmények halmazában vizsgált 1000 leggyakoribb kifejezés 6,4%-át fogadtam el kompetenciaként, vagy olyan elemként, ami nagy biztonsággal jelzi az igényt egy adott kompetenciára. De ezek a kifejezések leggyakrabban általánosabb jellegűek, puha készségek keresletére utalnak, és nem IT terület specifikusak. Például a „*problem solving skills*" a hirdetések 6%-ban jelent meg, vagy a „*time management skills*" (3%). A listában viszonylag előkelő helyen szerepeltek a kommunikációs készségekre vonatkozó kifejezések, például az „*excellent communication skills*" (8%), vagy „*written verbal communication*" (4%), ahogy az jól látszik a 15. ábrán. A hirdetések másfél százalékában jelent meg a végzettségi elvárást jelző „*degree computer science*" kifejezés.

15. ábra: Trigramokban előforduló kompetenciaként elfogadott elemek

A kísérletről összefoglalóan elmondható, hogy viszonylag nagy számítási igény és manuális munkabefektetés mellett, relatív kevés információt szolgáltatott az elemzett hirdetésekben keresett IT kompetenciákról. Az egyszerű frekvencián alapuló rendezése az előforduló kifejezéseknek nem sok segítséget jelent az általunk keresett kompetenciák beazonosításában. A feltárt IT készségre mutató kifejezések között több a Microsoft Office alkalmazásokhoz, illetve a Windows operációs rendszerhez kapcsolódott, illetve megjelent még a „*software development lifecycle*” fogalma is.

Mindenképpen meg kell említeni Luhn modelljét melyet leggyakrabban Zipf törvényének kvázi következményeként említenek. Zipf megfigyelése alapján elmondható, hogy ha egy korpusz szavait gyakoriságuk alapján csökkenő sorrendbe rendezzük, akkor negatív kitevőjű hatványfüggvénnyel közelíthető kapcsolat fedezhető fel egy adott szó rangsorban elfoglalt helye és gyakorisága között (Bíró,

1998), „melynek kitevője „-1”-hez közelít” (Bíró, 1997). Nagyon egyszerűen megfogalmazva; a rangsorban első szó körülbelül kétszer gyakoribb, mint a második, és körülbelül háromszor gyakoribb, mint a harmadik szó stb. Zipf munkájára építve Luhn azt találta, hogy amennyiben a gyakorisági rangsorra egy normál eloszlású görbét illesztünk, akkor a legnagyobb megkülönböztető erővel rendelkező szavak, melyek a szöveget legjobban jellemzik, a haranggörbe maximumhelye körül, a felső és az alsó kvartilis között összpontosulnak (Cummins és O'riordan, 2005; Hoeber és Liu, 2011). Jelen esetben azonban az eloszlás erősen jobbra elnyúló ($\beta_1 = 67$) és rengeteg eset szerepel a megfigyelések között azonos gyakorisággal, a 90%-os percentilis 3, a felső kvartilis 2 és a medián már 1. Mivel milliós nagyságrendben esnek megfigyelések ezekbe a kategóriákba, ezért a Luhn modellt nem lehet eredményesen alkalmazni.

Megvizsgálható azonban, hogy a gyakorisági kategóriákon belül lehet-e valamilyen másodlagos szempont szerinti sorrendet felállítani az elemek között, amely segíthet tovább finomítani az egyes kifejezések fontosságának megítélését.

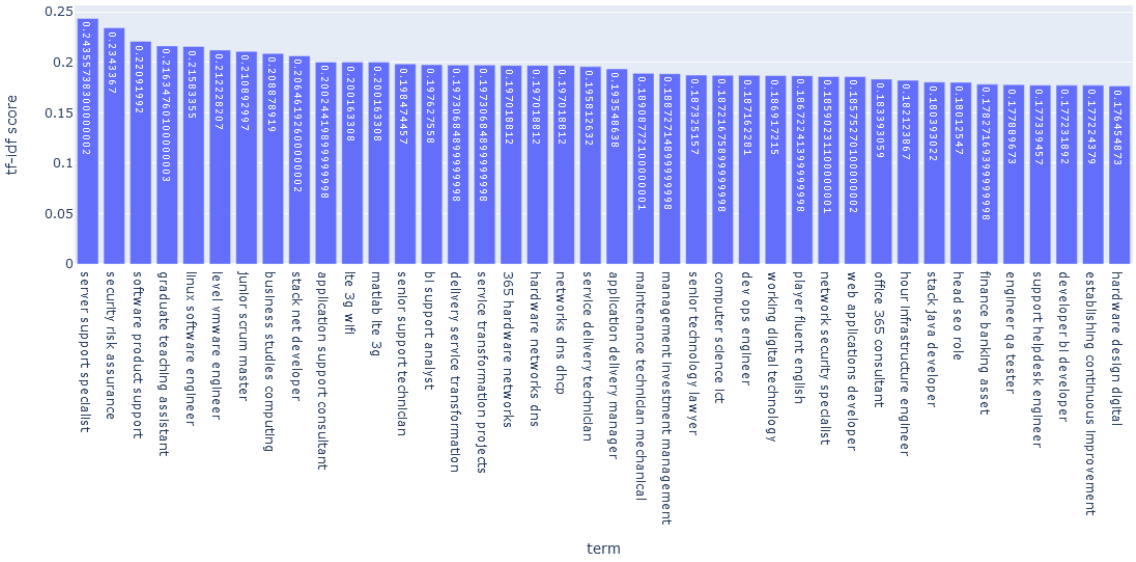
6.1.1.1 Az kifejezésgyakoriságon alapuló modell javításának lehetőségei

A Manning és munkatársai (2009) alapján tárgyalt *tf* normalizálási technikák nem változtatják meg a rangsort, így azok alkalmazásával az előzőekhez hasonló eredményeket kapnánk. Azonban az információkinyerés szakirodalmának jelen dolgozat 3.3.2 alfejezetében tárgyalt eredményei és megfontolásai alapján intuitív módon adódik a feltevés, hogy a kifejezésgyakoriság használatából adódó problémák egy részére megoldást nyújthat, ha egy-egy kifejezés súlyának megállapításához figyelembe vesszük az inverz dokumentumfrekvencia (*idf*) értékét is.

Ötvözve tehát az előzőekben leírtakat, a második kísérletben kiszámoltam a kifejezésgyakoriság *df* értéke mellett az adott kifejezés *tf-idf* értékét is[24]. Az adatokat gyakoriság (*df*), illetve a gyakorisági kategóriákon belül *tf-idf* értékek alapján rendeztem, majd a Luhn modell által javasoltaknak megfelelően, az adatokat leszűrtem. Egy ezer megfigyelést tartalmazó mintát vizsgáltam meg a teljes sokaság azon részéről – a 9. decilis környékéről –, ahol feltételezhetően nagyobb megkülönböztető erővel rendelkező kifejezések összpontosulhatnak.

[24] A *tf-idf* kalkulációhoz a *Scikit-learn* programcsomag *Feature extraction* modulját használtam (Pedregosa *et al.*, 2011)

A szintén a 2019 márciusi eredményekből vett mintában a kifejezések 40 százaléka (402 elem) fogadható el részben vagy egészben a kiindulási feladat szempontjából információt hordozónak. Az elfogadhatónak tartott kifejezések egy része foglalkozási kategóriát azonosít, mint például „*linux software engineer*” vagy „*dev ops engineer*”, akár szenioritási szint jelzésével kiegészülve, mint a „*senior technology lawyer*” vagy „*junior scrum master*”. A szóennesek egy másik része felsorolásos formában több területet, tudáselemet vagy készséget jelölt meg, mint „*iaas paas arm*” vagy „*aws docker linux”*, illetve szintén előfordultak olyanok, melyek csak részben jelöltek valamilyen ismeretet „*management investment management*” vagy éppen az ismeretnek csak egy részét tartalmazták „*stack net developer*”, amiből azonban könnyen következtetni lehet a teljes kifejezésre, ami a példa esetében „*full stack .net developer*”. A mintában kompetenciaként elfogadott elemek közül az első 40 látható a 16. ábrán[25].



16. ábra: A kifejezésgyakoriság 9. decilisénél található legnagyobb tf-idf értékkel rendelkező kompetenciaként elfogadott elemek

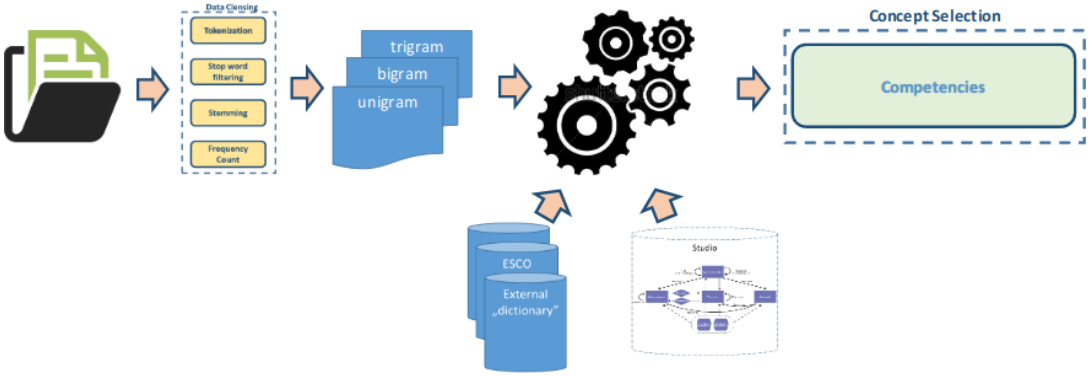
A kísérletről összességében elmondható, hogy magasabb információtartalmú, relevánsabb eredményeket kaptunk, mintha a leggyakoribb kifejezéseket önmagukban vizsgálnánk azáltal, hogy az adathalmazt az inverz dokumentumfrekvencia értéke alapján tovább rendeztük, illetve a korpusz specifikus stopszónak tekinthető elemeket Luhn törvényére hivatkozva eldobtuk. Ugyanakkor a minta egy részének eldobásával információt is veszítünk. Elmondható továbbá, hogy az eredmények teljes halmazának

[25] A teljes lista megtalálható: https://raw.githubusercontent.com/gneusch/phd_results/main/tf_idf/3_freq_tf_idf.csv

kiértékelése és feldolgozása továbbra is jelentős manuális munkabefektetést igényelne a rendszert használó, tantervfejlesztésben érdekelt felhasználótól, hiszen a mintanagyság még akkor is milliós nagyságrendű lehet egy-egy hónapra, ha Luhn megfigyelésére támaszkodva további elemeket zárunk ki a vizsgálatból. A manuális beavatkozásra mutatkozó igényt tovább csökkentendő kísérleteket folytattam arra, hogy miként lehetne egy úgynevezett „készségszótárat" a feldolgozási folyamatba kapcsolni, illetve annak vizsgálatára, hogy milyen eredmények érhetőek el ezzel az új rendszerkomponenssel.

6.1.2 Kompetenciaelemek beazonosítása szótár alapján

Egy másik, automatizáltabb megoldás lehet az álláshirdetésekből kinyert n-gramok átfuttatása valamilyen szemantikusan felépített tudásstruktúrán, amilyen például az ESCO vagy a STUDIO ontológia. Az ESCO ontológia 0-ás verziójának „SimpleConcept" osztálya például megközelítőleg 10500 olyan elemet tartalmaz[26], ami kompetenciaként értelmezhető, míg az 1-es verzió „*Skill*" osztálya hozzávetőlegesen 8 ezer egyedi kompetenciából[27,28] áll, és ezen elemek alternatív megnevezéseiből további 51 ezer[29] tartozik hozzá. A dolgozat véglegesítésének idején az ontológia 1.0.8-as verziója 13485 egyedi készség- és kompetenciaelemet tartalmazott[30]. A STUDIO ontológia jelenlegi verziója több mint 2500 tudáselemet tartalmaz (Vas, 2016). Ezeknek az ontológiáknak a tartalma jelen megközelítés alapján felfogható egy egyszerű szótárként.



17. ábra: A kompetenciaelemek szótár alapú feltárása

[26] 2017 március 14-i adat
[27] „*preferredLabel*" attribútum alapján
[28] 2019 június 19-i adat
[29] „*alternativeLabel*" attribútum alapján
[30] 2020 október 22-i adat

A kompetenciák „szótár alapú” beazonosításának logikai áttekintését adja a 17. ábra. Ebben az esetben a bemeneti dokumentumokból kibányászott szavakból minden lehetséges módon – de a szavak sorrendjének megtartásával – szóenneseket képzünk. Amennyiben egy így adódott kifejezés megtalálható valamely említett ontológia „szótárában” úgy a szóban forgó kifejezés elfogadható valós, információt hordozó kompetenciaként. Tehát teljes egyezés esetén elfogadhatjuk, hogy beazonosítottunk egy, a probléma kontextusában érvényes kompetenciát. Részleges egyezés esetén, a tartalmazás irányának függvényében feltehető, hogy vagy a kifejezés specifikálja az adott kompetencia-szótár egy elemét, vagy fordítva[31]. Mindkét esetben meggondolandó, hogy nem áll-e fent ilyenkor valamilyen hierarchikus reláció a két objektum között, mely lehetőség vizsgálata szintén célja lehet a későbbi kutatásnak.

A „szótár”, amelyben keresünk, egy külső, cserélhető modulként kell, hogy kapcsolódjon az implementálandó alkalmazáshoz, így több különböző forrás használatával is lehet tesztelni a modellt. Ilyen források lehetnek például a már említett STUDIO, vagy az ESCO ontológia, illetve egyéb külső rendszerek adatai. Ezeken a forrásokon kívül, a 2.4 alfejezetben tárgyalt kapcsolódó kutatások esetében, hasonló célra több kutató használta még a Wikipedia *API*-t, az O*NET adatait, de súlyozásra például a Google Search *API*-t is.

6.1.2.1 A kompetenciaszótár előállítása

Jelen dolgozatban elsősorban a munkaerőpiac ICT szegmensére fókuszáltam, így az ontológiákból képzett kompetenciaszótárat ennek megfelelően leszűrtem. Az ESCO esetében az “információs és kommunikációs technológiák” megnevezésű[32], *Concept* osztályba tartozó, magas szintű készségből, illetve az ugyanilyen nevű[33], *Taxonomy* osztályba tartozó készségcsoportból kiindulva, a specifikumok felé mutató kapcsolatokon keresztül gyűjtöttem le a releváns elemeket. Így 323 darab egyedi kompetenciaelemet[34], és azok alternatív címkéinek felhasználásával összesen 1511 készségként elfogadott kifejezést[35] szűrtem le az ESCO ontológiából. A Studio

[31]Természetesen az adott kifejezés, illetve a nem egyező rész irreleváns is lehet, amely például arra utalhat, hogy az n-gramok hossza rosszul lett megválasztva.

[32]https://ec.europa.eu/esco/api/resource/concept?uri=http://data.europa.eu/esco/skill/aeecc330-0be9-419f-bddb-5218de926004

[33] https://ec.europa.eu/esco/api/resource/taxonomy?uri=http://data.europa.eu/esco/concept-scheme/skill-ict-groups

[34] https://github.com/gneusch/phd_results/blob/main/skill_dict/ictSkills_filtered.jl

[35] https://github.com/gneusch/phd_results/blob/main/skill_dict/esco_ict_labels_piped.csv

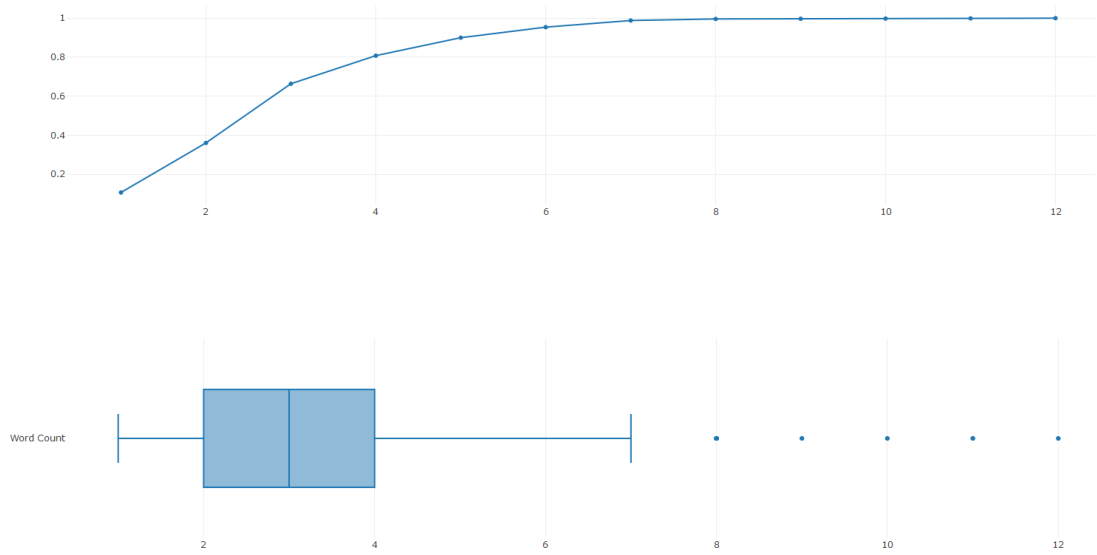
ontológiában további 474 releváns elemet[36] azonosítottam be hasonló módon, a témába vágó részontológiákból kiindulva. Pár kifejezést manuálisan hozzáadva alakult ki a későbbi kísérletek során használt, összesen 2110 elemet számláló kompetenciaszótáram.

6.1.2.2 Optimális n-gram hossz megállapítása

Az álláshirdetések alapján előállított n elemű kifejezések hosszát – azaz, hogy összesen hány szóból állhatnak – legtöbbször szakértői döntés alapján határozzák meg az adott feladat függvényében. A szakirodalom alapján elmondható, hogy általában, osztályozási feladatok esetében, a 2 és 3 hosszú szószekvenciák javítják az eredményeket, míg a hosszabb kifejezések kevésbé hasznosak (Fürnkranz, 1998). A hosszú kifejezések legtöbbször a következtetések levonásához nélkülözhetetlen adatok hiánya miatt nem modellezhetők megfelelően (Chong *et al.*, 2013). Jelen esetben, mivel a kompetenciák beazonosításához szótár alapú megközelítést is használni kívánok, a vizsgált szószekvenciák maximális hosszát a szótárban található kifejezések hossza alapján határoztam meg.

A 2110 elemű, szűrt kompetenciaszótárban található kifejezések hosszának gyakorisági statisztikái láthatók a 2. melléklet 13. táblázatában.

A kifejezések hosszának átlaga 3.3, a szórás 1.6, míg a módusz és a medián 3.



18. ábra: Kompetenciaszótár elemhossz gyakoriságok vizuális ábrázolása

[36] https://github.com/gneusch/phd_results/blob/main/skill_dict/filtered_studio_en_ict_labels.txt

A 18. ábra felső részében látható nyereségdiagram[37] alapján megállapítható, hogy a 3-nál több szóból álló n-gramok generálása egyre csökkenő nyereséggel jár (mivel a szótári elemek körülbelül 70%-a 3 vagy kevesebb szóból áll), viszont a költségeket számítási kapacitás, idő stb. formájában jelentősen növeli. A 18. ábra alsó felében látható dobozábra alapján csak a 7 fölötti szószámmal bíró kifejezések számítanak kiugró értéknek (*outlier*), de a kompetenciaszótár elemeinek 90%-a 5 vagy kevesebb szóból áll. Ha a szavak számának átlagához hozzáadom a szórást, szintén ~5 adódik. Amennyiben a stopszavak a kompetenciaszótár elemeiből eltávolításra kerülnek, ez az érték megközelítőleg 4-re csökken (14. táblázat és 33. ábra). Az előzőek alapján a feldolgozás során maximum 5 elemű n-gramokat fogok generálni ($N \in \{1,..5\}$) az álláshirdetések szövegéből, illetve azon esetekben, amikor a stopszavakat eltávolítom, az N értékét 4-ben maximalizálom. Így némi információvesztéssel kell ugyan számolnom, de ez a költségoldalon várhatóan megtérül.

6.1.2.3 A kizárólag szótáron alapuló kompetenciabeazonosítás lehetséges problémái

A szótár alapú megközelítés előnye, hogy teljes egyezés esetén, a beazonosított kompetenciaelemek között nem lesz zaj, azaz vélhetően nem követünk el másodfajú hibát, hogy egy adott elemet kompetenciaként fogadunk el, és nem az, amennyiben a hipotézisünk az, hogy az adott kifejezés kompetencia. Felmerülhet ugyanakkor az azonosalakúság problémája, hogy kontextus függvényében ugyanaz a kifejezés más-más entitást azonosít. Erre a problémára (*word sense disambiguation*) próbált megoldást találni például idézett cikkében Hoang szerzőtársaival a Google API rangsorainak segítségével (2018).

Ezzel ellentétben az elsőfajú hiba elkövetésének valószínűsége fennáll, hiszen nem feltételezhetjük, hogy a rendelkezésünkre álló kompetenciaszótár teljes, így biztos lesznek az álláshirdetésekben olyan elemek, melyek bár elfogadhatók lennének kompetenciaként, mégsem ismerjük fel őket. Ilyen jellegű hiba adódhat például amiatt, ha egy adott kompetenciaelem nem egy, a szótárban hozzá társított kifejezéssel leírva szerepel a hirdetésekben. Előfordulhatnak például elírások, eltérő igeidők, illetve a használt szavak más alakja, de akár alternatív megnevezések, szinonimák is. Azt feltételezem, hogy ezeknek a hibáknak egy részét ki lehet küszöbölni a szöveg

[37] Az ábra X tengelyén a szavak száma, míg az Y tengelyen a kumulatív gyakoriság szerepel. A diszkrét értékek azért lettek összekötve egy folytonos görbével, hogy vizuálisan is jól lehessen látni, hogy hol kezd el a meredekség csökkenni, ugyanis az ennél az elemszámnál több szót tartalmazó kifejezések bevonása a feldolgozásba egyre kisebb nyereséggel jár.

megfelelő előkészítésével, például a stopszavak vagy éppen a kontrol karakterek eltávolításával, illetve a szavak szótári alakra hozásával. Így a modell szenzitivitása növelhető, azonban ez esetben természetesen mivel információt veszítünk, nem feltételezhetjük tovább az eredmény zajmentességét, hiszen például egymás mellé kerülhetnek olyan szavak, melyeket addig írásjel választott el stb., azaz a modell precizitásának csökkenése várható.

Feltételezhető, hogy vannak továbbá olyan kompetenciakifejezések, amely nem az adatok valamilyen hibája miatt nem azonosíthatóak be, hanem egyszerűen azért, mert hiányoznak a kompetenciaszótárból. Ilyenek lehetnek például a nem általánosan, illetve csak periodikusan keresett, vagy időben nem releváns készségekre mutató hivatkozások (lásd például az utóbbi időben a Covid-19 vírus kapcsán a Cobol nyelvre megnövekedett keresletet (King, 2020)), vagy a legújabb technológiák ismeretére utaló kifejezések. Azaz összességében elmondható, hogy az eredményeket nagymértékben meghatározza a kiinduló szótár minősége és teljessége.

6.1.2.4 A tisztán szótárra épülő megközelítés eredményei

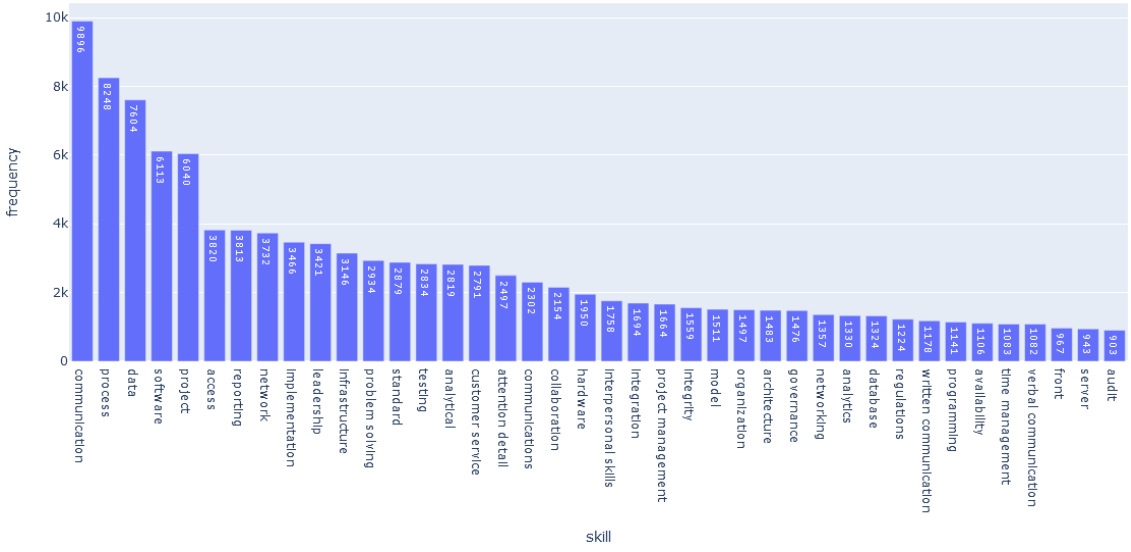
A jelen fejezetben bemutatott kísérletek során az előzőekben is használt 2019 októberi adathalmaz 22213 hirdetéséből 6.1.2.2 rész szerint képzett *n* hosszú kifejezéseket feleltettem meg a 6.1.2.1 alfejezetben leírt kompetenciaszótárnak. A kísérletet végrehajtottam előfeldolgozás nélkül, a stopszavak eltávolításával, illetve úgy is, hogy a két kifejezéshalmaz elemeit előzetesen lemmatizáltam. Az előfeldolgozás során, a stopszavak eltávolításához, illetve tokenizálásra az *NLTK* (Loper és Bird, 2002), míg a lemmatizáláshoz a *Stanza* (Qi *et al.*, 2020) Python könyvtárakat használtam. A kísérletek alapvető statisztikáit tartalmazza a 4. táblázat.

	Előfeldolgozás nélkül	**Stopszavak eltávolításával**	**Szótári alakra hozással**
Beazonosított szótári kifejezések száma	459	488	613
Találatok száma az összes dokumentumban	129307	137220	200945
Dokumentumfrekvencia (*df*) átlaga	281.7	281.1	327.8
Dokumentumfrekvencia maximuma	9896	9896	11859
Dokumentumfrekvencia 90. percentilise	522.5	514.5	618

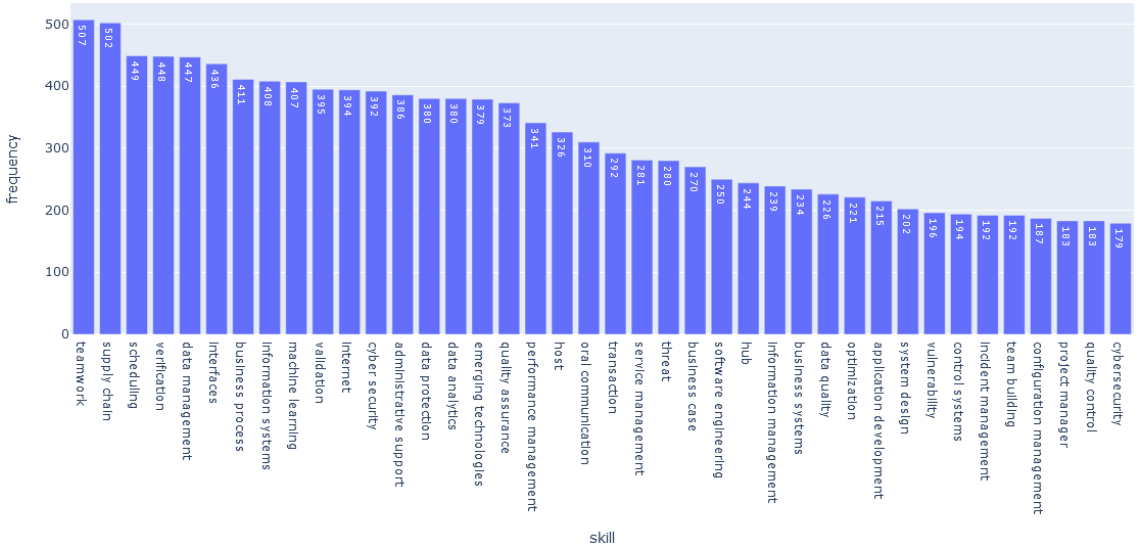
4. táblázat: a beazonosított kompetenciakifejezésekre vonatkozó alapstatisztikák

A szótárral végzett kísérletek statisztikáinak bemutatásakor azért tartottam fontosnak megemlíteni a feltárt kifejezések dokumentumfrekvenciájának maximumát, a 90. percentilissel egyetemben, mert látható, hogy a leggyakoribb kifejezés az összes vizsgált hirdetés 37,7%-ában megjelent, és feltételezhetjük, hogy az ilyen nagy számban szerepeltetett kompetenciák felfoghatók egyfajta korpuszspecifikus stopszóként is. Ez azt jelenti, hogy valószínűsíthető, hogy ezeket a kifejezéseket az állásajánlatokban mintegy rutinból szerepeltetik, így azok jelentős többletinformációt, illetve újdonságot nem hordoznak, így céljaink szempontjából kisebb jelentőséggel bírnak.

A fentiek alapján beazonosított kifejezések tüzetesebb vizsgálata során valóban bebizonyosodott, hogy a leggyakoribb felismert kompetenciák eléggé általánosak, illetve van köztük olyan is, ami puha készséget reprezentál és az ESCO ontológia automatizált leszűrése során „maradt” a szótárban (19. ábra). Az előzőekkel ellentétben a 90. percentilis körül található kifejezések céljaink szempontjából valóban relevánsabbnak tűnnek, bár továbbra is több közülük általánosabb jellegű kompetenciára mutat (20. ábra).



19. ábra: A leggyakoribb kifejezések



20. ábra: A 90%-os percentilis környéki kifejezések

Az statisztikák alapján jól látszik, ahogy az előzetesen is várható volt, hogy minden adattisztítási lépés folyamatba iktatásával növekszik a beazonosított kompetenciakifejezések száma. A stopszavak eltávolítása után például olyan releváns kifejezések kerültek beazonosításra, mint például „*service portfolio management"* vagy éppen „*principles data protection*"[38]. Drasztikus növekményt eredményezett a beazonosított kompetenciák számában a szavak lemmatizálása is. Ez utóbbi olyan addicionális találatokat eredményezett, mint például „*customer relationship management solution*", „*use query language*", „*business impact analysis*", illetve számos utalás az SQL Server különböző verzióira stb[38].

Nem szabad azonban elfelejteni, hogy az így azonosított szókapcsolatokat az információvesztésből adódóan lehetséges, hogy tévesen jelöljük kompetenciának (hamis pozitív találat). Ez adódhat például abból, hogy az írásjelek és a stopszavak, azaz a kontextus egy részének eltávolítása következtében egyes szavak egymás mellé kerülnek stb. Példának lehet hozni továbbá, a stopszavak eltávolítása után, az eredményekben nagy súllyal megjelenő „*audit*", „*regulations*", „*governance*" és „*standard*" unigramokat, melyeket az algoritmus az „*IT*" rövidítés eltávolítása miatt azonosított be, ugyanakkor a kontextus ismerete nélkül közel sem biztos, hogy kompetenciaigényt jelölnek. Általánosítva ezt a megfigyelést elmondható, hogy az előfeldolgozási lépések, illetve a későbbi kísérletek során bármilyen valószínűségi

[38] A teljes lista megtekinthető a 3. mellékletben.

modell alkalmazásával a hamis pozitív találatok száma nő, így az elérhető precizitás csökkenhet, amennyiben a helyes pozitív találatok aránya kisebb ütemben emelkedik.

Összességében a szótárral támogatott kompetenciakifejezések beazonosításának kísérletéről elmondható, hogy sokkal nagyobb információtartalmú eredményeket szolgáltatott azáltal, hogy jelentősen lecsökkentette a zaj mértékét, azaz az irreleváns szókapcsolatok számát. A modell egyik problémája a kontextus hiánya miatt nem feltétlenül egyértelmű jelentéstartalommal bíró kifejezések megjelenése az eredmények között, melyek esetében van némi bizonytalanság abban, hogy valós pozitív találatot reprezentálnak-e. A hamis pozitív találatok kiszűrésére különböző – például *WSD* (*word sense disambiguation*) és *NEN* – módszerek használhatók. Ezen megoldások jelen problémára alkalmazása túlmutat a dolgozat keretein, de a kutatás későbbi fázisainak mindenképpen részét kell képeznie. A másik felmerülő probléma az esetleges hamis negatív elemek lehetősége, vagyis az, hogy az alkalmazott szótár, annak teljessége és pontossága nagyban meghatározza a beazonosítható kompetenciák körét. A következőkben ez utóbbi problémára próbálok megoldásokat találni, és megvizsgálni, hogyan lehetne minél több olyan kompetenciaelemet is feltárni, ami nem mutat teljes egyezést a szótár egyik elemével sem.

6.1.3 Hasonlóság és távolság alapú modellek

Az érvényes kompetenciákat reprezentáló, a szótárban explicit nem szereplő kifejezések beazonosítása történhet a szóennesek közötti hasonlóság (s), távolság ($d = 1 - s$), illetve azok együttes előfordulási gyakoriságainak felhasználásával alkotott valószínűségi modellek alapján. A következőkben vázolt módszerek, a szótár alapú módszer használatával ötvözve, például a részlegesen egyező találatok listájának további, automatikus szűkítésére is lehetőséget adhatnak.

A karakterláncok (*string*) hasonlóságának megállapítására alkotott metrikákat több különböző csoportba sorolhatjuk. Gomaa és Fahmy (2013) például karakter- (*character*) és kifejezés- (*term, token*) alapú kategóriákat különböztet meg. Illetve külön csoportba szokták sorolni a Cilibrasi és Vitányi által kidolgozott normalizált tömörítési távolságot (2005) ami egy elméleti konstrukció, de tényleges tömörítési eljárásokat behelyettesítve szintén gyakran használják szövegek hasonlóságának számításához. A következőkben röviden bemutatom ezeket a csoportokat, illetve

azokat a leggyakoribb, legszélesebb körben használt konkrét metrikákat, melyeket egy valószínűségen alapuló, logisztikus regressziós modell építése során felhasználtam.

6.1.3.1 Kifejezéstávolság mérése tokenek alapján

A különböző hasonlóság vagy távolság alapú modellek alkalmazásának jelen problémára számos előnye lehet, akár ötvözve a szótár alapú megközelítés használatával. Egyik ilyen előny, amennyiben a kompetenciákat jelölő kifejezéseket egy vektortérben numerikusan ábrázoljuk, hogy két kifejezés hasonlósága 1 lesz, ha ugyanazokat a szavakat tartalmazzák, akkor is, ha a szavak sorrendje a két kifejezésben különböző, míg egyszerű szövegegyezés alapon ezt a kapcsolatot nem tártuk volna fel.

A kompetenciaszótárt és a hasonlóság alapú megközelítést ötvöző modell felépítésének első lépése a rendelkezésre álló szemantikus forrásokban található kompetenciákat reprezentáló kifejezések numerikus ábrázolása. Mivel ez esetben a dokumentumaink egy-egy rövidebb kifejezésből állnak, ezért gyakoriságok tárolásának nincs értelme, hiszen jellemzően csak azt tudjuk elmondani, hogy egy adott szó megtalálható-e a dokumentumban – azaz nem valószínű, hogy egy szó megismétlődik egy kompetenciát jelölő kifejezésben. A kompetenciák ábrázolására tehát bináris vektorokat alkalmazhatunk, melyek hasonlóságát praktikusan páronként tudjuk vizsgálni.

Vegyük az alábbi egyszerűsített példát, melyben a kompetencia-szótárunk az alábbi – az ESCO ontológia *Skill* osztályából választott – két elemből áll:

- ict network security risks
- information security strategy

Ebben az esetben a teljes szótárat egy 6 elemű vektorral lehet ábrázolni: [*ict, information, network, risks, security, strategy*]

Az eredeti kompetenciákat pedig a következő két bináris vektorral:

- [1,0,1,1,1,0]
- [0,1,0,0,1,1]

Az állásajánlatokban található kompetencia-elemek beazonosítása ezek után úgy történhet, hogy egy adott hirdetés feldolgozása során előállt *m* darab szóennes, és kompetenciaszótár elemeinek minden lehetséges párosítására megvizsgáljuk azok távolságát, az előzőekben bemutatott ábrázolás alapján. A két kifejezéslista elemeinek

ilyen, *token* alapú összehasonlítása történhet nem csak a kifejezéseket alkotó szavak, hanem a karakterek távolságának számításával is.

Choi és munkatársai (2010) munkájukban 76 olyan távolság- és hasonlóság-mérőszámot gyűjtöttek össze, melyek alkalmasak bináris tulajdonságvektorok összehasonlítására. Ahogy cikkükben kifejtik, a „bináris hasonlóság, illetve távolságmérés sok mintaelemzési probléma esetében kritikus fontosságú", de az elérhető eredmények minősége és az elemzés hatékonysága nagyban függ a megfelelő távolságmérték kiválasztásától (Choi *et al.*, 2010, p. 1).

Az egyik legszélesebb körben használt távolságmetrika a **Jaccard együttható**, vagy Jaccard index, ami az egyező tulajdonságok és az összes tulajdonság arányában határozza meg két megfigyelt objektum hasonlóságát. Ez a módszer mind binárisan kódolt adatokon, mind karakterláncokon használható. Általánosan az együttható értékét két halmaz esetében azok metszetének és uniójának hányadosa adja. Illetve bináris vektorok esetében amennyiben:

- $a = i \wedge j$,
- $b = \neg i \wedge j$,
- $c = i \wedge \neg j$,
- $d = \neg i \wedge \neg j$

A Jaccard együttható számítása a következő képlet alapján történik (Choi *et al.*, 2010, p. 2).

$$S_{JACCARD} = \frac{a}{a+b+c}$$

Ez alapján az előző példában szereplő két kifejezés Jaccard hasonlósága 1/6 ≈ 0.17. Az eredmény a két karakterláncban megegyező tokenek arányát mutatja meg. A **Sørensen-Dice** együttható annyiban különbözik a Jaccard indextől, hogy a vizsgált halmazok metszetének kétszeresével kalkulál (Verma és Aggarwal, 2020).

$$S_{Sørensen-Dice} = \frac{2a}{2a+b+c}$$

Szintén széles körben használják az információkinyerés és a természetesnyelv-feldolgozás területén a **koszinusz hasonlóság**ot, amit két *N* dimenziós vektor *x* és y esetén, Li és Han (2013, p. 1) alapján a következőképpen kalkulálhatunk.

$$S(x,y) = \frac{\sum_{i=1}^{N} x_i \times y_i}{\sqrt{\sum_{i=1}^{N} x_i^2}\sqrt{\sum_{i=1}^{N} y_i^2}}$$

Melyet bináris vektorok esetén a következőképpen is felírhatunk Choi *et al.* (2010)[39] alapján.

$$S_{koszinusz} = \frac{a}{\sqrt{(a+b)(a+c)}}$$

Ezek alapján a két kompetencia koszinusz hasonlósága a fenti példából $\frac{1}{\sqrt{12}} \approx 0.29$. Gugnani és Misra (2020, p. 5) a koszinusz távolságot $(1 - S_{KOSZINUSZ})$ használja az álláshirdetésekből képzett kifejezések kompetenciaszótár elemeitől vett távolságának kiszámítására.

Látható, hogy a koszinusz hasonlóságuk alapján a két kifejezés közelebb van egymáshoz, mint a Jaccard együttható alapján, mivel az első esetében az eltérések kisebb súllyal számítanak, mint utóbbinál. Ha karakter és nem szó alapon számoljuk ezt a két mutatót, akkor a két kifejezés hasonlósága, a mutatók bemutatásának sorrendjében 0,571 és 0,728.

6.1.3.2 Hozzávetőleges karakterlánc illesztés

A teljes szótár bináris leképezése helyett távolság-mérőszámok számolhatók sztringeken (például szavak vagy karakterek) is. A számítástudományban azokat a módszereket és algoritmusokat, amelyek karakterfüzérek hibát megengedő megfeleltetésével foglalkoznak, hozzávetőleges karakterlánc-illesztésnek, vagy közelítésnek, angolul *approximate string matching* (*ASM*) nevezik. Formálisabban fogalmazva; a hozzávetőleges karakterlánc-illesztés során szövegek olyan pozícióit keressük, amelyek maximum k hibával megfelelnek egy adott mintának (Navarro, 2001, p. 6). Ebből a megközelítésből két karakterlánc közötti távolság $d(x,y)$ azon műveletek sorának legkisebb költsége, amelyek x-et y-á alakítják át (Navarro, 2001, p. 7; Ukkonen, 1985, p. 1). Ebből a távolságfüggvény a következőképpen adódik:

$d: S \times S \rightarrow \mathbb{R}$ ahol S egy véges ABC

Egy karakterláncpárt x és y képez a valós számok halmazára (Navarro, 2001, p. 7). Az ily módon adott távolságfüggvény, Hall és Dowling (1980, p. 387) alapján a következő tulajdonságokkal bír.

[39] Az idézett cikkben a képlet hibásan szerepel, így ez az idézet nem teljesen szöveghű.

- $d(x,y) \geq 0$
- $d(x,y) = 0 \Leftrightarrow x = y$
- $d(x,y) = d(y,x)$
- $d(x,y) + d(y,z) \geq d(x,z)$

Azok az algoritmusok például, melyek szerkesztési távolságot (*edit distance*) számítanak, a beillesztés, törlés, helyettesítés és (egyes esetekben) a felcserélés (transzpozíció) műveleteit használják fel annak kiszámítására, hogy x karakterlánc milyen költséggel (például összes szükséges művelet száma) alakítható át y-ná. Szerkesztési távolság számolható egyes sztringeken karakter, de akár szó alapon is.

Az egyes konkrét szerkesztési távolságmetrikák eltérnek egymástól abban, hogy mely műveleteket engedélyeznek, továbbá abban is eltérhetnek, hogy az egyes műveletekhez milyen költséget rendelnek. Azokat a metrikákat, amelyek megkülönböztetik az egyes műveleteket és/vagy az egyes karaktereket, és azokhoz más-más költséget rendelnek, *általános*, míg azokat az algoritmusokat, melyek mindent 1 költséggel számolnak, *egyszerű* távolságmetrikának nevezi a szakirodalom (Navarro, 2001).

Az előzőek alapján például a jeltovábbítás területén is használt **Hamming távolság** azt a minimális helyettesítésszámot jelöli, amellyel x karakterszekvenciát y sztringgé lehet átalakítani. A **Levenshtein távolság** a felhasználható műveletek közé felveszi a beillesztést és a törlést is. A **Damerau-Levenshtein távolság** továbbmegy, és engedélyezi a szomszédos karakterek felcserélésének lehetőségét is. Ezek a távolságmetrikák megengedik, hogy mind x, mind y karakterláncokra alkalmazhassuk az egyes műveleteket (Bard, 2007; Damerau, 1964; Navarro, 2001).

Két karakterlánc **Jaro hasonlóságát** a két kifejezésben megegyező karakterek száma, és azok egymástól vett távolsága határozza meg (Gomaa és Fahmy, 2013). Két azonos karakter egyezőségét nem csak akkor fogadja el a metrika, ha ugyanabban a pozícióban vannak az egyes sztringekben, hanem akkor is, ha távolságuk maximum

$$w = \left\lfloor \frac{\max(|x|,|y|)}{2} \right\rfloor - 1.$$

Az így azonosnak elfogadott karaktereket m-el, az összes megegyezőnek elfogadott karakterpár halmazában, a nem megegyező pozíciók számának felét (transzpozíciók)

pedig t-vel jelölve, a két kifejezés Jaro hasonlósága a következőképpen alakul (Dreßler és Ngonga Ngomo, 2017):

$$S_{JARO} = \begin{cases} \frac{1}{3}\left(\frac{m}{|x|} + \frac{m}{|y|} + \frac{m-t}{m}\right) & : m > 0 \\ 0 & : minden\ más\ esetben \end{cases}$$

Az előzőek Winkler által javasolt kiegészítését, ami azt feltételezi, hogy a karakterláncok elején található egyező karakterek a legfontosabbak hasonlóság szempontjából, **Jaro-Winkler hasonlóságnak** nevezi a szakirodalom. Ebben az esetben, amennyiben a vizsgálat tárgyát képező karakterláncok Jaro hasonlósága nagyobb egy adott b_t küszöbértéknél, az eredeti értékhez hozzáadódik egy 1-nél kisebb szám, melyet a maximálisan figyelembe vett egyező előtaghossz, egy skálaérték és a Jaro *távolság* szorzatából képzünk (Keil, 2019).

6.1.3.3 Szekvencián alapuló algoritmusok

Szintén az egyező karakterek számosságát vizsgálja a **Ratcliff-Obershelp hasonlósági** algoritmus, ami az egyező karakterek számának kétszeresét elosztja a két karakterlánc hosszával. Az algoritmus egyező karakterek számának megállapításához először megkeresi az összehasonlítandó két szöveg leghosszabb megegyező részsztringjét (*LCS*, *longest common substring*), majd annak két oldalán is rekurzív módon megkeresi az egyező részsztringeket (Ratcliff és Metzener, 1988).

6.1.3.4 Normalizált tömörítési távolság

Egy véges bináris sorozat $x \in \{0,1\}^*$ Kolmogorov bonyolultsága, $K(x)$ az a legrövidebb bináris program, amivel az univerzális Turing gép x-et kiszámítja (Bennett *et al.*, 1998). Hasonlóképpen a feltételes Kolmogorov komplexitás, $K(x|y)$ „annak a legrövidebb y bináris szónak a hossza, ami inputtal az univerzális Turing gép az x szót kinyomtatja” (Bártfai, 2010, p. 100). Mivel a Kolmogorov bonyolultság felfogható úgy, mint az a minimális mennyiségű információ, ami x vagy y előállításához szükséges, ezért x és y (univerzális) *információs távolsága* (*information distance*) felfogható úgy, mint az a legrövidebb p program, ami x-et előállítja y-ból vagy fordítva (Bennett *et al.*, 1998, p. 5).

$$E(x,y) = |p| = max\{K(y|x), K(x|y)\}$$

Ez a távolság azonban abszolút, míg praktikus alkalmazásokban, például az adatbányászatban, egyes objektumok távolságának meghatározásához jobban megfelel egy relatív mutató. Ez a relatív metrika a *normalizált információs távolság*

(*normalized information distance, NID*), mely értékeit a [0,1] intervallumon veszi fel, és melyet Vitányi et al. (2009, p. 47) alapján a következőképpen írhatunk fel:

$$e(x,y) = \frac{max\{K(y|x), K(x|y)\}}{max\{K(y), K(x)\}}$$

Azonban a *normalizált információs távolság* képletében K függvény nem kiszámítható, ezért a gyakorlatban *normalizált tömörítési távolságot* (*normalized compression distance*, *NCD*) alkalmaznak, ahol is valós tömörítési algoritmusokat (Z) használnak, mint például a gzip, bzip2 stb. (Cilibrasi és Vitanyi, 2005; Vitányi *et al.*, 2009).

$$e_Z(x,y) = \frac{Z(xy) - min\{Z(x), Z(y)\}}{max\{Z(x), Z(y)\}}$$

A *normalizált tömörítési távolság* kalkulálásához használható például az entrópia kódolás, ahol a valószínűségek tulajdonképpen az egyes tokenek relatív gyakoriságai.

$$Z = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

6.1.3.5 Értékelés

A példa kifejezések (*ict network security risks*, *information security strategy*) között számított hasonlósági és távolságértékeket tartalmazza a 5. táblázat[40].

Metrika	Távolság	Normált távolság	Hasonlóság
Jaccard			0,167
Sørensen-Dice			0.286
Koszinusz			0,288
Hamming	18	0,62	
Levenshtein[41]	16	0,552	
Jaro		-	0,66
Jaro-Winkler		-	0,695
Ratcliff-Obershelp			0,545
$NCD_{Entrópia}$			0,94

5. táblázat: példa kifejezések hasonlóság- és távolságértékei

Az információkinyerés témakörébe tartozó feladatok esetén a hasonlósági mérőszámok vektortérmodellben súlyként való szerepeltetése ellen gyakori érv az, hogy nem veszik figyelembe a kifejezések gyakoriságát, így nehezen lehet megítélni az egyes feltárt elemek fontosságát egy adott dokumentum esetében. Ebből kifolyólag a fent tárgyalt mérőszámokból származó információkat nem önmagukban fogom értékelni, hanem a hirdetésekből generált szóenneseknek egy jól meghatározott

[40] A számítások a TextDistance nevű Python könyvtár felhasználásával készültek (Voronov, 2020).
[41] A két kifejezés Damerau-Levenshtein távolsága megegyezik a Levenshtein távolságukkal.

kifejezéshalmaztól (kompetenciaszótár) való távolságát fogom vizsgálni egy valószínűségalapú modellben, és arra keresem a választ, hogy egy adott kifejezés milyen valószínűséggel fogadható el kompetenciaként. Úgy gondolom, hogy a fent bemutatott metrikák csak egy komplexebb modell kontextusában, mint magyarázó változók lehetnek relevánsak és jól alkalmazhatók a problémára.

6.1.4 Kompetenciakifejezések valószínűségalapú beazonosítása

Egy valószínűségalapú modell építésével célom annak vizsgálata volt, hogy egy előzetesen, manuálisan felcímkézett tanítóhalmaz segítségével felírható-e egy, az előzőekben részletezett metrikákon alapuló egyenlet, amelynek segítségével beazonosíthatóak olyan kifejezések, melyek úgy mutatnak rá egyértelműen a kompetenciaszótár valamely elemére, hogy az adott kapcsolatot a teljes egyezés vizsgálata során nem tudtam volna feltárni. További célom volt, hogy megvizsgáljam, hogy a modell alkalmas-e kapcsolódó kifejezések beazonosítására is, melyek segítségével a kiindulási szótár, illetve a forrásul szolgáló ontológiák bővíthetőek. Bizonyos mértékben tehát ez a feladat átfed a látens kompetenciák feltárására irányuló erőfeszítéseimmel.

Az egyik módszer, ami egy kétértékű függő változó becslésére alkalmazható, a bináris logisztikus regresszió vagy logit modell. Tulajdonképpen egy osztályozó eljárásról van szó, mely jelen céljaimnak tökéletesen megfelel, hiszen azt „akkor használjuk, ha előre definiált, egymást kölcsönösen kizáró csoportok egyikébe soroljuk be a megfigyeléseket a magyarázó változókból nyert információ alapján” (Kovács, 2014).

6.1.4.1 A modell tanítása

A kísérlet előkészítése során véletlenszerűen kiválasztott álláshirdetésekből – a mondathatárokat figyelembe véve – a stopszavak eltávolítása után[42] négyhosszú szósekvenciákat képeztem (6.1.2.2 alfejezet), és azoknak a kompetenciaszótár összes eleméhez vett hasonlóságát az összes tárgyalt metrikával megállapítottam. Tekintve, hogy az így kapott eredménymátrix sorainak száma megegyezik a szótár és az álláshirdetésekből alkotott szóennesek szorzatával, ami még relatíve kis számú hirdetés esetében is százezres nagyságrend – mely adathalmazt manuálisan kívántam felcímkézni – ezért úgy döntöttem, hogy a token alapú Jaccard hasonlósági metrika

[42] A kísérletet végrehajtottam a stopszavak eltávolítása nélkül, és a szöveg előzetes lemmatizálása után is, de a legjobban illeszkedő modellt ebben az esetben, azaz a stopszavak eltávolításával kaptam, így itt most ezt ismertetem.

eredménye alapján meghatározok egy döntési/vágási értéket, és azokat az eseteket, ahol a kapott érték kisebb, mint ez a szám, eldobom. A szűkítést azért alkalmaztam, mert a jelen modellel nem az volt a célom, hogy a hirdetésekben megjelenő kompetenciákat maradéktalanul feltárjam, hanem az, hogy a szótári elemek segítségével minél több azokhoz hasonlót megtaláljak. Nem csak azért választottam a Jaccard indexet erre a célra, mert az az egyik legszélesebb körben alkalmazott hasonlósági metrika, hanem mert mivel két halmaz metszetének vizsgálatára épül, ezért nullától különböző értéke azt jelenti, hogy a vizsgált két kifejezés legalább egy szóban megegyezik. Token alapon számítva továbbá a Jaccard együtthatót a két halmaz elemei között, tekintve, hogy maximum négy- illetve öthosszú szóenneseket (6.1.2.2 alfejezet) képeztem a hirdetések szövegéből, a lehetséges értékkészlet nem annyira számottevő, így jobban megítélhető, hogy az eredményt hogyan befolyásolja a szavak száma az összehasonlított kifejezésekben, illetve az egyező szavak aránya. Például elmondható, hogy amennyiben két összehasonlított kifejezés közül az egyik teljes mértékben tartalmazza a másikat, úgy a Jaccard szerinti hasonlóságuk magasabb lesz, így nagyobb súlyt kaphatnak az egymást specifikáló, egy ontológia szempontjából (valószínűsíthetően) egymással hierarchikus viszonyban lévő kifejezések. Ezáltal lehetővé válhat az implicit tudás, készség és kompetenciaelemek egy bizonyos körének beazonosítása is.

Az összesen 30321 megfigyelést felcímkéztem oly módon, hogy 1 értéket kaptak azok az esetek, melyeknél a hirdetés szövegéből képzett adott szószekvenciát elfogadtam, mint valós kompetenciaelemet azonosító kifejezés. Az így kapott érték lett a logit modell függő változója. Természetesen ilyen módon az eredményeket nagyban befolyásolja a saját – egy adott részterületen nem feltétlenül szakértői – értékítéletem, és a kutatás későbbi szakaszában, megfelelő erőforrások birtokában, érdemes lenne a kísérletet szakterületi szakértők bevonásával megismételni. Továbbá említésre érdemes, hogy a kísérlet során a stopszavak eltávolítása és a hasonlósági metrikák alkalmazása miatt azt az elvet követtem, hogy a szóenneseken belül a szavak szigorú sorrendiségét nem követeltem meg egy-egy kifejezés „értelmességének” megítélésekor.

A magyarázó változók közé az egyes kifejezések hosszából képzett kategóriákat és a normalizált hasonlóságértékeket vettem fel. Megpróbálkoztam a két halmaz kifejezéseinek POS címkéiből képzett kategóriák modellben való használatával,

azonban rengeteg csoport jött létre többségében egy-két megfigyeléssel, és azok logikus összevonására nem találtam megfelelő módszert, így e változó alkalmazását elvetettem. Bár a *tf-idf* értékek felhasználásával kifejezetten jól teljesítő és jól illeszkedő modellt tudtam felállítani, végül ennek a mérőszámnak a használatát is elvetettem, mivel azt erősen befolyásolja a vizsgált álláshirdetések korpusza, ami természetesen a rendszer éles működése során folyamatosan változik, így a modellből kapott együtthatók nem lennének helyesek.

Bár a logit modell nem követeli meg a magyarázó változók szigorú függetlenségét, többek között azért, mert a kategóriaváltozók között korrelációt nem mérünk (Kovács, 2014), igyekeztem a modellből a multikollinearitást kiszűrni. Ennek megfelelően, kísérleti alapon, az átfedő tartalmú mutatók közül a rosszabbul teljesítőket kihagytam. Így eltávolítottam a Ratcliff-Obershelp és a Jaro metrikákat, a koszinusz és a Jaro-Winkler hasonlósági mérőszámokkal mutatott erős korrelációjuk miatt (4. melléklet, 15. táblázat). A független változók között így a két halmaz (szótár és a hirdetésleírásokból képzett kifejezések) szavainak száma alapján előállt kategóriák, a Jaccard, koszinusz, Levenshtein, Jaro-Winkler hasonlóság és normalizált tömörítési távolság kapott helyet a modellben.

Amennyiben azokat az eseteket fogadom el találatnak, melyekre a becsült valószínűség 40%-nál nagyobb (*cut value = 0,4*), úgy a végső modell tanítóhalmazon elért felidézési aránya 84,6%, míg a precizitás 73,7%. Ebben az esetben a 0,5-ös döntési értékhez képest a precizitás csak 2,8%-kal csökken, viszont a felidézési arány 11,6%-kal nő. Amennyiben a valós kompetenciaelemként elfogadott kifejezések mintában megfigyelt arányát (0,15) használjuk vágási értékként, úgy a modell precizitása 58,8%-ra csökken 100%-os felidézési arány mellett. A klasszifikáció eredményét, a választott 0,4-es vágási érték mellett, a 6. táblázat[43] (kereszttábla) mutatja.

[43] A kísérlet során a számítások elvégzésére az IBM SPSS Statistics programcsomag 25-ös verzióját használtam.

Classification Table[a]

Observed			Predicted hit 0	Predicted hit 1	Percentage Correct
Step 1	hit	0	1385	78	94.7
		1	40	219	84.6
	Overall Percentage				93.1

a. The cut value is .400

6. táblázat: Megfigyelések besorolása a modell becslése alapján, 0,4-es vágási érték mellett

A modell globális mutatói megfelelőek. Az *Omnibus* teszt alapján minden szokásos szignifikancia szinten elfogadhatjuk az alternatív hipotézist, azaz biztos, hogy van olyan változó a modellben melynek együtthatója szignifikáns. A pszeudo R^2 mutatók közepes determináltságot jeleznek. Cox és Snell mutatója alapján 45%-ban határozzák meg a magyarázó változók annak esélyét, hogy a kifejezés valós kompetenciaelemet azonosít, míg Nagelkerke R^2 mutatója alapján a determináltság 78,8%-os. De ezen mutatók közvetlen értelmezése félrevezető lehet, mert csak annyit „mondanak, hogy a csak konstanst tartalmazó modellhez tartozó log *likelihood* értéket hány százalékkal sikerült csökkenteni" (Kovács, 2014; Fliszár *et al.*, 2016, p. 46). Az irodalomban a megfelelőség vizsgálatára inkább a Hosmer-Lemeshow tesztet ajánlják. Ennek során a megfigyeléseket és a becsült valószínűségeket decilisekre osztjuk, és azt a hipotézist vizsgáljuk, hogy a ténylegesen bekövetkező események száma megegyezik-e az előrejelzettel az egyes decilisekre. Ezt a hipotézist jelen modellre elfogadhatjuk. A modell globális illeszkedését leíró mutatókat a 21. ábrán látható SPSS kimenet tartalmazza.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	1029.407	9	.000
	Block	1029.407	9	.000
	Model	1029.407	9	.000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	428.829[a]	.450	.788

a. Estimation terminated at iteration number 12 because parameter estimates changed by less than .001.

Hosmer and Lemeshow Test

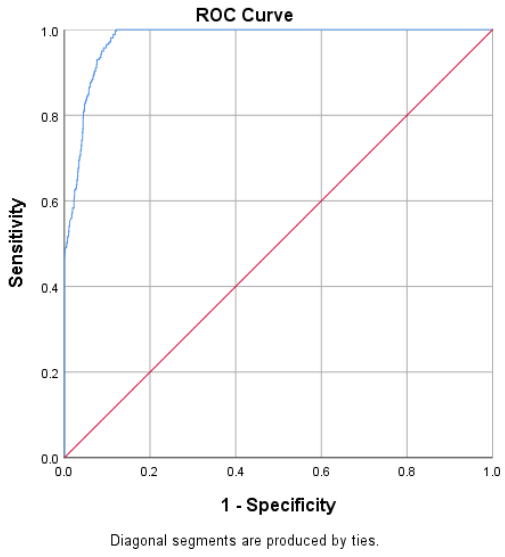
Step	Chi-square	df	Sig.
1	1.651	8	.990

Contingency Table for Hosmer and Lemeshow Test

		hit = 0		hit = 1		
		Observed	Expected	Observed	Expected	Total
Step 1	1	173	172.997	0	.003	173
	2	177	176.994	0	.006	177
	3	174	173.992	0	.008	174
	4	172	171.989	0	.011	172
	5	173	172.987	0	.013	173
	6	174	173.983	0	.017	174
	7	173	172.976	0	.024	173
	8	145	143.960	27	28.040	172
	9	84	88.744	89	84.256	173
	10	18	14.378	143	146.622	161

21. ábra: A logit modell globális illeszkedését mutató mérőszámok

A ROC görbe is a modell jó illeszkedését mutatja (22. ábra). A görbe alatti terület nagysága 0.977 és minden szokásos szignifikancia szinten különbözik a 45 fokos egyenestől (4. melléklet, 16. táblázat).



22. ábra: A ROC görbe a modell jó illeszkedését mutatja

A modellben a *Backward Wald* változószelekciós eljárással történt futtatás, illetve az átfedő tartalmú hasonlósági mutatók eltávolítása után bennmaradt magyarázó változók a szótár és a hirdetésekből képzett szóennesek szavainak számossága alapján generált kategóriák, illetve a kifejezések Jaccard, koszinusz és Jaro-Winkler hasonlósága. A Wald statisztika alapján elmondható, hogy mindegyik magyarázó változó együtthatója szignifikáns. A találat esélyére a legnagyobb pozitív hatással a két kifejezés Jaccard távolsága van, de a másik két, modellben lévő távolságmetrika együtthatója is pozitív előjelű. A vizsgált kompetenciajelölt szavainak számából képzett kategóriaváltozó esetében a referenciakategória az egyszavas kifejezések csoportja (*Nposting-Cat(1)*), amihez viszonyítva a kettő (*Nposting-Cat(2)*), illetve három és háromnál több (*Nposting-Cat(3)*) szóból álló szóennesek csoportjába tartozás mind növeli annak esélyét, hogy az adott szó kompetenciakifejezés. Ezzel ellentétben a készségszótár elemei alapján hasonlóképpen képzett változó esetében a referenciakategória a három és háromnál több szóból álló szókapcsolatok csoportja (*Nskill-Cat(3)*), melyhez viszonyítva a másik két kategóriába tartozás mind csökkenti az esélyt (4. melléklet 17. táblázat).

6.1.4.2 A modell tesztelése

A tesztadatok egyes megfigyeléseire – az előzőekben részletezett előkészítési lépések elvégzése után – számolt Jaccard, koszinusz és Jaro-Winkler hasonlósági metrikák, illetve a szószám alapján képzett kategóriák és a modell által számolt együtthatók segítségével kiszámoltam, hogy a hirdetésekben lévő szóennesek mekkora valószínűséggel reprezentálnak kompetenciának elfogadható kifejezést. A Python nyelvű implementáció kódját az 4. mellékletben a 34. ábra tartalmazza. A modell teljesítményének visszamérése céljából a manuálisan meghatározott (*manual_label*) és a kalkulált eredményeket (*hit*) kereszttáblával hasonlítottam össze.

hit * manual_label Crosstabulation

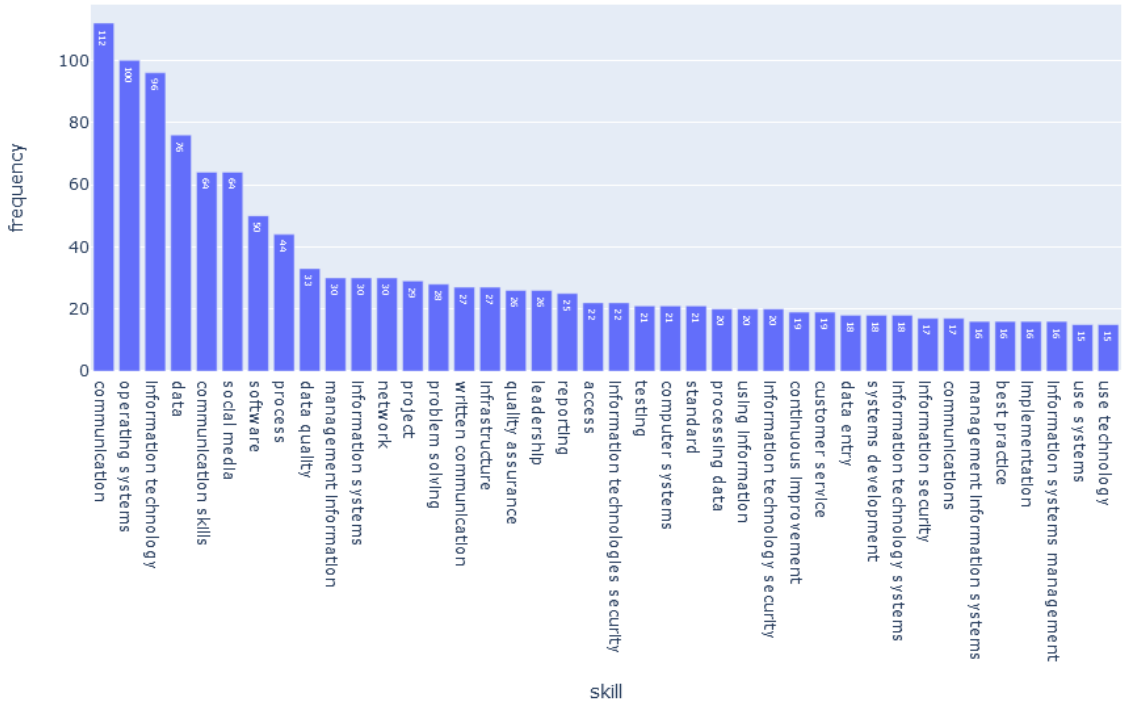
			manual_label		
			0	1	Total
hit	0	Count	23713	549	24262
		% within hit	97.7%	2.3%	100.0%
		% within manual_label	95.1%	15.0%	84.8%
	1	Count	1219	3116	4335
		% within hit	28.1%	71.9%	100.0%
		% within manual_label	4.9%	85.0%	15.2%
Total		Count	24932	3665	28597
		% within hit	87.2%	12.8%	100.0%
		% within manual_label	100.0%	100.0%	100.0%

7. táblázat: A tesztadatok automatikus- (hit), és manuális (manual_label) besorolásainak kereszttáblás összehasonlítása

Ahogy az a 7. táblázatból látható, a modell tesztadatokon mért felidézési aránya 85%, míg a precizitás 71,9%. A két változó függetlensége a Pearson khi-négyzet teszt alapján minden valószínűségi szint mellett elvethető, a változók között közepesnél erősebb szignifikáns kapcsolat áll fent (4. melléklet, 18. táblázat és 19. táblázat).

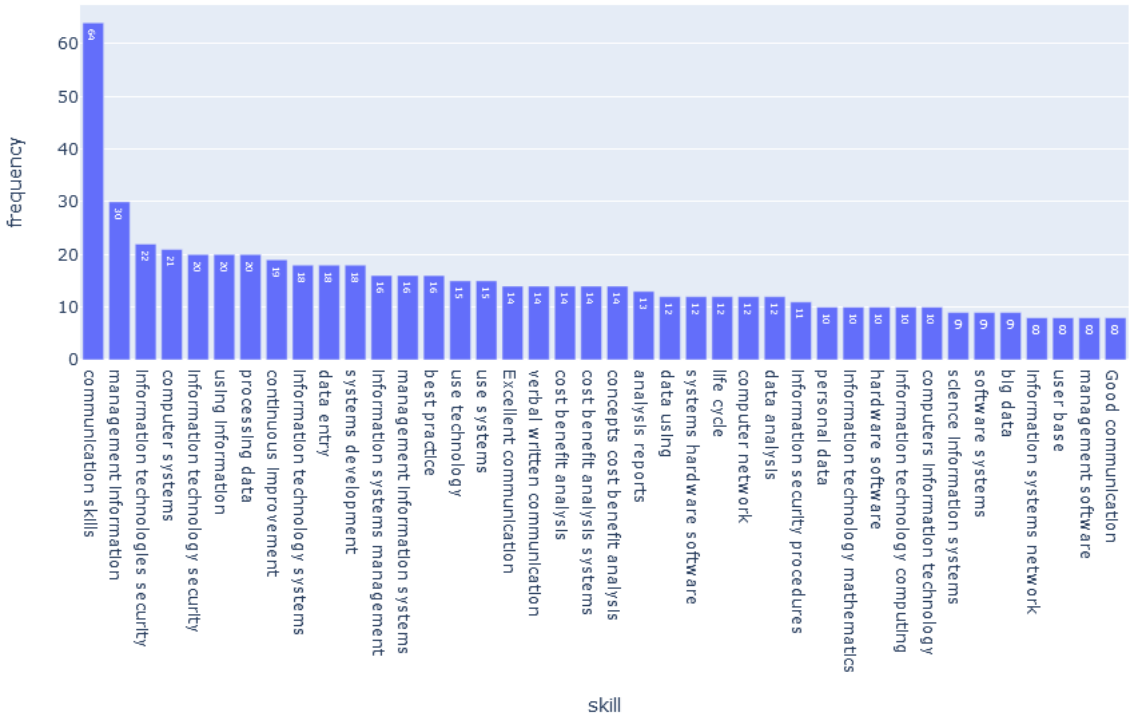
A tesztelésre használt hirdetésekben 894 különböző kifejezést fogadtunk el kompetenciaként[44], melyek közül a leggyakrabban megjelenő 40-et mutatja a 23. ábra. Az ábrán látható, hogy sok megjelenő kompetencia vagy tudáselem egy szóból áll és egy az egyben megfeleltethető egy kompetenciaszótári elemnek. Ezért információtartalom szempontjából érdekesebb azon kifejezések vizsgálata, melyeket a szótár nem tartalmaz.

[44] A teljes listát, annak mérete miatt sem a főszövegben, sem a mellékletben nem közlöm, azonban elérhető a dolgozathoz tartozó GitHub repozitóriumban: https://raw.githubusercontent.com/gneusch/phd_results/main/logit_results/test_set_identified_skills.txt



23. ábra: A tesztadatok halmazában kompetenciaként elfogadott, 30 leggyakrabban megjelenő kifejezés

A kifejezések között 786 olyan található, amely nem egyezik meg teljesen egy kompetencia-szótárbeli elemmel sem. A 40 leggyakoribbat ezen szókapcsolatok közül a 24. ábra mutatja. A leghosszabb így beazonosított kifejezés 4 szóból áll, míg a szókapcsolatokban található elemek számának módusza 2.



24. ábra: A tesztadatok halmazában kompetenciaként elfogadott, és egyetlen szótári elemmel sem közvetlenül megegyező 30 leggyakoribb kifejezés

6.2 Eredmények értékelése és a további kutatási irányok

A bemutatott *logit* modell a tesztadatokon elfogadhatóan teljesített; a felidézési arány 85%, míg a precizitás 71,9%. Mivel a folyamatba való manuális beavatkozás kezdetben elkerülhetetlen, azaz mielőtt elfogadhatnánk ezeket a kompetenciajelölteket valós kompetenciaként, egy szakértőnek át kell néznie az eredményeket, így a modell elfogadható, mint ami hasznos információkkal tud szolgálni és hozzáadott értékkel bír. A modellt abból a szempontból is elfogadom, hogy megfelelő választ ad a második kutatási kérdésemre, és alkalmas az explicit megjelenő kompetenciák szignifikáns részének beazonosítására.

A teszteredmények értékelése során derült fény arra, hogy a tanítóhalmazban csak olyan egy szóból álló kifejezések kerültek „találatként" felcímkézésre, melyek egy az egyben megfeleltek valamelyik kompetenciaszótári elemnek. Az unigramok esetében azért csak a teljes egyezést fogadtam el pozitív kimenetként, mert a tanítóadatok közé a mintának ebből a halmazából csak nagyon általános kifejezések kerültek. Azt a logikát próbáltam követni a címkézés során, hogy egy specifikusabb tartalmú kompetenciaszótári elem alapján nem következtetek egy generikusabb tudásterület vagy egyéb számunkra kompetenciát reprezentáló entitás ismeretére utaló elvárásra. Továbbá az általánosabb koncepciókat jelölő szavak esetében, mint például „*data*" vagy „*communication*", azzal a feltételezéssel éltem, hogy az ilyen kifejezésekre mutató találatok csekély információtartalma miatt, a tananyagfejlesztésben támogatni kívánt szakemberek ezeket nem tudnák használni, mivel a feladat célja szempontjából hozzáadott értéket nem adnak.

Ez a tanítóhalmaz sajátosságaiból adódó meghatározottság azt eredményezte, hogy a modell ezt a mintát megtanulta. Ezt jól mutatja, hogy azok között a teszthalmazban feltárt kifejezések között, melyeket kompetenciaként a modell alapján elfogadtunk, és emellett nem felelnek meg egyetlen szótárbeli elemnek sem, nincsen egyetlen unigram sem. Ez a jelenség elsősorban a tulajdonnevek, például az egyes technológiák megnevezései esetében jelent problémát, hiszen számtalan egy szóból álló technológianév létezik és a modell célja az, hogy hosszútávon rámutasson az újonnan megjelenő trendekre is. Ilyen szempontból tehát az eredményeket, a feltárható információk körét továbbra is nagyban meghatározza a kiindulási kompetenciaszótár minősége és teljessége.

Az előző problémára megoldásként, illetve általánosan a modell teljesítményének javítására több utat is látok, melyeket a kutatás későbbi szakaszaiban implementálni kívánok.

- Az első és legkézenfekvőbb a kompetenciaszótárban található elemek körének bővítése. Ehhez az egyik legfontosabb addicionális forrás az O*NET adatbázisa lehet, mely számtalan készség-, képesség- és tudáselemet tartalmaz és szabadon elérhető alkalmazásprogramozási interfészt nyújt az információk eléréséhez. A szótár természetesen az álláshirdetések feldolgozása során kompetenciának elfogadott elemekkel is bővíthető, ennek a modellre gyakorolt hatása azonban további vizsgálatot igényel.
- A 6.1.4. alfejezetben ismertetett kísérlet előfeldolgozási lépése során a szövegeket mondathatáron tagoltam, és minden a szöveget azon belül tovább tagoló, strukturáló karaktert eltávolítottam, mielőtt a szótári elemekkel való összevetéshez szükséges szóenneseket generáltam. A kifejezések környezetére vonatkozó információk így a modellben nem jelentek meg. A pontosabb eredmények elérésének érdekében a kutatás következő lépéseinek egyike, hogy a szöveget ne csak mondathatáron, hanem mondaton belül is tagoljuk, és ennek eredményeire építve haladjunk tovább, egyre több strukturális információt felhasználva.
- Mint ahogy az a gyakoriságra, illetve a *tf-idf* értékek vizsgálatára épülő modellből jól látszik (6.1.1 alfejezet), illetve Nasir és szerzőtársai (2020) is megerősítik az álláshirdetésekből képzett szóennesek között jelentős mennyiségben találhatók jelen feladat szempontjából zajként felfogható, a munkaadói ajánlatot illetve értékpreferenciát leíró elemek. Ezen megfigyelések előzetes kiszűrése javíthatja a modellek illeszkedését, teljesítményét. A kutatás későbbi szakaszaiban a hirdetések belső struktúrájának feltárására és a releváns szakaszok beazonosítására nagyobb figyelmet kívánok fordítani. Ennek érdekében olyan előfeldolgozási lépéseket implementáltam, amelyek használatával beazonosítható azon szakaszok kezdete, ahol általában az elvárások megjelennek, például valamilyen jellemző kifejezéshez kötve, mint amilyen a „*requirements, job description, what we need from you, the candidate possesses*” stb. Bár a kezdeti tesztek bíztató részeredményeket produkáltak, a teljes elemzés erre alapuló megismétlésére jelen

tézis véglegesítéséig nem nyílt alkalmam, a kapcsolódó forráskód azonban megtekinthető a dolgozathoz kapcsolódó GitHub repozitóriumban[45].

- Vizsgálni szeretném továbbá azt, hogy a szavak száma alapján particionált adathalmazokra épített modellekkel milyen eredmények érhetőek el. Az adatok megbontása ezen az elven arra is lehetőséget adhat, hogy a kifejezések egyes szavaihoz tartozó POS tageket eredményesen szerepeltessük a modellekben, hiszen így a lehetséges kategóriák száma jelentősen csökkenthető.
- A szóhatárokon megbontott kiindulási adatokra épülő uni- és bigram modellek esetében a karakterek száma alapján alkotott kategóriák magyarázó változók közötti szerepeltetését is érdemes lehet megvizsgálni.
- A kevesebb tagból álló kifejezések esetében a névelem (*NER*) információk felhasználásával az új, vagy a szótár alapján ismeretlen technológiák is felismerhetővé válhatnak.
- Az ismertetett modellben az egyes kifejezések akár több kompetenciaszótár-elemmel való kisebb-nagyobb hasonlóság miatt is a mintába kerülhetnek, akár többször is. Ezek a megfigyelések a hasonlóság mértékétől, a szótári elemben szereplő szavak számától stb. függően más-más valószínűségi értékeket is kaphatnak. Az egyes esetek valószínűségei alapján célszerű lenne egy, a kifejezések elfogadhatóságát önállóan meghatározó valószínűségi érték kalkulálása.
- A fenti javaslatok közül a névelemek felismerésére, a POS tagekre, illetve a hasonlósági alapú metrikákra épülő megoldások akár külön részrendszereknek is tekinthetők. Az ezek által függetlenül generált relevanciaérték közös értékelésére szintén kidolgozhatóak megoldások, mint ahogy hasonló javaslattal él Gugnani és Misra (2020) is.

Végül, de nem utolsósorban magas prioritással szerepel a jövőbeli kutatási irányok között a modell adaptálása magyar nyelvre. Vadász és Simon (2019) több morfológiai annotációs sémát és címkekészletet ismertet a magyar nyelvhez. Jelen kutatás szempontjából az egyik legfontosabb a Szeged Dependenciakorpusz (*Szeged*

[45] https://github.com/gneusch/phd_results/blob/main/cleaning_descriptions/clean_posting_descs.ipynb

Dependency Treebank) amely elérhető és felhasználható a *Stanford NLP Group Stanza* rendszerén keresztül is (Vincze *et al.*, 2009, 2010).

7 Látens kompetenciák feltárása

Szövegbányászati és NLP módszerek, illetve a rájuk épülő gépi tanulási algoritmusok, mint például az előzőekben bemutatott logisztikus regresszió segítségével beazonosítható az álláshirdetések leírásában explicit megjelölt kompetenciák egy része. Ezek, amennyiben összekapcsolhatók az adott pozíció által meghatározott munkakörrel vagy foglalkozással is, úgy nem csak az adott állásajánlatban leírt, nyitott pozícióhoz kapcsolódóan explicit megjelenő kompetenciák gyűjthetőek össze, hanem a pozíció és a betöltéséhez szükséges kompetenciahalmaz közötti kapcsolat – egy ontológia segítségével – magasabb, „absztraktabb", általános szinten is megadható. Minél több állásajánlatot dolgozunk fel egy adott munkakörtípushoz kapcsolódóan, feltehetjük, hogy annál pontosabban tudjuk majd leírni a kapcsolódó pozíciók hatékony betöltéséhez szükséges kompetenciák körét, illetve a szükséges kompetenciák időbeli változását, amennyiben a kapcsolat leírását idődimenzióval is kibővítjük. Mindezt akkor fogadhatjuk el, ha feltesszük, hogy az álláshirdetésben leírt pozíció ellátásához implicit szükségesek a kapcsolódó munkakör vagy foglalkozás által kijelölt, de az adott kontextusban nem megjelenő kompetenciák is.

A kapcsolat egy adott pozíció és a betöltéséhez szükséges teljes, explicit nem feltétlenül megjelenő kompetenciahalmaz között az előzőek alapján tehát úgy is feltárható, ha a hirdetés valamely attribútuma alapján a kapcsolódó foglalkozást egyértelműen be tudjuk azonosítani. Ekkor ugyanis újrafelhasználhatóak a már említett külső, szemantikus források és sztenderd nomenklatúrák, mint például az ESCO ontológia, vagy az O*NET adatbázisa, melyekből kigyűjthetők az adott foglalkozáshoz általában szükséges kompetenciák, illetve visszacsatornázhatók a folyamatba iteratív módon saját, a feldolgozás pillanatához viszonyítva múltbéli adataink is, melyeket információt hordozó elemként elfogadtunk.

A külső források „szemantikus" volta, a kapcsolatok definiáltságában rejlő hozzáadott értéke ebben az esetben ténylegesen kihasználható, a foglalkozás felől induló élek mentén azonosítva be a pozíció kontextusában relevánsnak ítélt elemeket, ellentétben azzal mikor csak sima kompetenciaszótárként funkcionáltak. A megfeleltetést nem fogadhatjuk el teljesen pontosnak, hiszen például előfordulhat, hogy egy-egy pozíció betöltéséhez, bizonyos specifikus esetekben a sztenderdtől eltérően több vagy más jellegű kompetencia szükséges, azonban céljaink szempontjából ezzel a módszerrel egy jó közelítést és hasznos információkat nyerhetünk; hiszen az általánosan az adott

pozíciótípushoz kapcsolódó kompetenciakör minden egyes feldolgozott állásajánlattal tovább bővíthető a megjelenő új elemekkel. **Ezekre, az ontológiára épülő következtetés alapján beazonosítható tudáselemekre, készségekre stb. hivatkozom tehát implicit, vagy látens kompetenciaként a dolgozatban**. Jelen fejezetben, a disszertáció harmadik kutatási kérdését vizsgálva, azokat a kísérleteket mutatom be, melyeket az implicit kompetenciák feltárása érdekében végeztem.

7.1 Implicit kompetenciák feltárása a foglalkozáson keresztül

Az implicit elemek feltárásának egyik módja tehát, ha a külső szemantikus forrásokban szereplő foglalkozásokhoz tudjuk egyértelműen rendelni az egyes hirdetéseket, mivel így a kapcsolódó, explicit meg nem jelenő kompetenciaelemek ismeretének szükségességét is el tudjuk fogadni. A foglalkozás-hirdetés kapcsolat feltárásának két irányát látom, melyeket meg kívánok vizsgálni.

1) Az adott foglalkozás vagy szerepkör az álláshirdetés címében vagy leírásában azonosítható, és valamilyen technikával, például lexikográfiai vagy szemantikus hasonlóság alapján egyértelműen hozzákapcsolható a foglalkozásontológia egy eleméhez.

2) Az álláshirdetések tartalmuk alapján foglalkozási kategóriákba rendezhetők, ami alapján kapcsolatuk az ontológiához meghatározható.

7.1.1 Foglalkozások beazonosítása az álláshirdetések címében

Amato és szerzőtársai (2015) több módszerrel vizsgálták foglalkozások beazonosíthatóságát hirdetések címében. Szakterületi szakértők segítségével beazonosítottak 412 álláshirdetést melyeket manuálisan felcímkéztek, annak megfelelően, hogy azok melyik foglalkozásnak felelnek meg a CP2011 (*Classificazione Delle Professioni*) olasz foglalkozási nomenklatúrában. Majd egy kereskedelmi forgalomban elérhető szabály alapú rendszert, illetve több gépi tanulási algoritmust, úgymint lineáris tartóvektor-gép (*Support Vector Machine*), perceptron osztályozó, LDA (*Latent Dirichlet Allocation*) teszteltek, hogy milyen mértékben közelíthető velük a manuális besorolás. Bár a módszerek használatát a szerzők mélységében nem részletezik, de a közölt eredmények alapján elmondható, hogy a felidézés és a precizitás átlagosan az LDA esetében 50% körül, míg a többi módszer esetében 25-35% között alakult.

A dolgozat céljának megfelelően kísérleteim során azt vizsgáltam, hogy automatikus eszközökkel miként azonosíthatóak be foglalkozások az álláshirdetések címeiben[46]. Először felépítettem egy egyszerű reguláris kifejezéseket használó szabályalapú módszert, hogy a foglalkozások neveit megtisztítsam az esetleges prefixumoktól és szuffixumoktól, azt vizsgálandó, hogy az ilyen módon megtisztított kifejezések és a használt ontológiák elemei között tapasztalható-e teljes, illetve tartalmazáson alapuló szövegegyezés. Ezek után azon hirdetések egy halmaza esetében, amelyeket ebben az első lépésben nem sikerült foglalkozáshoz társítani, azt vizsgáltam meg, hogy lexikográfiai- és „kvázi-szemantikai" hasonlóságmetrikák segítségével milyen eredményeket nyújtó döntési fán alapuló modellt lehet felépíteni. Kísérleteimet az előzőekben már használt 2019 októberi álláshirdetések (22213 egyedi dokumentum) adataival végeztem, melyeket az ESCO és az O*NET[47] ontológiákból leszűrt 2758 releváns – azaz informatikai területhez kapcsolódó – foglalkozás-megnevezéssel vetettem össze[48].

7.1.1.1 Foglalkozások beazonosítása reguláris kifejezésekkel

A felépített szövegfeldolgozási folyamat azon a megfigyelésen alapul, hogy a hirdetett pozíciók megnevezése általában egy jól meghatározható mintát követ. Ez a séma a következő „formulával" ragadható meg: „prefixum + foglalkozásra utaló általános kifejezés + szuffixum"; ahol a prefixum általában a szenioritási szintre utal (például „*senior*" vagy „*medior*"), vagy ritkábban valamiféle absztraktabb cím (mint „*director of*" stb.) míg a szuffixum egy munkavállalói kategóriát (pl. „*intern*", „*officer*" stb.) jelez[49]. A kiépített megfeleltetési folyamatot a pozíciónevek és az ontológiában szereplő foglalkozások felbontása után a középső, foglalkozásra utaló lényegi részek összehasonlítására alapoztam. Természetesen a pozíció-megnevezések ilyetén felbontása implicit azt a feltételezést is hordozza, hogy az igényelt kompetenciák köre a prefixumok és a szuffixumok mentén megegyezik, de legalábbis jelentősen átfed.

[46] A fejezetben bemutatott kísérletekhez írt forráskódok megtalálhatók a disszertációhoz tartozó GitHub repozitóriumban: https://github.com/gneusch/phd_results/blob/main/occupation_dict/occupation_analysis.ipynb

[47] O*NET Web Services are sponsored by the U.S. Department of Labor, Employment and Training Administration (USDOL/ETA), and developed by the National Center for O*NET Development.

[48] A foglalkozások listája, illetve az összegyűjtésükhöz használt kódok elérhetőek a dolgozat GitHub repozitóriumából: https://github.com/gneusch/phd_results/tree/main/occupation_dict

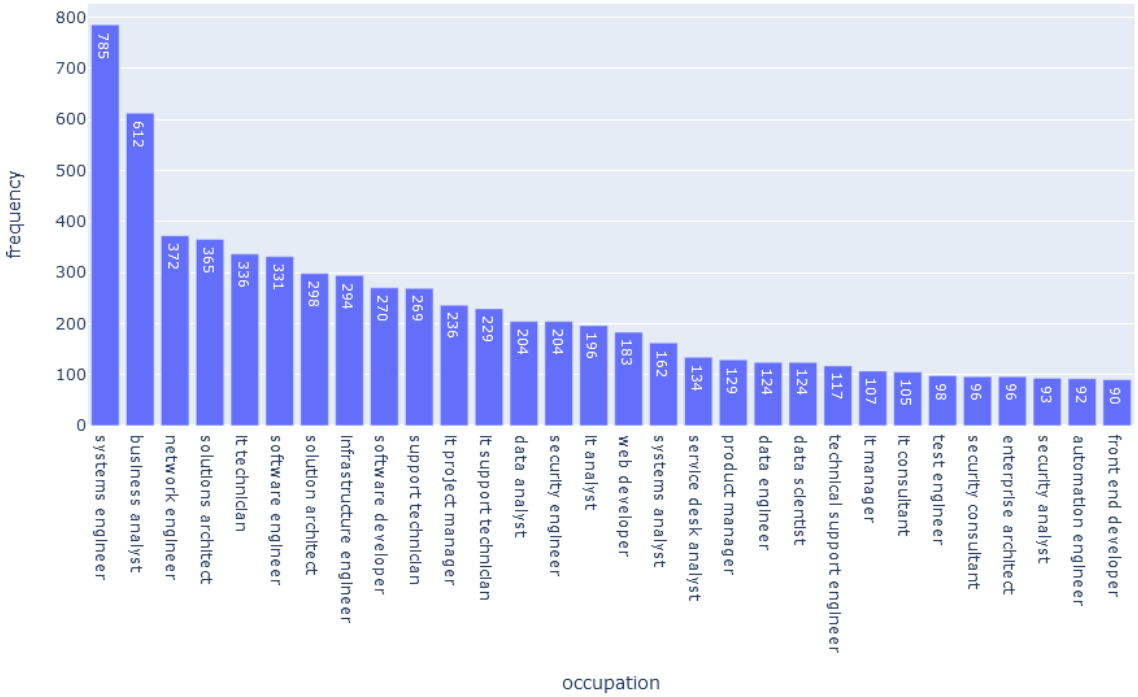
[49] Az így kinyert szuffixumok és prefixumok részben átfednek hirdetésekhez tárolni kívánt változókkal, mint foglalkozás típusa (*employment_type*), tapasztalat (*experience_level*) és pozíció típusa (*job_type*) (lásd 3. táblázat: Az álláshirdetésekből tárolni kívánt adatkörök).

Ennek a feltételezésnek a vizsgálatára és bizonyítására kutatásom egy későbbi szakaszában kívánok visszatérni.

Összesen 30 különböző prefixumot és 28 szuffixumot különböztettem meg a hirdetésekben, melyek listáját a 5. melléklet tartalmazza. Továbbá a pozíciók megnevezéseit egyes írásjelek mentén megbontottam, és ezeket az elemeket a feldolgozás során külön kezeltem, így előfordult néhány esetben, hogy egy hirdetéshez több prefixum, szuffixum, illetve „azonosító kifejezés" is társult. A megfeleltetési folyamat a következő lépéseket követte sorrendben, ahol az egyes lépések során az előző szakaszokban már foglalkozáshoz társított hirdetéseket nem vizsgáltam tovább.

1. Teljes egyezés vizsgálata előfeldolgozás nélkül (1668 találat, 7,5%).
2. Hirdetések címének felbontása írásjelek mentén, majd teljes egyezés vizsgálata az ontológiákból leszűrt foglalkozáscímkék felbontása nélkül (898 találat, 4%).
3. Részleges egyezés vizsgálata, a hirdetés címe tartalmazza a foglalkozás eredeti megnevezését (2716 találat, 12,2%).
4. Teljes egyezés vizsgálata mind a hirdetéscímek, mind az foglalkozáscímkék felbontása után (978 találat, 4,4%).

A fenti feldolgozási lépések során a beazonosítás pontosságának esélye minden lépéssel csökken. Összesen 16499 hirdetés, azaz a teljes populáció 74%-a követte valamilyen formában a mintát, ami alapján a felbontást elvégeztem, melyből 5282 (23,7%) elemet tudtam egyszerű szabályok felállításával és reguláris kifejezések használatával foglalkozáshoz kapcsolni, melyek közül a 30 leggyakoribbat a 25. ábra mutatja. Az eredményt manuálisan ellenőriztem és 97.5%-ban fogadtam el, hogy a fent ismertetett módszerrel beazonosított foglalkozás valóban megegyezik azzal, amit a hirdetésben keresett pozíció kijelöl.



25. ábra: Reguláris kifejezések és egyszerű szabályok segítségével beazonosított 30 leggyakoribb foglalkozás

7.1.1.2 Foglalkozások beazonosítása hasonlósági metrikák és döntési fa segítségével

A hirdetések azon populációjának egy véletlenszerűen leszűrt részét, melyet az előző alfejezetben leírt kísérletek során nem sikerült foglalkozáshoz rendelnem, döntési fa modellel kívántam tovább vizsgálni.

A döntési fák a példák alapján felügyelt tanulási módszerek csoportjába tartoznak, ugyanis a használt magyarázó attribútumok alapján, döntési szabályokon keresztül alkotnak egy hipotézist – jelen esetben, azaz diszkrét értékkészletű célfüggvény esetén – a megfigyelések osztályokba sorolására, azaz egy jóslást adnak a célváltozó értékére vonatkozóan. Egy döntési fa a kiinduló gyökérelemből, belső csomópontokból, és levélelemekből áll. Minden belső csomópontban egy attribútum szelekciós módszer segítségével kiválasztott magyarázó változóra vonatkozó tesztet végzünk el, a tovább induló élek mentén pedig a megfigyeléseinket a teszt eredményének megfelelően megbontjuk. Azaz jelen esetben egy, a bináris célváltozóra vonatkozó, a vizsgált változó alapján meghozott döntést végzünk el. A vizsgálatot addig folytatjuk, amíg vagy minden példánkat nem tudjuk egy döntéssel egyértelmű osztályokba sorolni, vagy valami előzetes korlátozó feltétel nem teljesül. Ilyen feltétel lehet a fára vonatkozóan például annak maximális mélysége, a levélelemek maximális száma stb. A levélelemek tartalmazzák az osztályba sorolás végső eredményét (Russell és Norvig, 2005).

A modell célpredikátumaként, az osztályozás alapjául a hirdetések címéből és a két felhasznált foglalkozásontológia foglalkozás-megnevezéseiből, az előzőekben részletezett módon generált felbontásainak megfelelőségét választottam, tehát a modellel egy diszkrét értékű osztályozási feladatot kívántam elvégezni. Azt vizsgáltam tehát, hogy egy adott nomenklatúraelemhez kapcsolódó kompetenciák szükségesek lehetnek-e az álláshirdetésben leírt pozíció ellátásához, annak címe, illetve tartalma alapján. Az adathalmazt manuálisan címkéztem fel, így a modellben legjobb szándékom ellenére szükségszerűen megjelennek saját előfeltevéseim, és bizonyos témák részletes ismeretének hiányából adódó félreértéseim vagy torzításaim.

7.1.1.2.1 A modell magyarázó változói

Magyarázó változókként a 6.1.3. alfejezetben leírt hasonlósági mérőszámokat, illetve két – az összehasonlítandó kifejezések távolságát a keresőmotorokban elért találataik száma alapján megragadó (*Search Engine Distance*) – metrikát használtam. Ez utóbbi metrikák arra a megfigyelésre épülnek, hogy azok a kifejezések, melyek hasonló jelentéssel bírnak, gyakrabban fordulnak elő együtt webdokumentumokban. Az első és legelterjedtebb ilyen metrika, a *Normalizált Google Távolság* (*Normalized Google Distance, NGD*), szintén Cilibrasi és Vitányi nevéhez köthető (2007). Munkájuk bevezetőjében a szerzők úgy fogalmazzák meg annak létjogosultságát, hogy szavak és kifejezések szemantikus távolságát egyes keresőmotorokban egymáshoz viszonyított találati aránya alapján határozzunk meg, hogy azok jelentésüket igazából társadalmi kontextusban szerzik meg az által, ahogy az emberek használják őket (p. 1). A *Normalizált Google Távolságot* a következőképpen kalkulálhatjuk:

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}}$$

Ahol $f(x)$ és $f(y)$ azon oldalak száma, melyek tartalmazzák x és y keresett kifejezéseket, illetve N a keresőmotor által indexált szavak maximális száma[50], melyet a gyakorlatban a „*the*” keresőszóra adott találatok számában határoznak meg. A két kifejezés annál távolabb van egymástól, minél nagyobb *NGD*.

Megvizsgáltam továbbá még egy együttes előforduláson alapuló metrikát, melynek alkalmazásához nem szükséges az összes indexált dokumentum számának a priori ismerete, mivel azt tapasztaltam, hogy ez a szám több keresőmotor esetében is kvázi

[50] Kísérleteim során a Microsoft Azure Bing Search alkalmazásprogramozási interfész (*API*) szolgáltatását használtam, ahol *N* értékének 9,500,000,000 adódik.

állandó, és nem tükrözi valójában az összes dokumentum számosságát, illetve egyáltalán nem követi annak változását. Nunes doktori disszertációjában ismerteti az „együttes előforduláson alapuló” (*Co-occurrence Based Measure, CBM*) metrikáját (2014), mely a következő módon számolandó:

$$CBM(x,y) = \begin{cases} 0, ha\ f(x) = 0\ vagy\ f(y) = 0\ vagy\ f(x,y) = 0 \\ 1, ha\ f(x) = 1\ vagy\ f(y) = 1\ vagy\ f(x,y) = 1 \\ \frac{\log f(x,y)}{\log f(x)} \times \frac{\log f(x,y)}{\log\ f(y)}, minden\ más\ esetben \end{cases}$$

A fenti képlettel a kifejezések távolságát a [0,1] intervallumon adjuk meg, ahol az 1 érték rendkívül szoros kapcsolatot jelez. Ebből kiindulva én a fenti képletet kissé átdolgozva használtam, mert úgy gondolom, hogy hibás 1 értéket visszaadni, amennyiben valamelyik keresőkifejezés 1-gyel tér vissza, habár értem a mögöttes matematikai megfontolást. Hiszen a fentiek alapján, amennyiben $f(\text{“}it\ service\ desk\text{”}, \text{“}it\ information\ and\ knowledge\text{”}) = 1$, az azt jelentené, hogy a két kifejezés rendkívül szoros kapcsolatban van egymással, holott a helyzet valószínűsíthetően pont ellentétes, hiszen összesen 1 dokumentumban szerepeltek együtt a keresőmotor találatai között. Így saját implementációm során, amennyiben a keresés eredményeként 1-et kaptam vissza, úgy 1.1-gyel számoltam a továbbiakban a képlet harmadik esetét feltételezve, hiszen ez az érték egy jó nagyságrendi közelítést ad.

Összességében elmondható a keresési találatok számán alapuló távolságmetrikákról, hogy bár nagyon informatívak lehetnek, de kalkulációjuk rendkívül költséges. A leggyakoribb keresőszolgáltatók igen hamar kiszűrik, ha a kereséseinket böngészőt szimulálva próbáljuk végrehajtani, és akkor is letiltják erőfeszítéseinket, ha másodperces nagyságrendű várakozási időt iktatunk be két lekérdezés közé, így a keresési távolsághoz szükséges információk megszerzése csak a szolgáltatók által biztosított módokon lehetséges. Két kifejezés távolságának meghatározásához három lekérdezés szükséges, míg a szolgáltatók keresési interfészeik használatáért jellemzően a keresések számának arányában számláznak. Kísérleteim során a *Microsoft Azure Bing Search* alkalmazásprogramozási interfész (*API*) szolgáltatását használtam, ahol a számlázás 1000 lekérdezéses egységekben történik, azaz számlázási egységenként maximum 333 kifejezéspár távolsága számolható ki, ez mindössze pár hirdetés vizsgálatát teszi lehetővé. Ez természetesen azt is jelenti, hogy

bármennyire is vonzóak ezek a mutatószámok jelentéstartalmilag, használatuk egy éles rendszerben csak valamely szolgáltatóval történt megállapodás alapján, vagy üzleti alapon képzelhető el, ami erős korlátja lehet a későbbi keretrendszerben való implementációjuknak. Jelen kísérleteimet szigorúan a szükséges információ megszerzésére fókuszálva, az üzleti megvalósíthatósági szempontokat nem vizsgálva végeztem el.

7.1.1.2.2 Döntési fa modell illesztése

Többek között a költségvonzatok miatt is, csak azokat az eseteket vizsgáltam a felépített modellben, ahol a hirdetés címe és a foglalkozás címkéje a 7.1.1.1 fejezetben részletezett módon, a prefixumok és szuffixumok eltávolítása után legalább egy szóban megegyezett, azaz $S_{JACCARD}(x, y) > 0$ teljesült. Sok irreleváns hirdetést, melyeket az IT kategóriában tettek közzé, de valójában nem, vagy csak rendkívül marginálisan kapcsolódtak a területhez, kiszűrtem, és összesen 2239 megfigyelést fogadtam el és címkéztem meg olyan szemmel, hogy az adott foglalkozás, illetve a hozzá kapcsolódó kompetenciák relevánsak-e a hirdetett pozíció szempontjából. A megfigyelések 84,7%-ban vetettem el a fenti gondolatmenet alapján a foglalkozás relevanciáját.

Láthatóan a célváltozó osztályainak eloszlása nem egyenletes, azaz egy rendkívül kiegyensúlyozatlan (*imbalanced*) adathalmazt állítottam elő. Tekintve, hogy a döntési fa algoritmusok hajlamosak a domináns osztályok irányában elfogult, hibás modellek létrehozására, ezért ezt a problémát a kísérleteim során a kisebbségi osztályba tartozó megfigyelések felül-mintavételezésével kívántam ellensúlyozni. Módszertani szempontból a megfigyelések viszonylag alacsony száma miatt döntöttem emellett az eljárás mellett a többségi osztály alul-mintavételezésével szemben. Kísérleteim során teszteltem saját implementációmat – mely véletlenszerű helyettesítéses mintavételezésre a *Scikit-learn* (Pedregosa *et al.*, 2011) programcsomag „*resample*” függvényét használja – a modell teljesítményének mérésére különböző elemszámú felül-, illetve alul-mintavételezés mellett, illetve a *SMOTE* (*Synthetic Minority Over-sampling TEchnique*) algoritmus *Imbalanced-learn* (Lemaître *et al.*, 2017) programcsomagban implementált változatát is. A *SMOTE* algoritmus a véletlenszerű helyettesítéses mintavételezésen alapuló felül-mintavételezés helyett, ahol tulajdonképpen az eredeti pontok mintában való megismétlésére kerül sor, új „szintetikus” elemeket állít elő. Az egyes mintapontoknak a felül-mintavételezés

mértékének megfelelően a magyarázó változók vektorai alapján vesszük a k legközelebbi szomszédját, majd a minta és a szomszédok jellegzetességvektorainak különbségét megszorozzuk egy 0 és 1 közötti véletlen számmal. Az így kapott értékeket hozzáadjuk a mintapont független változóinak megfelelő elemeihez, így áll elő az új, szintetikus elem tulajdonságvektora (Chawla *et al.*, 2002).

Szintén a megfigyelések relatíve alacsony száma miatt a modellparaméterek finomhangolására és tesztelésére, illetve az egyes modellek predikciós hibájának becslésére a keresztvalidáció technikáját választottam a különálló tanító, validációs és teszt halmazok használata helyett. A keresztvalidáció nem csak arra jó, hogy kis elemszám mellett megszünteti a validációs halmaz iránti igényt, a modellparaméterek kiválasztása során, hanem Russel és Norvig (2005, p. 583) alapján a túlilleszkedést is segít csökkenteni, észrevenni. A technika lényege, hogy a tanító halmazt k egyenlő részre osztjuk, amiből $\frac{k-1}{k}$ résszel tanítjuk a modellünket, majd a fennmaradó $\frac{1}{k}$ résszel validáljuk annak teljesítményét. Ezt az eljárást megismételjük k összes variációjára, majd az eredmény átlagát tekintjük a modellünk teljesítményének. Mindezek után egy eddig nem használt teszthalmaz segítségével mérjük a kiválasztott modell erejét a tanítás során nem látott, új adatokon. Ezt az eljárást K-szoros (*K-fold*) keresztvalidációnak nevezik és elterjedten használják még speciális eseteit, mint a megismételt K-szoros keresztvalidáció, melynek során az eljárást l alkalommal megismétlik különbözően előállított k számú halmazokkal vagy a rétegzett mintavétellel készülő K-szoros keresztvalidációt. Az úgynevezett *Leave One Out* (*LOO*) keresztvalidáció esetében $k = n$.

Az implementáció során az adatok kiegyensúlyozatlansága miatt a tanító- és a teszthalmazokat rétegzett mintavételezéssel, keveréssel állítottam elő. A modellezést a már említett *Scikit-learn* szoftver segítségével hajtottam végre, amely az egyik legelterjedtebben használt és legátfogóbb gépi tanulási modelleket és a modellezéshez szükséges eszköztárat is tartalmazó programcsomag. Az alkalmazás lehetőséget ad feldolgozási láncok (*pipeline*) létrehozására, azaz modellezési lépések szekvenciális végrehajtására az adatokon. A *Scikit-learn* feldolgozási láncába egyszerűen bekapcsolhatók az *Imbalance-learn* alkalmazás szolgáltatásai is. A *Scikit-learn* segítséget nyújt továbbá a modellparaméterek állapotterének kimerítő tesztelésére (*hyper-parameter tuning*), és egy adott metrika, például a találati arány vagy az F mutató stb. alapján a legjobban teljesítő modell kiválasztására (*grid search*). A

paraméteroptimalizálás során a feldolgozási láncokban a keresztvalidáció alkalmazásával a kukucskálás (*data peeking*), azaz „a teszthalmaz által hordozott információ tanuló algoritmusba szivárgása” a folyamat során kizárható (Russell és Norvig, 2005, p. 582).

Kiinduló viszonyítási alapként a 0R (*ZeroR*), „triviális” osztályozót használtam, mely annak megítéléséhez hasznos, hogy a létrehozni kívánt döntési fa modellnek van-e hozzáadott értéke ahhoz az esethez képest, amikor az összes megfigyelésünket a leggyakoribb osztályba soroljuk (Bodon és Buza, 2013). Természetesen a 0R osztályozó esetében a találati arány megegyezik a többségi osztályhoz tartozó gyakorisággal, ami esetünkben 84,7% (9. táblázat tartalmazza az egyes modellek összehasonlítását). Ehhez képest az alapbeállításokkal és keresztvalidációval futtatott döntési fa modell találati aránya kicsit magasabb (90%) amellett, hogy az F mutató értéke, azaz a fedés és a pontosság harmonikus átlaga 69%. A modellt a teszthalmazon futtatva igazolta a keresztvalidációt, az eredmények rendre 90 és 70%-on alakultak. Ezek alapján elmondható, hogy bár láthatóan létezik valamiféle, a magyarázó változók alapján feltárható szabályszerűség a foglalkozás-megnevezések és a hirdetések címei között, de az alapmodell ezt csak közepes eredménnyel tudja megtalálni.

A *Scikit-learn DecisionTreeClassifier* osztálya alapbeállítás szerint vágási függvényként a Gini indexet használja. A Gini index egy szennyezettségi[51] (*impurity*) mérőszám, a vágás jóságát/hibáját mutatja. Értéke 0,5 ha a vágás során a célváltozó egyes csoportjaiból egyforma arányban kerültek be egyedek és 0 akkor, ha a vágás tökéletes, azaz a választott attribútum segítségével egyértelműen egy jól meghatározott osztályba sorolhatóak a megfigyelések. Azaz „minél kisebb a szennyezettség mértéke, annál ferdébb az eloszlás” (Tan *et al.*, 2011, p. n.a.). A Gini index kifejezhető egy adott halmazba tartozó elemek különböző osztályainak gyakoriságával, azaz J különböző osztály esetén, ahol $j \in \{1, 2 \dots k\}$ Bodon és Búza (2013, p. 151) alapján:

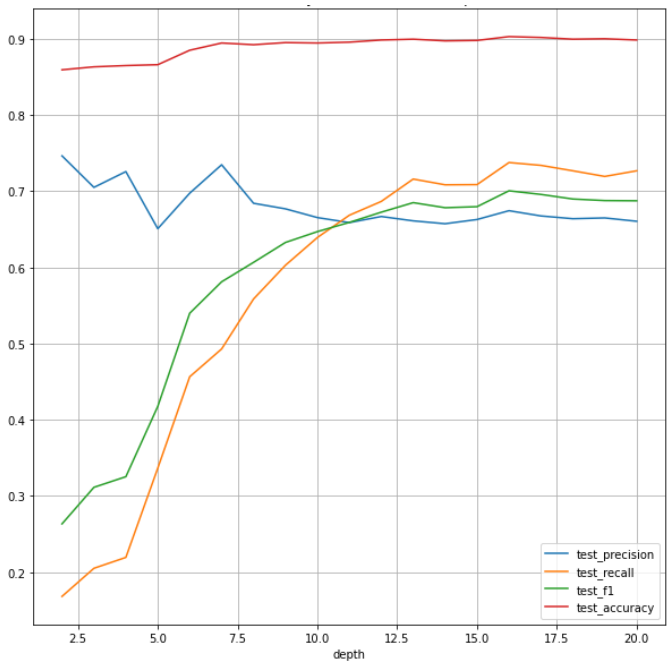
$$Gini(\mathrm{P}) = 1 - \sum_{j=1}^{k} p_j^2$$

A Gini index mellett vágási függvényként használatos még az információtartalmat (entrópia) vizsgáló információnyereség is, amely egy adott a attribútum tesztje előtti

[51] Szokásos még tisztasági mértéknek is fordítani.

információszükséglet és a teszt után szükséges maradék információszükséglet különbségeként írható le (Russell és Norvig, 2005, p. 580).

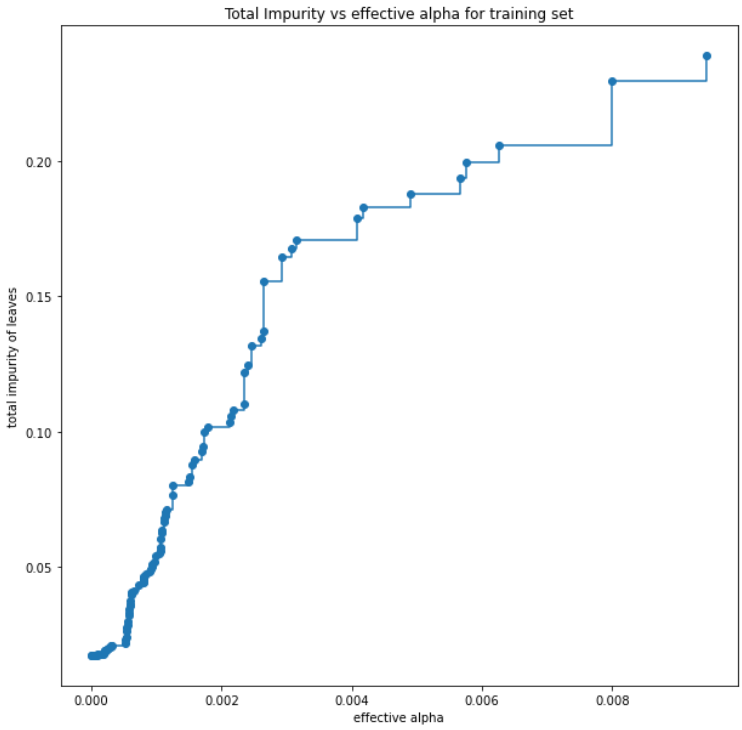
Alapbeállításként nincs semmi megkötés a további modellparaméterekre, például a fa méretére, vagy a levelenkénti találatok minimális számára vonatkozóan, azonban ezek szabályozására a szoftver természetesen lehetőséget ad. A 26. ábra mutatja a modell keresztvalidációval mért teljesítményértékeinek alakulását különböző mélységű döntési fák esetén. Az ábrán jól látszik, hogy a fa mélységi paraméterértékének növelésével a pozitív esetek felismerésének aránya növekszik, ezzel párhuzamosan, bár kisebb mértékben, a tévesen pozitívnak ítélt megfigyelések aránya is nő. Sajnos a fa méretének növelésével nem csak a modell tanulóhalmazon mért teljesítménye, de a túltanulás valószínűsége is nő, ahogy ez az ábrán is látszik, a modell teszthalmazon mért pontossága egyre kisebb ütemben nő a fa mélyítésével, majd egy bizonyos mélység után stagnálni kezd, és valószínűsíthetően a későbbiekben csökkenésnek indulna.



26. ábra: A modell teljesítményértékeinek alakulása különböző mélységű döntési fák esetén. Az ábra a Matplotlib alkalmazással készült (Hunter, 2007).

A túltanulás megelőzésére két stratégiát szokás alkalmazni. Egyrészt a döntési fa valamely paraméterére beállított küszöbértékkel megakadályozható, hogy a fa túl nagyra nőjön (korai leállás), másrészt a már elkészült fa valamilyen metszési (*pruning*) technikával visszavágható. A költség- vagy hibaarány-komplexitási metszés (*cost complexity pruning*) például egy gyakran alkalmazott utólagos döntési fa metszési

technika. A hibaarány lehet egy adott részfa esetében a rosszul osztályozott elemek aránya, míg a komplexitás mérhető például a levelek számával. Az egységnyi komplexitás és az egységnyi hiba költségének hányadosát α szimbólummal jelölik. Az α effektív, vagy kritikus, amikor adott értéke mellett egy belső csomópont költsége megegyezik a belőle induló részfa összetett költségével. Az a belső csomópont adja a leggyengébb vágást, melyhez tartozó α_{kritikus} érték a legkisebb, hiszen az ahhoz tartozó részfában csökken legkevésbé a hibaarány (Pataki, 2018). A *Scikit-learn DecisionTreeClassifier* osztályának azt a minimális α_{kritikus} (*ccp_alpha*) paramétert lehet megadni, aminél kisebb α_{kritikus} esetében egy adott csomópontoknál vágni szeretnénk. A 27. ábra az alapbeállításokkal futtatott döntési fa modell esetében alakuló α_{kritikus} értékeket mutatja. Látható, hogy számos csomópont esetében egészen kicsi érték szerepel, így érdemes lehet a modell *ccp_alpha* paraméter mentén is finomhangolni.



27. ábra: $\alpha_{kritikus}$ értékek alakulása a szennyezettség függvényében

Az legoptimálisabb modellparaméterek feltárása a kukucskálás elkerülése mellett bonyolultabb programozási problémát jelenthetne, azonban a fentebb említett módon a *Scikit-learn* által nyújtott feldolgozási láncok segítségével a feladat leegyszerűsíthető. A túlillesztés, azaz a tanítóhalmazra, annak sajátosságaira, hibáira való túlzott rátanulás elkerülése és a lehető legpontosabb modell megtalálása érdekében a mélység, a minimális levelenkénti mintaszám és a hibaarány-komplexitási

metszés effektív α paraméterének különböző értékei mentén kerestem az elérhető legjobb modellt, azaz azt a döntési fát mellyel maximalizálható a keresztvalidáció során elért eredmény. Vágási feltételként megvizsgáltam mind a Gini indexet, mind az entrópiát. A numerikus paraméterek tesztelt értékeit a 8. táblázat tartalmazza. A maximális mélység paraméter tesztelt értékeit a 26. ábra, míg α_{kritikus} 27. ábra alapján határoztam meg.

Attribútum	**Kiindulási érték**	**Végső érték**	**Lépésköz**
Fa maximális mélysége	2	15	1
Minimális elemszám levelenként	5	10	1
α_{kritikus}	0.0	0.003	0.0003
Kisebbségi és többségi osztály aránya felül-mintavételezésnél	0.3	1.0	0.1

8. táblázat: Numerikus paraméterek tesztelt értékei

Az értékelés (*scoring*) alapjául az F mutatót választottam, mivel a modellt a kapcsolódó foglalkozások megtalálására szerettem volna optimalizálni. A modellválasztáshoz K szoros keresztvalidációt használtam, k értékét 10-ben határoztam meg. Egyrészt mert a szakirodalom alapján k tipikus értékei az 5 és 10 (Hastie *et al.*, 2009, p. 242), másrészt mert saját, k különböző értékeivel végzett kísérleteim során azt találtam, hogy a tanító halmazon k nagyobb értékeinél pontosabb modellt kapunk, ami – tekintettel a tanító pontok relatíve kis elemszámára – viszonylag könnyen belátható.

Modell	**Találati arány**	**Precizitás**	**Felidézés**	**F mutató**
0R*	0.85	0	0	0
Alapbeállításokkal futtatott modell	0.9	0.67	0.73	0.70
Attribútumtér vizsgálattal[52], felül-mintavételezés nélkül	0.94	0.86	0.74	0.79
Attribútumtér vizsgálattal, véletlenszerű helyettesítéses felül-mintavételezéssel	0.93	0.74	0.91	0.82
Attribútumtér vizsgálattal, *SMOTE felül-mintavételezéssel*	0.93	0.74	0.87	0.8

9. táblázat: Az egyes modellek tanító halmazon, illetve keresztvalidációval elért eredményei*

A 9. táblázat alapján elmondható, hogy a felül-mintavételezéssel készült, keresztvalidáció során kiválasztott modellek az F mutató alapján jobban teljesítettek, mint az alapbeállításokkal futtatott modell, főleg a felidézés tekintetében, azaz

[52] A 8. táblázat különböző attribútumkombinációinak vizsgálatával futtatott modell.

arányaiban kevesebb esetet címkéztek tévesen negatív kimenetelűnek. Azonban a precizitás alapján látható, hogy a relevánsnak címkézett elemek több mint negyede valójában irreleváns ezen eljárások használata esetén. Ez alapvetően a cél szempontjából felfogható lenne jó eredménynek, hiszen valamennyi manuális szakértői beavatkozás a folyamatba egyelőre mindenképpen szükséges, amit a modell alkalmazásával láthatóan csökkenteni lehetne. Azonban a teszthalmazon mért eredmények csak az arányok tekintetében igazolják vissza a keresztvalidáció alapján levont következtetéseket (10. táblázat). A táblázatból jól látható, hogy a felül-mintavételezéssel készült modellek a teszthalmazon rosszabbul teljesítenek az alapmodellnél. Ez az eredmény azért lehetséges, mert a helyes osztályba tartozó megfigyelések arányának a tanító halmazban való növelésével az algoritmusnak több esélye van az esetlegesen zajos, kiugró értékekkel rendelkező pozitív megfigyelésekre való túlilleszkedésre. Az előző következtetést megerősíteni látszik, hogy a *SMOTE* eljárás alkalmazásával, azaz szintetikus tanulópontok létrehozásával jobb eredmények érhetőek el, mint a pozitív megfigyelések egyszerű, véletlenszerű ismétlésalapú felül-mintavételezésével, annak ellenére, hogy utóbbi a tanító halmazon marginális mértékben jobban teljesít.

Modell	**Találati arány**	**Precizitás**	**Felidézés**	**F mutató**
0R	0.85	0	0	0
Alapbeállításokkal futtatott modell	0.9	0.67	0.72	0.70
Attribútumtér vizsgálattal, felül-mintavételezés nélkül	0.88	0.67	0.53	0.59
Attribútumtér vizsgálattal, véletlenszerű helyettesítéses felül-mintavételezéssel	0.85	0.51	0.63	0.57
Attribútumtér vizsgálattal, *SMOTE felül-mintavételezéssel*	0.875	0.58	0.68	0.63

10. táblázat: Az egyes modellek teszt halmazon elért eredményei

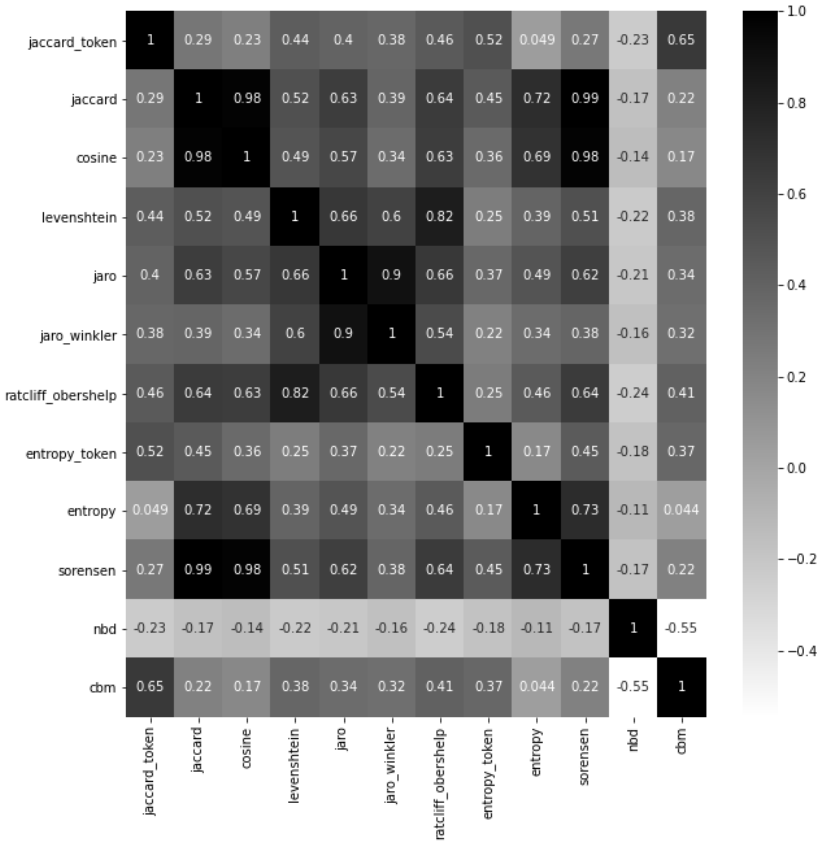
A 10. táblázat alapján tehát elmondható, hogy az alapmodell teszthalmazon mért teljesítményét az attribútumtér kimerítő vizsgálatával kiválasztott modellek segítségével nem sikerült javítani, azaz a felül-mintavételezés során az adatok közé bevitt zajt nem sikerült a modellparaméterek változtatásával ellensúlyozni. Az attribútumtér vizsgálata során kiválasztott modellek paraméterértékeit a 11. táblázat tartalmazza.

Modell	1/0 arány	Mélység	Minimum találat/levél	α_{kritikus}	Vágási függvény
Alapbeállításokkal futtatott modell	0.18	17	1	0.0	Gini
Attribútumtér vizsgálattal, felül-mintavételezés nélkül	0.18	14	5	0.0	Entropy
Attribútumtér vizsgálattal, véletlenszerű helyettesítéses felül-mintavételezéssel	0.7	13	5	0.0009	Gini
Attribútumtér vizsgálattal, SMOTE felül-mintavételezéssel	0.7	13	5	0.0	Gini

11. táblázat: A legjobban teljesítő modellek paramétereinek alakulása különböző modellezési megközelítések mellett

7.1.1.2.3 Magyarázó változók független komponenseinek használata

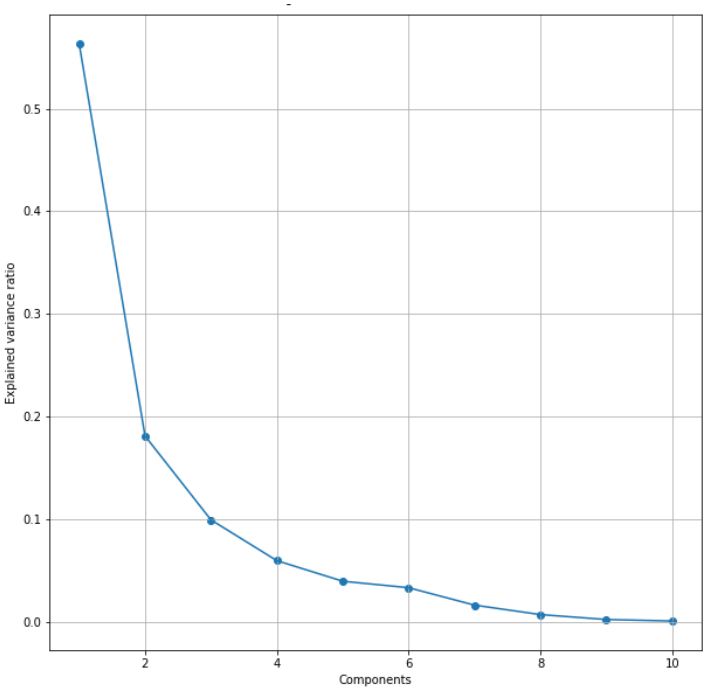
A modellezési folyamat előzőekben ismertetett gyenge eredményeit okozhatja egyebek mellett a független változók közötti erős korreláció is. Így a munka következő lépéseként ennek lehetőségét vizsgáltam meg, illetve azt, hogy a változók közötti esetlegesen magas korreláció kiiktatásával javítható-e a modell teljesítménye. A magyarázó változók között fennálló korrelációról ad képet a 28. ábra, melyről egyértelműen látszik, hogy egyes mérőszámok között, mint például a Jaccard index, a koszinusz- és a Sørensen távolságok vagy Jaro- és a Jaro-Winkler metrikák között egészen erős pozitív korreláció áll fent. Az ábráról egy meglepő összefüggés is leolvasható; a tokenek távolságán alapuló Jaccard index és a CBM mérőszám között is a közepesnél magasabb korreláció áll fönt, annak ellenére, hogy a kalkuláció alapja teljesen más. Az első egy lexikográfiai tulajdonságokon alapuló, míg utóbbi egy webdokumentumokban megfigyelt együttes előforduláson alapuló adat. Ugyan a korreláció nem implikál kauzalitást, és okozati összefüggést levonni hibás lenne, főleg egy ilyen kis elemszámú, rendkívül speciális mintából, de az elmondható, hogy a mintában szereplő feldolgozott pozíció- és foglalkozás megnevezések esetén a két változó között közepesnél erősebb lineáris kapcsolat fedezhető fel.



28. ábra: Korreláció a magyarázó változók között

Amikor, mint jelen esetben, a magyarázó változóink erősen korrelálnak, kiválasztható néhány, az elemzés szempontjából relevánsabb, de egymással nem korreláló változó, vagy „képezhetünk egymásra merőleges faktorokat, melyek független változóként használhatók" (Kovács, 2014, p. 148). A főkomponens-elemzés (*Principal Component Analysis, PCA*) módszer célja, hogy a sok egymással korreláló változóból, néhány egymással korrelálatlan főkomponenst állítson elő, amelyek közül az első néhány az eredeti változók varianciájának minél nagyobb részét magyarázza.

Az adataimat, főleg a *Normalizált Bing Távolság* értékei miatt, az elemzés előtt sztenderdizáltam, hogy minden változó értékeinek átlaga nulla és szórása egységnyi legyen. Hogy az adott minta megfelelő-e főkomponens-elemzésre, a Kaiser-Meyer-Olkin mértékkel vizsgáljuk, mely jelen esetben a közepes és a jó határán, 0,79 alakult. A Bartlett-teszt null-hipotézise, miszerint a változók között nincs korreláció, a khi-négyzet teszt alapján minden szokásos szignifikancia szinten elvethető, azaz a minta alkalmas a főkomponenselemzésre. Két változó, a *Normalizált Bing Távolság* és a token alapon számolt Entrópia esetében azonban a magyarázott hányad annyira kis mértékű, hogy azokat végül kihagytam az elemzésből.



29. ábra: A főkomponensek által magyarázott variancia hányada

A 29. ábra mutatja a főkomponensek által magyarázott variancia hányadának alakulását. Látható, hogy az első három főkomponens relatíve fontosabb, az összes varianciának 84,3%-át magyarázzák, de a harmadik főkomponens sajátértéke már valamivel 1 alatt alakul (0,99). A további elemzéshez ezt a harmadik komponenst még megtartottam.

	Component_1	**Component_2**	**Component_3**
jaccard_token	0.087	0.197	0.788
jaccard	0.954	0.213	0.181
cosine	0.950	0.175	0.138
levenshtein	0.357	0.537	0.420
jaro	0.388	0.850	0.224
jaro_winkler	0.137	0.892	0.211
ratcliff_obershelp	0.498	0.472	0.452
entropy	0.706	0.268	-0.054
sorensen	0.962	0.207	0.169
cbm	0.046	0.157	0.734

12. táblázat: A változók és a főkomponensek közötti korrelációk

A változók és a főkomponensek közötti korrelációkat szemlélteti a 12. táblázat. Látható, hogy az első főkomponenssel a karakter alapon számolt Jaccard, koszinusz és Sorensen hasonlósági mérőszámok mutatnak erős pozitív korrelációt. Hasonlóképpen a második komponenssel leginkább a Jaro és a Jaro-Winkler, míg a harmadik

komponenssel a token alapon számolt Jaccard és az együttes előforduláson alapuló CBM mutatók korrelálnak legjobban.

Az így kialakult három főkomponenssel, mint magyarázó változókkal szintén végrehajtottam a 8. táblázatban ismertetett paraméterek terében a legjobb modellt kereső eljárást. A legjobban teljesítő modell teszthalmazon elért eredményei szerényen alakultak, precizitása 56%-os, a felidézése 66%-os, míg az F mutató 62% volt.

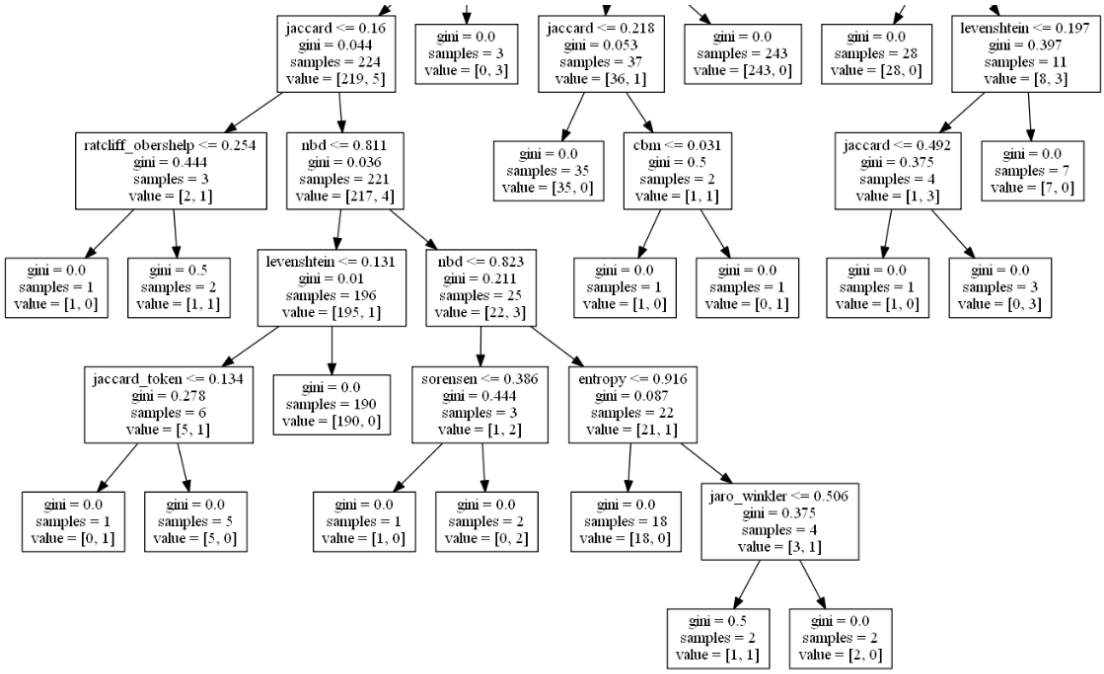
7.1.1.2.4 Értékelés

Általánosságban a döntési fákról elmondható, hogy minél komplexebbek, annál pontosabb lesz tanítóhalmaz esetében a modell, de annál valószínűbb a túltanulás esete, ami a teszthalmazon elért gyenge teljesítményben mutatkozik meg. Általánosságban az elfogadott módszerek a túltanulás kiküszöbölésére, a döntési fa metszése, illetve a korai leállás parametrikus szabályozása. Jelen esetben azonban minden beavatkozás a modellparaméterekbe javított ugyan a keresztvalidáció eredményén, de csak rontotta a modell tesztadatokon elért teljesítményét. A keresztvalidáció során rendre 90% körüli találati arányt felmutató modellek mindannyiszor jelentősen rosszabb eredményekkel teljesítettek a teszthalmazon, ami tehát a döntési fák túltanulását mutathatja, melynek hatását úgy tűnik sem a korai leállási feltételek szigorításával, sem a végső fa metszésével, sem felül-mintavételezéssel nem sikerült jelen esetben csökkenteni. A felidézés tekintetében elért gyenge eredmények – melyek azt mutatják, hogy a modellek a pozitív esetek jelentős hányadát nem tárják fel – arra utalnak, hogy azok csak korlátozottan lesznek használhatóak a látens kompetenciák feltárására.

A gyenge teljesítmény oka lehet az adatok kiegyensúlyozatlansága, a kisebbségi osztály elemei között található túl sok kiugró érték, vagy az is, hogy a választott magyarázó változók egyszerűen nem alkalmasak arra, hogy a megfigyelések esetében az egyes pozíció-foglalkozás kapcsolatokat hatékonyan előrejelezzék. Az is lehetséges továbbá, hogy a kapcsolat egy másik módszer használatával nagyobb hatásfokkal tárható fel. Ennek megfelelően az eredmények javítása érdekében a kutatás későbbi fázisaiban, amennyiben többlet erőforrást tudok bevonni a feldolgozásba, szeretném nagyobb elemszámú, lehetőleg szakértők segítségével felcímkézett adatokon megismételni a kísérleteket. Tervezem továbbá, hogy a foglalkozás és a pozíció közötti hasonlósági metrikákon alapuló kapcsolatot más módszerek – például logisztikus

regresszió, vagy tartóvektor gép (*Support Vector Machine, SVM*) – segítségével is megpróbálom feltárni.

Bár az összteljesítmény tekintetében az alapmodell teljesített legjobban, azonban a fa tanulmányozása során egyértelműen látszik, hogy az számos olyan levelet tartalmaz, melyek esetében az elemszám 1 vagy hasonló kis összeg. Ezt remekül mutatja a 30. ábra a komplexitás elhanyagolható részének szemléltetésével is.



30. ábra: Túltanulás a döntési fa modellben

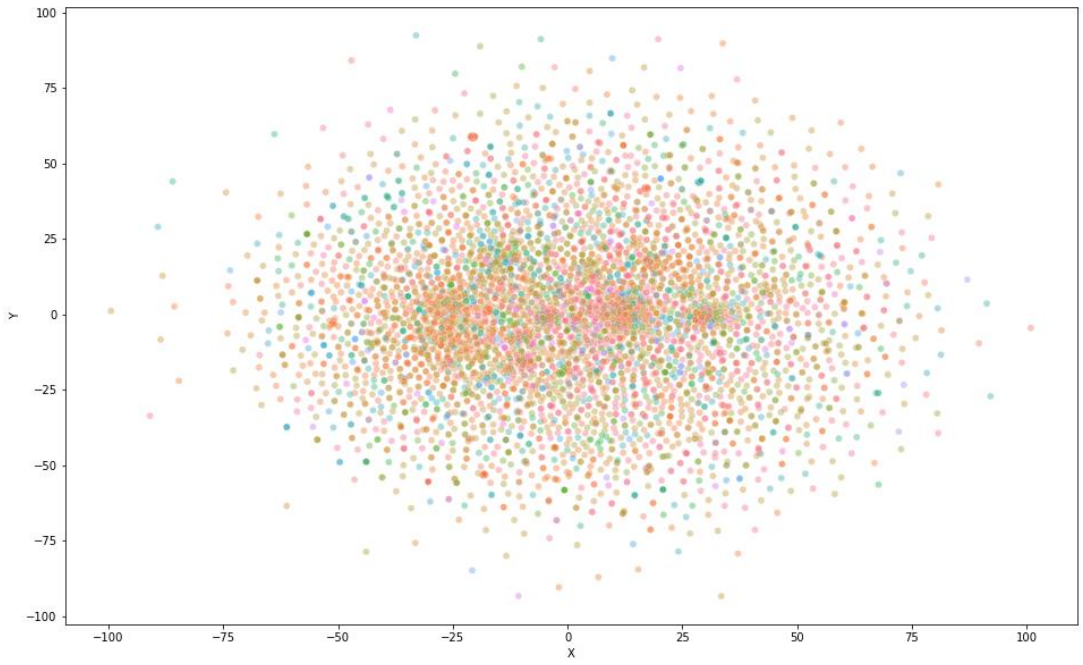
Az előzőekből kifolyólag tehát, a feladat céljainak szempontjából a *SMOTE* felülmintavételezési eljárással és a korai leállást irányító paraméterek szabályozásával készült, a teszthalmazon 68%-os felidézési értéket elérő modell tekinthető a legjobbnak.

7.2 Hirdetések témájának beazonosítása

A hirdetéshez kapcsolódó foglalkozás, vagy foglalkozási csoport beazonosítása akkor is hasznos lehet, ha azt egyébként nem tudjuk egyetlen, a szemantikus forrásainkban definiált elemhez sem kötni, hiszen az például kapcsolódhat egy, a piacon újonnan megjelenő igényhez. Illetve ezen, számunkra ismeretlen foglalkozásokon keresztül is tovább bővíthetőek, „finomíthatóak” az ontológiáink (*ontology refinement*). A hirdetésekből képezhető klaszterek feltárása ehhez a feladathoz nyújthat segítséget. Csepregi (2020) az álláshirdetések leírásaiból képzett *tf-idf* mátrix *LDA, LSA és K-*

közép klaszter elemzése során arra a következtetésre jutott, hogy az álláshirdetéseket témájuk szerint nem lehet elkülöníteni ezekkel a nem felügyelt osztályozási módszerekkel. Saját kísérleteim során *t-SNE* (*t-distributed Stochastic Neighbor Embedding*) módszer alkalmazásával meg tudtam erősíteni a szerző eredményeit.

A kísérletet a 7.1.1.1 pontban leírt módszerrel, reguláris kifejezésekkel és egyszerű szabályok alkalmazásával foglalkozáshoz rendelt, több mint 5000 hirdetésen végeztem. Arra voltam kíváncsi, hogy a hirdetések leírásának előfeldolgozása[53] után előállt adathalmaz alapján készült *tf-idf* mátrix kisebb dimenzióban történt vizualizációja segítségével elkülöníthetők-e a hirdetések jól megkülönböztethető csoportjai. Ezzel arra kerestem a választ, hogy érdemes-e a hirdetések témáját, azok leírásának numerikus reprezentációja alapján tovább keresni. A *t-SNE* módszer remekül használható erre a feladatra, mivel segítségével nagy dimenziószámú megfigyeléseket képezhetünk le két, három stb. dimenziókba úgy, hogy az adatok lokális struktúrájának jelentős hányadát megőrizzük, míg az eljárás a dimenziócsökkentés segítségével a globális struktúráról is tár fel, az ember számára is könnyen feldolgozható, új, vizuális információkat, például az adatok esetleges klasztereinek jelenlétét (Maaten és Hinton, 2008).



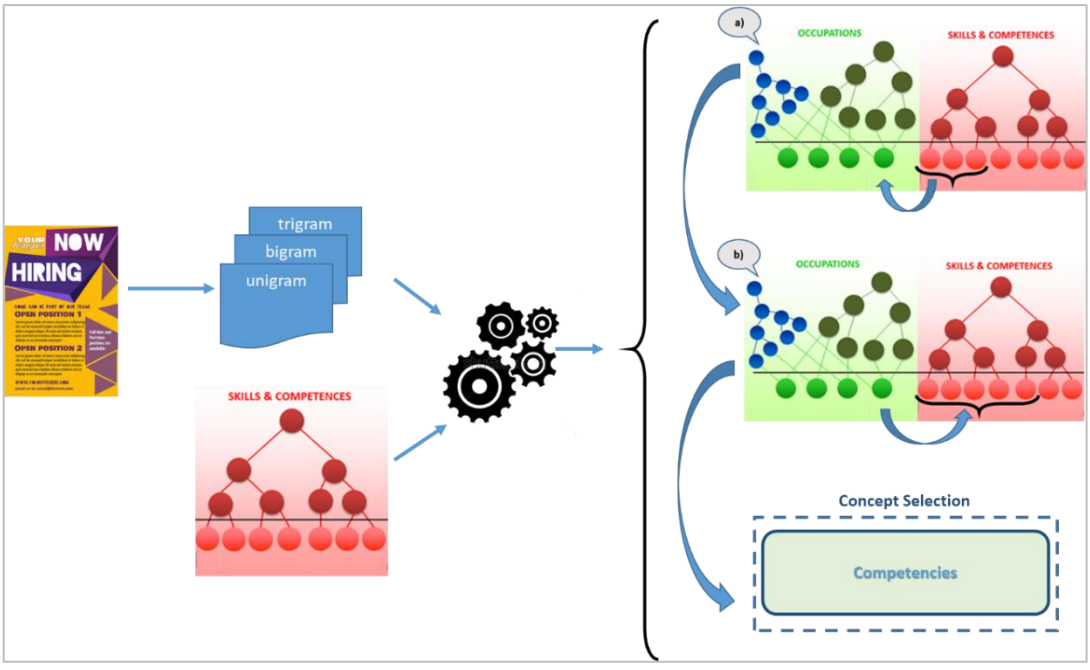
31. ábra: Hirdetések távolsága a tf-idf mátrixon futtatott t-SNE algoritmus alapján

[53] Előfeldolgozás alatt itt 6.2 alfejezetben említett lépéseket értem, melynek során azokat a szakaszokat igyekeztem beazonosítani, ahol jellemzően az elvárások megjelennek, hogy kiszűrjem a zajnak vehető (például a hirdető cégre, annak értékvállalásaira stb. vonatkozó) adatokat. Ennek implementációja megtalálható a dolgozathoz tartozó GitHub repozitóriumban (lásd. 45. lábjegyzet)

A 31. ábra az egyes hirdetéseknek a *tf-idf* mátrix alapján, a *t-SNE* algoritmus segítségével kalkulált távolságát mutatja. A színek az egyes foglalkozások szerint különböztetik meg az egyes pontokat. Az ábrán jól látszik, hogy a hirdetések tartalmuk alapján egyáltalán nem különülnek el jól körülhatárolható csoportokba, bár néhány helyen felfedezhetőek tömörülések, ami azt jelzi, hogy bizonyos hirdetések közelebb állnak egymáshoz tartalom szempontjából. Csepregi (2020) és a saját eredményeim alapján úgy döntöttem, hogy az álláshirdetések – leírásuk alapján történő – klaszterezésére irányuló törekvéseimmel a kutatás jelen szakaszában felhagyok, mivel egyetértek a szerző előbb idézett megállapításával, miszerint az álláshirdetések témája a vizsgált megközelítéssel nem, vagy csak nagyon korlátozottan, kis hatásfokkal beazonosítható.

7.3 Látens igények feltárása az explicit megjelenő kompetenciák kapcsolatai alapján

Amennyiben semmilyen, a foglalkozásra utaló kifejezés nem azonosítható be algoritmikusan az álláshirdetés címében vagy leírásában, azt is meg lehet vizsgálni, hogy a feltárt kompetenciahalmazokon keresztül, indirekt módon be lehet-e egyértelműen azonosítani egy kapcsolódó foglalkozást az ontológiában (32. ábra a.). Ha ez a lépés eredménnyel jár, úgy már ismét a foglalkozás felől, meg kell vizsgálni, hogy létezik-e olyan egyéb kapcsolódó kompetencia, melyet az álláshirdetés nem tartalmazott (32. ábra b.).



32. ábra: Látens kompetenciák beazonosítása az ESCO segítségével (saját szerkesztés Boomgaert, 2013 alapján)

Egy foglalkozás akkor azonosítható be ilyen módon, ha tudunk találni azt leginkább jellemző megkülönböztető kompetencia- illetve tudáselemeket vagy halmazokat. Ebben az esetben a megkülönböztető kompetencia az, amely egy adott foglalkozás, vagy azok egy jól meghatározható körének esetében (például fejlesztő) specifikus, míg egy általános kompetencia több foglalkozás ellátásához is szükséges. Ilyen megkülönböztető kompetenciákat jelölhetnek például egyes programozási vagy adatbázis ismeretek, specifikus könyvelési vagy kontrolling tudáselemek stb.

Amennyiben a szükséges kompetenciahalmaz állásajánlatokban való beazonosításához egy sztenderd nomenklatúrát vagy specifikus ontológiát használtunk, úgy a hirdetésekben feltárt kompetenciakifejezések és egyes absztraktabb ontológiaelemek közötti kapcsolat is meghatározható. Fogalmazhatunk úgy, hogy ezekben az esetekben az absztrakt kompetencia iránti igény szintén implicit jelenik meg. Az így a pozícióhoz kapcsolható magas szintű, általános, szélesebb tudásterületet reprezentáló ontológiaelemek a kínálati oldalon tanulási eredményeket (*learning outcome*) jeleníthetnek meg, így azok feltárása szintén hozzáadott értéket jelenthet a lehetséges felhasználóink számára. Vegyük a következő példát:

- Ha az állásajánlat szövegében szereplő szó/kifejezés egy a következők közül: “Java 8”, “JDBC”, “Hibernate”,
- akkor a kapcsolódó absztrakt kompetenciák a következők: “objektum orientált programozás”, “relációs adatmodell”, “objektum-relációs leképezés”.

Ehhez természetesen szükség van arra, hogy az elemek hierarchiája rendelkezésre álljon, azaz meg tudjuk mondani egy szemantikus adatforrás, például egy ontológia alapján, hogy egy adott absztrakt kompetenciának milyen instanciái lehetnek. Erre a célra a STUDIO-t (4.1.2 alfejezet), és az ESCO-t (4.1.1 alfejezet) is alkalmasnak látom, melyek esetében a kapcsolódó tudáselemek beazonosítása történhet az ontológia metamodellje alapján, a kapcsolatok szemantikus tulajdonságait felhasználva, vagy gráfelméleti alapon is. Ha például szemantikai alapon vizsgálódunk, a STUDIO ontológia egy tudásterülete esetében az „ISMERETÉT KÖVETELI” reláció például egyértelműen rámutat azokra a tudáselemekre, melyek ismerete addicionálisan szükséges – a STUDIO terminológiájában – „a kérdéses tudásterület ismeretének elfogadásához”, azaz általánosan fogalmazva egyfajta hierarchikus kapcsolatot jelez.

Mesterképzési szakdolgozatomban (Neusch, 2014) szemantikai és gráfelméleti megközelítést használva szabom le a szakterületi ontológiát adott képzési igényeknek megfelelően, és építem fel így, az adott területek tudásának tesztelését lehetővé tevő fogalomköröket. A szakdolgozatban a tudáselemek listája képzési igényeket testesít meg, de az ott leírt modell pozíciókra – a kapcsolódó álláshirdetésekben beazonosított kompetenciák segítségével – szintén használható.

Az algoritmus egy tudáselemlistából kiindulva kezdi bejárni az ontológiát a hierarchikus relációk mentén, az adott részontológia kezdőpontját reprezentáló belépési pont irányába. Ez a STUDIO ontológia esetében azt jelenti, hogy a specifikustól az általánosabb, absztraktabb tudáselem felé haladunk. Ha egy adott, előre meghatározott k távolságon belül az algoritmus talál egy olyan, másik tudáselemet, mely szintén szerepelt az input listában, akkor azt feltételezi, hogy a két elem relatív közelsége miatt, az azokat összekötő úton található többi tudáselem is releváns a képzési igény (itt pozíció) szempontjából, akkor is, ha explicit módon nem jelentek meg az elvárt elemek listájában, azaz az álláshirdetésben.

Az algoritmus az így beazonosított elemeket szabja le Fogalomkör objektumokba, úgy, hogy megőrzi az ontológia által nyújtott, a kapcsolatokban megtestesülő szemantikus információkat is. Az ilyen módon leszabott Fogalomkör objektumok pedig jelen tézisben felvázolt kontextusban megfeleltethetők az adott pozíció által igényelt kompetenciahalmaznak. A kutatás későbbi szakaszaiban tovább kívánom vizsgálni, ezen látens kompetenciák beazonosításának lehetőségeit is, illetve szeretném a már felépített modellt kiegészíteni a dolgozatban előzőleg ismertetett kulcsszótávolsági metrikák beépítésével.

Szabadszöveges, félig- vagy rosszul strukturált dokumentumok elemzésével is találhatunk az adott probléma kontextusában releváns tudást reprezentáló kifejezéseket. Ilyen dokumentumok lehetnek például a folyamatmodellek esetében a különböző vállalati szabályzatok és egyéb belső kiadványok, a szakterülethez kapcsolódó szakkönyvek, szakfolyóiratok vagy akár hiperszöveges dokumentumok stb. is. Bár annak vizsgálata, hogy hogyan lehet a látens kompetenciákat a szabadszöveges dokumentumokban feltárni, és ezzel gazdagítani az adott pozícióról tárolt információhalmazt, további irány lehet a kutatás későbbi szakaszaiban, azonban jelen dolgozatban bővebben nem foglalkozom a kérdéssel.

Az álláshirdetésben explicit módon nem említett, látens kompetenciák feltárását a pozícióelnevezések alapján beazonosítható foglalkozásokon keresztül terveztem megvalósítani. Ugyanis az ESCO ontológiában már rendelkezésre áll a foglalkozás-kompetencialista összekapcsolás. Így amennyiben az álláshirdetés kevés információt tartalmaz, azt ezen a kapcsolaton keresztül a hirdetésben nem megjelenő látens elemekkel ki lehet egészíteni. Ugyanez igaz a Studio ontológia esetére, ahol az előbb említett gráfelméleti eljárásokkal még újabb elemek tárhatóak fel az ontológián belül. Mind a két eljárással ki lehet egészíteni az egyes álláshirdetésekhez, azaz a pozíciókhoz tartozó kompetencialistákat olyan új elemekkel, amelyek nem voltak konkrétan megemlítve a hirdetésekben, de relevánsak lehetnek. Ezáltal sokkal pontosabb képet kapnak a döntéshozók az elvárt igényekről. A pozíciók elnevezésén keresztül a foglalkozások beazonosítására tett kísérleteimet még a jövőben finomítani kell, hogy alkalmas módszer váljon belőlük. Azonban az itt bemutatott modell pontossága nem marad el az idézett Amato és szerzőtársai által publikáltaktól (lásd. 7.1.1 fejezet), sőt, ha csak csekély mértékben is, de jobban teljesít. A jövőben folytatni kívánom a modell pontosításával (például más eljárásokkal való kísérletezés réven), és amennyiben ezt megvalósítom, már mind az ESCO ontológián, mind – a mestertézisben vázolt módon – a Stúdió ontológián keresztül tudok látens kompetenciákat rendelni az álláshirdetésekhez.

7.4 Összefoglalás és lehetséges jövőbeli irányok

Jelen fejezetben a disszertáció 3. kutatási kérdéséhez kapcsolódóan a hirdetésekben explicit nem megjelenő, implicit vagy látens kompetenciák beazonosításának lehetőségeit vizsgáltam. Koncepcionálisan három nagyobb irányt vázoltam fel.

Az első, a hirdetésekhez kapcsolódó foglalkozás beazonosításán keresztül, az annak kontextusában – a felhasznált külső ontológiák alapján – releváns kompetenciák elfogadása, mint amik implicit szükségesek az adott pozíció megfelelő ellátásához. Annak vizsgálatára, hogy ez az irány mennyire járható, kísérleteket végeztem a foglalkozásoknak a hirdetések címében való beazonosíthatóságát vizsgálva. Egy reguláris kifejezések használatán, illetve egyszerű döntési szabályok alkalmazásán alapuló módszerrel, a vizsgált hirdetéshalmaz közel negyedét tudtam egy-egy – valamely felhasznált külső ontológiában is megjelenő – foglalkozáshoz kapcsolni. Ezek után a maradék, az előző módszerrel nem beazonosítható hirdetések egy részének címeit vetettem össze – lexikográfiai és kváziszemantikai módszerekkel – az ESCO és

az O*NET ontológiákban található foglalkozás-megnevezésekkel. Az erre épített döntési fát használó gépi tanulási modell közepes eredményeket mutat. A modellt a jövőben nagyobb elemszámú halmazzal tanítva, illetve más módszereket is felhasználva, a kísérletet megismételni kívánom, hiszen az így beazonosított foglalkozások alapján az ontológiákban hozzájuk kapcsolódó kompetenciák is elfogadhatóak, mint amik a hirdetés szempontjából implicit relevánsak. A harmadik kutatási kérdés szempontjából ezt az irányt elfogadom, mint egy lehetséges megoldást, azaz az implicit kompetenciák a hirdetésekben kijelölt foglalkozások beazonosításának segítségével feltárhatóak, azonban ennek a beazonosítási folyamatnak a hatékonysága javításra szorul.

A második nagy, vizsgált irányt azonban, hogy a hirdetések tartalma alapján azok csoportokba sorolhatóak, mely csoportok alapján következtetni lehet az átfedő kompetenciatartalomra, vagy valamiféle foglalkozási csoportra, elvetettem. Ezt a döntést Csepregi (2020) és saját *t-SNE* módszerrel végzett kísérletem alapján hoztam meg.

A fejezet utolsó részében bemutattam a 3. lehetséges irányt a látens kompetenciák feltárására, az explicit megjelenő kompetenciák ontológiakapcsolatainak segítségével. Ezt a témát jelen disszertációban részletesen nem tárgyaltam, azonban kitértem a mester tézisemben megvalósítottak ismertetésére. Az ott elért eredményeim alapján ezt az irányt szintén el tudom fogadni, mint ami alkalmas a látens kompetenciák beazonosítására, ahogy azt a harmadik kutatási kérdésben felvetettem, és a kutatás jövőbeli szakaszaiban ezt a területet is fejleszteni kívánom.

8 Összefoglalás és további kutatási lépések

A munkaerőpiacon igényelt tudás, készségek és képességek rendkívüli ütemben változnak, összhangban az üzleti problémák megoldásához szükséges kompetenciaigényekkel. Az ismeretek a szervezetekben funkcionális szerepet töltenek be, azaz az egyes tevékenységek elvégzéséhez kapcsolódnak. Ebből kifolyólag előfordul, hogy egyes kompetenciák iránti igény lecsökken, amikor az adott feladat elvégzésére már nincs szükség, vagy az emberi erőforrást kiváltják a folyamatban. Ez egyre gyakrabban, többek között az új technológiák által a munkaerőpiacra kifejtett bomlasztó (*disruptive*) hatásának köszönhetően, meg is történik. Ezzel együtt persze az új technológiák okozta változások rengeteg új üzleti lehetőséget is generálnak, és így számos új pozíció is megjelenik a piacon, melyek ellátásához teljesen új, vagy átalakult kompetenciakészlettel kell rendelkeznie a munkavállalóknak. A kompetenciák iránti igény regionálisan is igen változó. A vállalatok költségcsökkentési célból egyes munkaköröket, vagy akár komplett üzletágakat is kiszerveznek, áttelepítenek akár egy másik kontinensre is, szinte egyik napról a másikra.

Ilyen gyors ütemű, nagy ívű változások közepette különösen szükség van arra, hogy a képzőintézmények olyan kompetenciákkal vértezzék fel a hallgatókat, amelyek segítségével karrierjük kezdetén el tudnak helyezkedni, és egy stabil munkaerőpiaci pozíciót tudnak kiépíteni maguknak, melyre építve folyamatosan meg tudnak újulni, hogy lépést tartsanak az egyre gyorsuló igényekkel. Ehhez persze arra van szükség, hogy a tantervek is kövessék a legfrissebb munkaerőpiaci trendeket és tükrözzék a változásokat, sőt proaktívan megpróbáljanak elébe is menni azoknak. Ezt a munkát természetesen olyan információkkal kell megalapozni, amelyek nem csak egy adott időpillanatban írják le a munkaerőpiaci kompetenciakeresletet, hanem annak időbeli változását is képesek megmutatni.

Jelen tézisben ennek a törekvésnek a támogatásához dolgoztam ki egy koncepciót. Az első kutatási kérdésemmel összhangban egy olyan keretrendszer megvalósíthatóságának egyes aspektusait vizsgáltam, melyet munkaerőpiaci adattárháznak nevezek. A kutatási kérdés vizsgálata során azt vettem górcső alá, hogy miként lenne célszerű ezt a rendszert feltölteni internetes állásportálokról gyűjtött hirdetésekkel, illetve azok feldolgozása után milyen struktúrában érdemes tárolni ezeket a hirdetéseket és az azokat jellemző dimenziókat, melyek közül a legfontosabbak a kutatás szempontjából a kapcsolódó kompetenciák. A dolgozat 5.

fejezetében, mely az adattárház megvalósíthatóságát tárgyalja, kitértem az adatok gyűjtésének témakörére, bemutattam azt a *scraper* alkalmazást, amit erre a feladatra implementáltam, és ami 2019 elejétől gyűjtötte az adatokat, amiken későbbi elemzéseimet is végrehajtottam. Bemutattam az álláshirdetések leírásaiból kinyerni és tárolni kívánt információk listáját, és azt a struktúrát, amiben tárolásukat megvalósíthatónak látom. Megvizsgáltam és összehasonlítottam továbbá egy komplex szempontrendszer alapján egyes adattárolási elveket és megoldásokat is. Ez alapján az összehasonlítás alapján pedig javaslatot tettem a problémának legmegfelelőbb adattárolási platform kiválasztására, illetve bemutattam a megvalósításra javasolt hibrid architektúrát, és a mellette szóló érveket.

A munkaerőpiaci adattárház legfontosabb céljaként fogalmaztam meg, hogy a hirdetések kompetenciatartalmáról, annak időbeli- és térbeli alakulásáról hasznos információkat szolgáltasson a döntéshozók számára. Azonban ezek a kompetenciák strukturált formában nem állnak rendelkezésre az álláshirdetésekben, hanem azok leírásából kell ezt az információt a természetesnyelv-feldolgozás és a gépi tanulás témaköréhez kapcsolódó technikákkal kinyerni. Ennek megfelelően, a második kutatási kérdéssel összhangban a dolgozat 6. fejezetében olyan módszereket mutattam be, melyek alkalmasak lehetnek arra, hogy a strukturálatlan szövegben „minőségi kifejezéseket”, ebben az esetben kompetenciajelölteket azonosítsunk be segítségükkel.

- A fejezet elején statisztikai alapon meghatároztam – a későbbi feldolgozás során használt – szóennesek hosszát.
- Bemutattam azt, hogy miért nem, vagy csak nagyon korlátozottan alkalmasak a kifejezés- és dokumentumgyakoriságra (*tf, df*), illetve a *tf-idf* értékre épülő modellek a kompetenciajelöltek beazonosítására.
- Bemutattam továbbá egy külső ontológiák tartalma alapján felépített kompetenciaszótár használatában rejlő lehetőségeket az információt hordozó kifejezések feltárására. Tovább részleteztem ennek a módszernek a határait is.
- A fejezet utolsó blokkjában szakirodalmi áttekintés alapján ismertettem olyan metrikákat, melyek kifejezések hasonlóságának, illetve távolságának meghatározására használhatóak. Ezek után számszerűsítettem az álláshirdetések leírásaiból képzett szóennesek és a kompetenciaszótár elemei közötti hasonlóságot.

- Az így kapott eredményeket magyarázó változóként használva egy logisztikus regressziós modellben, arra kerestem a választ, hogy megadható-e a segítségükkel egy olyan egyenlet, amivel számszerűsíthető, hogy egy kifejezést elfogadhatunk-e kompetenciajelöltként. A modell felépítésére az *IBM SPSS Statistics* programcsomagot használtam. Mivel a modellparaméterek tesztelését a tanító halmazon végeztem el, és nem állt fent a „kukucskálás" (*data peeking*) veszélye, ezért nem képeztem az adatokból külön validációra használt csoportot. Változószelekcióra a *Backward Wald* eljárást használtam.
- A felépített modell a tesztadatokon elfogadhatóan teljesített; a felidézési arány 85%, míg a precizitás 71,9%. Mivel a folyamatba való manuális beavatkozás kezdetben elkerülhetetlen, azaz mielőtt elfogadhatnánk ezeket a kompetenciajelölteket valós kompetenciaként, egy szakértőnek át kell néznie az eredményeket, így a modell elfogadható, mint ami hasznos információkkal tud szolgálni és hozzáadott értékkel bír.

A fejezet összefoglalásában kiemeltem azokat az irányokat, melyekben a jövőben a kutatást folytatni szeretném. Ezek közül a legfontosabb a hirdetések szövegének előfeldolgozása által a zaj csökkentése a modell pontosságának növelése érdekében. Ebben az irányban azóta történtek is fejlesztések, amikre alapozva megvizsgáltam a 7.2 alfejezetben – a *t-SNE* módszer használatával –, hogy lehet-e arra következtetni, hogy a hirdetések témájuk alapján jól elkülönülő klaszterekbe rendeződnek. A másik fontos irány, amerre a kutatással a jövőben haladni szeretnék, az a modell adaptálása magyar nyelvre is, hogy a keretrendszer hosszabb távon alkalmazható legyen hazai kompetenciakereslet elemzésére, annak területi összehasonlítására stb. is.

A kutatás harmadik nagy blokkjában (7. fejezet) megvizsgáltam, hogy az álláshirdetésekben közvetlenül nem megjelenő, látens kompetenciák beazonosítására milyen módszereket és alkalmazásokat találok alkalmasnak. A legfontosabb irány, amit részletesen elemeztem, az ESCO és az O*NET ontológiák tartalma segítségével az állásajánlatokban meghirdetett pozícióhoz kapcsolódó foglalkozás beazonosíthatósága.

- Ennek érdekében kidolgoztam egy reguláris kifejezésekre és egyszerű szabályokra alapuló módszert, aminek segítségével a 2019 októberi

álláshirdetések 23,7%-át tudtam foglalkozáshoz kapcsolni, 97,5%-ban helyesen.

- A fenti cél elérése érdekében részletesen ismertettem továbbá egy döntési fára alapuló osztályozó modellt. Ehhez a feladathoz a döntési kritériumokat a hirdetések – előkészítő lépések során – megtisztított címének, és a felhasznált ontológiákban rögzített foglalkozás-megnevezéseknek a lexikográfiai- és „kvázi-szemantikai" hasonlóságértékei szolgáltatták. Egy ilyen modell segítségével a későbbiekben, az ontológiákban a foglalkozásokhoz kapcsolódó, de a hirdetésekben explicit nem megjelentő, látens kompetenciák szintén eltárolhatóak az adattárházban. A legjobbként elfogadott modell teszthalmazon mért precizitása 58%, míg felidézési aránya 68%, ami bár nem kiemelkedő, de felveszi a versenyt az irodalomban fellelhető, hasonló feladatra megalkotott modellekkel.
- A *t-SNE* módszer segítségével igazoltam Csepregi (2020) eredményeit, miszerint álláshirdetések klaszterei (témacsoportok) nem, vagy csak nagyon korlátozottan, kis hatásfokkal beazonosíthatóak a leírásukból készült *tf-idf* mátrix alapján.

A 7. fejezet lezárásaként bemutattam olyan módszereket, melyek segítségével a látens igények, az explicit megjelenő kompetenciák kapcsolatain keresztül tárhatóak fel. Ezeket az irányokat az értekezésben részletesen nem vizsgáltam, azonban további kutatásaim során mindenképpen érdemesnek tartom őket górcső alá venni.

A kutatásnak a dolgozatban részletesen nem vizsgált területei közül mindenképpen meg kívánom vizsgálni a közeljövőben, hogy milyen riportokat és elemzéseket lehet egy, a felvázolt módon megvalósított munkaerőpiaci „adattárház" tartalmából kinyerni, melyek kielégíthetik a tárgyfelelősök információigényeit. Elsősorban a trendekre koncentrálva fogom megvizsgálni a kompetenciák alakulását, fókuszban azzal, hogy láthatóak-e a bevezetőben részletezett változásokra utaló jelek.

Amennyiben a szükséges erőforrások biztosíthatóak és a rendszert ténylegesen implementálni tudom, úgy a kutatás végcélja egy olyan elemző eszköz nyújtása a szak- és tárgyfelelősöknek, aminek a segítségével a tantervek objektív és validálható módon tarthatók összhangban a munkaerőpiaci kereslet változásával.

9 Irodalomjegyzék

Acemoglu, D. and Autor, D. (2010), *Skills, Tasks and Technologies: Implications for Employment and Earnings*, Working Paper No. 16082, National Bureau of Economic Research, available at: https://doi.org/10.3386/w16082.

Adam, S. (2004), "Using Learning Outcomes: A consideration of the nature, role, application and implications for European education of employing learning outcomes at the local, national and international levels.", presented at the United Kingdom Bologna Seminar, Herriot-Watt University, Edinburgh, July, available at: http://www.aic.lv/ace/ace_disk/Bologna/Bol_semin/Edinburgh/S_Adam_Bacgrerep_presentation.pdf (accessed 6 August 2019).

Ahmed, F., Capretz, L.F. and Campbell, P. (2012), "Evaluating the Demand for Soft Skills in Software Development", *IT Professional*, Vol. 14 No. 1, pp. 44–49, available at: https://doi.org/10.1109/MITP.2012.7.

Amato, F., Boselli, R., Cesarini, M., Mercorio, F., Mezzanzanica, M., Moscato, V., Persia, F., *et al.* (2015), "Classification of Web Job Advertisements: A Case Study", *23rd Italian Symposium on Advanced Database Systems (SEBD 2015)*, presented at the 23rd Italian Symposium on Advanced Database Systems (SEBD 2015) (Gaeta, 14/06/2015 - 17/06/2015), Curran, Gaeta, Italy, pp. 144–151, available at: http://hdl.handle.net/10863/10468.

Autor, D.H., Levy, F. and Murnane, R.J. (2003), "The Skill Content of Recent Technological Change: An Empirical Exploration", *The Quarterly Journal of Economics*, Vol. 118 No. 4, pp. 1279–1333, available at: https://doi.org/10.1162/003355303322552801.

Bard, G.V. (2007), “Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric”, *Proceedings of the Fifth Australasian Symposium on ACSW Frontiers - Volume 68*, Australian Computer Society, Inc., AUS, pp. 117–124, available at: (accessed 27 October 2020).

Bártfai, N. (2010), *Mobiltelefonos játékok tervezése és fejlesztése*, PhD Dissertation and Thesis, University of Debrecen, Debrecen, Hungary, available at: https://mad-hatter.it.unideb.hu/portal/displayDocument/Szervezeti%20t%C3%A1rak/Kari%20t%C3%A1rak/IK/Dokumentumt%C3%A1r/Doktori%20v%C3%A9d%C3%A9sek/Mobiltelefonos%20j%C3%A1t%C3%A9kok%20tervez%C3%A9se%20%C3%A9s%20fejleszt%C3%A9se%20%C3%A9rtekez%C3%A9s.pdf (accessed 28 October 2020).

Beck, M. and Libert, B. (2017), “The Rise of AI Makes Emotional Intelligence More Important”, *Harvard Business Review*, 15 February, available at: https://hbr.org/2017/02/the-rise-of-ai-makes-emotional-intelligence-more-important (accessed 4 May 2019).

Bennett, C.H., Gacs, P., Ming Li, Vitanyi, P.M.B. and Zurek, W.H. (1998), “Information distance”, *IEEE Transactions on Information Theory*, presented at the IEEE Transactions on Information Theory, Vol. 44 No. 4, pp. 1407–1423, available at: https://doi.org/10.1109/18.681318.

Bíró, T. (1997), “Fizika és Nyelvészet”, available at: http://www.birot.hu/publications/TTKsNyuz-nyelvisarok-fizikaesnyelveszet.htm (accessed 1 November 2020).

Bíró, T. (1998), “Some Statistical Games with Written Texts”, *DOXIMP 3: Graduate Students’ Third Linguistics Symposium, June 5, 1998, Budapest, Selected Papers*, Vol. 6, Theoretical Linguistics Program, Eötvös Loránd University, Research Institute for Linguistics, Hungarian Academy of Sciences, pp. 1–10.

Bodon, F. (2010), “Adatbányászati algoritmusok”, Dr. Bodon Ferenc, available at: http://www.cs.bme.hu/~bodon/magyar/adatbanyaszat/tanulmany/adatbanyaszat.pdf (accessed 1 October 2019).

Bodon, F. and Buza, K. (2013), “Adatbányászat”, Elektronikus tananyag, available at: http://www.cs.bme.hu/~buza/pdfs/adatbanyaszat-cover.pdf (accessed 20 December 2020).

Boomgaert, W. (2013), “ESCO - A multilingual classification of European Skills, Competences, Qualifications and Occupations”, presented at the National Workshop on ISCA, Sofia, 24 September, available at: https://www.bia-bg.com/uploads/files/News/ESCO_Presentation_Sofia_24September2013.pdf (accessed 8 September 2019).

Brewer, E. (2012), “CAP twelve years later: How the ‘rules’ have changed”, *Computer*, presented at the Computer, Vol. 45 No. 2, pp. 23–29, available at: https://doi.org/10.1109/MC.2012.37.

Brown, P.F., deSouza, P.V., Mercer, R.L., Pietra, V.J.D. and Lai, J.C. (1992), “Class-based N-gram Models of Natural Language”, *Comput. Linguist.*, Vol. 18 No. 4, pp. 467–479, available at: http://dl.acm.org/citation.cfm?id=176313.176316 (accessed 16 March 2017).

Budapesti Corvinus Egyetem. (2018), “GAZDASÁGINFORMATIKUS (MSc) MESTERKÉPZÉSI SZAK”, available at: http://gazdalkodastudomany.uni-

corvinus.hu/fileadmin/user_upload/hu/gazdalkodastudomanyi_kar/files/Hallgatoi_info/szakleirasok_mester/MGINF.pdf (accessed 10 June 2019).

Budapesti Corvinus Egyetem. (n.d.). “Tantárgyi adatlap: Software Engineering”, available at: http://portal.uni-corvinus.hu/index.php?id=22720&tanKod=2SZ31NBK02M (accessed 10 June 2019).

Burleson, C. (2016), *Introduction to Linked Data and the Semantic Web*, Technics Publications, available at: https://www.oreilly.com/library/view/introduction-to-linked/9781634622141/ (accessed 20 June 2019).

Caballero, J.G., Szabó, I.B., Castello, V. and Vettraino, L. (2014), “DEVELOPING AND ALIGNING COMPETENCES IN THE TOURISM INDUSTRY. THE SMART PROJECT EXPERIENCE”, *ICERI2014 Proceedings*, pp. 1013–1022, available at: https://library.iated.org/view/GUERREROCABALLERO2014DEV (accessed 14 September 2019).

Castello, V., Mahajan, L., Flores, E., Gabor, M., Neusch, G., Szabo, I., Guerrero, J., *et al.* (2014), “THE SKILL MATCH CHALLENGE. EVIDENCES FROM THE SMART PROJECT”, in Gómez Chova, L., López Martínez, A. and Candel Torres, I. (Eds.), *ICERI2014 Proceedings*, IATED Academy.

Ceri, S., Bozzon, A., Brambilla, M., Valle, E.D., Fraternali, P. and Quarteroni, S. (2013), *Web Information Retrieval*, Springer-Verlag, Berlin Heidelberg.

Chang, H.-C., Wang, C.-Y. and Hawamdeh, S. (2018), “Emerging trends in data analytics and knowledge management job market: extending KSA framework”, *Journal of Knowledge Management*, Vol. 23 No. 4, pp. 664–686, available at: https://doi.org/10.1108/JKM-02-2018-0088.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002), "SMOTE: synthetic minority over-sampling technique", *Journal of Artificial Intelligence Research*, Vol. 16 No. 1, pp. 321–357, available at: https://dl.acm.org/doi/10.5555/1622407.1622416.

Choi, S., Cha, S. and Tappert, C.C. (2010), "A Survey of Binary Similarity and Distance Measures", *Journal of Systemics, Cybernetics and Informatics*, Vol. 8 No. 1, pp. 43–48.

Chong, T.Y., Banchs, R.E., Chng, E. and Li, H. (2013), "Modeling of term-distance and term-occurrence information for improving n-gram language model performance.", *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, Sofia, Bulgaria, pp. 233–237.

Cilibrasi, R. and Vitanyi, P.M.B. (2005), "Clustering by compression", *IEEE Transactions on Information Theory*, presented at the IEEE Transactions on Information Theory, Vol. 51 No. 4, pp. 1523–1545, available at: https://doi.org/10.1109/TIT.2005.844059.

Cocchiarella, N. (1991), "Formal Ontology", in Burkhardt, H. and Smith, B. (Eds.), *Handbook of Metaphysics and Ontology*, Philosophia Verlag, pp. 640–647.

Codd, E.F., Codd, S.B. and Salley, C.T. (1993), "Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate", E. F. Codd and Associates, available at: http://www.estgv.ipv.pt/PaginasPessoais/jloureiro/ESI_AID2007_2008/fichas/codd.pdf (accessed 2 January 2021).

Corcho, O., Fernández-López, M. and Gómez-Pérez, A. (2003), "Methodologies, Tools and Languages for Building Ontologies: Where is Their Meeting Point?",

Data & Knowledge Engineering, Vol. 46 No. 1, pp. 41–64, available at: https://doi.org/10.1016/S0169-023X(02)00195-7.

Corcho, O. and Gómez-Pérez, A. (2000), "Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages", *Proceedings of the ECAI 2000 Workshop on Application of Ontologies and Problem-Solving Methods*, Berlin.

Csepregi D. (2020), *Első Lépés az Automatizált Oktatásfejlesztés Felé: Hogyan tehet szert előnyre szövegelemzési eszközökkel a Corvinus Gazdaságinformatikus képzése?*, TDK dolgozat, Budapesti Corvinus Egyetem, Gazdálkodástudományi kar, Budapest, available at: http://publikaciok.lib.uni-corvinus.hu/publikus/tdk/csepregi_d_2020a.pdf (accessed 27 December 2020).

Cummins, R. and O'riordan, C. (2005), "Evolving General Term-Weighting Schemes for Information Retrieval: Tests on Larger Collections", *Artificial Intelligence Review*, Vol. 24 No. 3, pp. 277–299, available at: https://doi.org/10.1007/s10462-005-9001-y.

Damerau, F.J. (1964), "A technique for computer detection and correction of spelling errors", *Communications of the ACM*, Vol. 7 No. 3, pp. 171–176, available at: https://doi.org/10.1145/363958.363994.

Derényi, A., Loboda, Z., Szebeni, K., Szent-Léleky, G. and Szlamka, E. (2015), "Referencing and Self-certification Report of the Hungarian Qualifications Framework to the EQF and to the QF-EHEA", EQF National Coordination Point, Educational Authority, available at: https://www.oktatas.hu/pub_bin/dload/kepesitesek/referencing_report_HuQF_EQF.pdf (accessed 25 July 2019).

Derényi, A. and Vámos, Á. (2015), *A Felsőoktatás Képzési Területeinek Kimeneti Leírása – Ajánlások*, Oktatási Hivatal, available at: https://www.oktatas.hu/pub_bin/dload/unios_projektek/tamop413/eredmenyek/kimeneti_leirasok.pdf (accessed 10 June 2019).

Deshpande, A. and Kumar, M. (2018), *Artificial Intelligence for Big Data: Complete Guide to Automating Big Data Solutions Using Artificial Intelligence Techniques*, Packt Publishing Ltd.

Dreßler, K. and Ngonga Ngomo, A.-C. (2017), "On the efficient execution of bounded Jaro-Winkler distances", *Semantic Web*, IOS Press, Vol. 8 No. 2, pp. 185–196, available at: https://doi.org/10.3233/SW-150209.

DuCharme, B. (2013), *Learning SPARQL*, 2nd ed., O'Reilly Media, Inc., Online, available at: https://www.oreilly.com/library/view/learning-sparql-2nd/9781449371449/ (accessed 30 June 2019).

European Commission. (2018), "ESCO - Skills/competences - European Commission", available at: http://data.europa.eu/esco/skill/8259c2c0-2749-41e2-a4a8-4a64a76eab37 (accessed 8 August 2019).

EuroVoc. (n.d.). "Equivalence relationship | EuroVoc", available at: http://eurovoc.europa.eu/drupal/?q=node/322 (accessed 19 April 2017).

Evangelopoulos, N. and Visinescu, L. (2012), "Text-Mining the Voice of the People.", *Communications of the ACM*, Vol. 55 No. 2, pp. 62–69, available at: http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=71681310&site=eds-live.

Fajszi, B. and Cser, L. (2004), *Üzleti Tudás Az Adatok Mélyén*, Budapesti Műszaki és Gazdaságtudományi Egyetem.

Fajszi, B., Cser, L. and Fehér, T. (2010), *Üzleti Haszon Az Adatok Mélyén*, Alinea Kiadó - IQSYS Informatikai és Tanácsadó Zrt.

Falus, I. (2010), "Javaslat az OKKR szintjeire és szintleírásaira (Vitaindító)", Nemzeti Fejlesztési Ügynökség, available at: http://ofi.hu/sites/default/files/ofipast/2010/01/javaslat_falus_1_21_v3.pdf (accessed 7 August 2019).

Fazekas, K. (2017), *Nem Kognitív Készségek Kereslete És Kínálata a Munkaerőpiacon*, Institute of Economics, Centre for Economic and Regional Studies, Hungarian Academy of Sciences, Budapest, available at: http://www.mtakti.hu/wp-content/uploads/2017/11/FK-BWP1709-jav-OE_FKjav.pdf (accessed 20 July 2019).

Field, J. (2001), "Lifelong education", *International Journal of Lifelong Education*, Vol. 20, pp. 3–15, available at: https://doi.org/10.1080/09638280010008291.

Fliszár, V., Kovács, E., Szepesváry, L. and Szüle, B. (2016), *Többváltozós Adatelemzési Számítások*, Budapesti Corvinus Egyetem, available at: http://unipub.lib.uni-corvinus.hu/2438/.

Fowler, M. (2003), *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd ed., Addison-Wesley Professional, Boston.

Fox, A. and Brewer, E.A. (1999), "Harvest, yield, and scalable tolerant systems", *Proceedings of the Seventh Workshop on Hot Topics in Operating Systems*, presented at the Proceedings of the Seventh Workshop on Hot Topics in Operating Systems, pp. 174–178, available at: https://doi.org/10.1109/HOTOS.1999.798396.

Fürnkranz, J. (1998), “A Study Using n-gram Features for Text Categorization”, *Technical Report OEFAI-TR-98-30*, Austrian Research Institute for Artificial Intelligence.

Gábor, A. (2019), “Képzési Kimeneti Követlemények Magyarországon”, 6 August.

Gábor, A., Kő, A., Szabó, Z. and Fehér, P. (2016), “Corporate Knowledge Discovery and Organizational Learning: The Role, Importance, and Application of Semantic Business Process Management—The ProKEX Case”, in Gábor, A. and Kő, A. (Eds.), *Corporate Knowledge Discovery and Organizational Learning*, Springer International Publishing, pp. 1–31, available at: https://doi.org/10.1007/978-3-319-28917-5_1.

Gajdos, S. (2019), *Adatbázisok*, A 2015. évi kiadás negyedik javított utánnyomása., BME, Budapest, Magyarország.

Gilbert, S. and Lynch, N. (2002), “Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services”, *ACM SIGACT News*, Vol. 33 No. 2, pp. 51–59, available at: https://doi.org/10.1145/564585.564601.

Gillani, S. and Kő, A. (2014), “Process-based Knowledge Extraction in a Public Authority: A Text Mining Approach”, in Kő, A. and Francesconi, E. (Eds.), *Lecture Notes in Computer Science*, Vol. 8650, presented at the Electronic Government and the Information Systems Perspective. EGOVIS 2014., Springer, Cham.

Gillani, S.A. (2015), *From Text Mining to Knowledge Mining: An Integrated Framework of Concept Extraction and Categorization for Domain Ontology*, PhD thesis, Corvinus University of Budapest, Budapest, Hungary, available at: http://phd.lib.uni-corvinus.hu/887/ (accessed 14 March 2017).

Gomaa, W.H. and Fahmy, A.A. (2013), “A Survey of Text Similarity Approaches”, *International Journal of Computer Applications*, Foundation of Computer Science (FCS), Vol. 68 No. 13, pp. 13–18, available at: https://doi.org/10.5120/11638-7118.

Gómez-Pérez, A. (1999), “Ontological Engineering [PowerPoint presentation]”, available at: http://icc.mpei.ru/documents/00000823.pdf (accessed 23 January 2014).

Gomez-Perez, A. and Corcho, O. (2002), “Ontology languages for the Semantic Web”, *IEEE Intelligent Systems*, Vol. 17 No. 1, pp. 54–60, available at: https://doi.org/10.1109/5254.988453.

Gottdank, T. (2006), *Szemantikus Web*, Computerbooks.

Gruber, T. (2009), “Ontology”, in Liu, L. and Özsu, M.T. (Eds.), *Encyclopedia of Database Systems*, Springer-Verlag US, available at: http://tomgruber.org/writing/ontology-definition-2007.htm (accessed 16 February 2014).

Gruber, T.R. (1993), “A Translation Approach to Portable Ontology Specifications”, *Knowledge Acquisition*, Vol. 5(2), pp. 199–220, available at: http://tomgruber.org/writing/ontolingua-kaj-1993.htm.

Grundke, R., Jamet, S., Kalamova, M., Keslair, F. and Squicciarini, M. (2017), “Skills and global value chains”, *OECD Science, Technology and Industry Working Papers*, available at: https://doi.org/10.1787/cdb5de9b-en.

Grundke, R., Marcolin, L., Nguyen, T.L.B. and Squicciarini, M. (2018), “Which skills for the digital era?”, *OECD Science, Technology and Industry Working Papers*, available at: https://doi.org/10.1787/9a9479b5-en.

Guarino, N. and Giaretta, P. (1995), “Ontologies and knowledge bases: Towards a terminological clarification”, in Mars, N.J.I. (Ed.), *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, IOS Press, Amsterdam, pp. 25–32.

Guarino, N. and Welty, C. (2000), “Towards a Methodology for Ontology Based Model Engineering”, in Bézivin, J. and Ernst, J. (Eds.), *First International Workshop on Model Engineering*, Nice, France.

Gugnani, A. and Misra, H. (2020), “Implicit Skills Extraction Using Document Embedding and Its Use in Job Recommendation”, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34 No. 08, pp. 13286–13293, available at: https://doi.org/10.1609/aaai.v34i08.7038.

Gulla, J.A. and Brasethvik, T. (2008), “A Hybrid Approach to Ontology Relationship Learning”, *Natural Language and Information Systems*, presented at the International Conference on Application of Natural Language to Information Systems, Springer, Berlin, Heidelberg, pp. 79–90, available at: https://doi.org/10.1007/978-3-540-69858-6_9.

Guo, J. (1997), “Critical Tokenization and Its Properties”, *Computational Linguistics*, Vol. 23 No. 4, pp. 569–596, available at: http://dl.acm.org/citation.cfm?id=972791.972799 (accessed 1 October 2019).

Hall, P.A.V. and Dowling, G.R. (1980), “Approximate String Matching”, *ACM Computing Surveys*, Vol. 12 No. 4, pp. 381–402, available at: https://doi.org/10.1145/356827.356830.

Han, J., Kamber, M. and Pei, J. (2011), *Data Mining: Concepts and Techniques*, 3rd edition., Morgan Kaufmann, Haryana, India; Burlington, MA.

Handel, M.J. (2012), “Trends in Job Skill Demands in OECD Countries”, *OECD Social, Employment and Migration Working Papers*, Vol. 143, available at: https://doi.org/10.1787/5k8zk8pcq6td-en.

Hanushek, E.A., Schwerdt, G., Wiederhold, S. and Woessmann, L. (2015), “Returns to skills around the world: Evidence from PIAAC”, *European Economic Review*, Vol. 73, pp. 103–130, available at: https://doi.org/10.1016/j.euroecorev.2014.10.006.

Harari, Y.N. (2018), *21 Lessons for the 21st Century*, Spiegel & Grau, New York.

Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, 2nd ed., Springer-Verlag, New York, available at: https://doi.org/10.1007/978-0-387-84858-7.

Hecklau, F., Galeitzke, M., Flachs, S. and Kohl, H. (2016), “Holistic Approach for Human Resource Management in Industry 4.0”, *Procedia CIRP*, Vol. 54, pp. 1–6, available at: https://doi.org/10.1016/j.procir.2016.05.102.

Heckman, J.J. and Kautz, T. (2013), *Fostering and Measuring Skills: Interventions That Improve Character and Cognition*, Working Paper No. 19656, National Bureau of Economic Research, available at: https://doi.org/10.3386/w19656.

Heflin, J. (2005), “Az OWL Web-ontológianyelv - Alkalmazási esetek és követelmények”, translated by Pataki, E., 25 April, available at: http://www.w3c.hu/forditasok/OWL/REC-webont-req-20040210.html (accessed 6 April 2017).

Hoang, P., Mahoney, T., Javed, F. and McNair, M. (2018), “Large-Scale Occupational Skills Normalization for Online Recruitment”, *AI Magazine*, Vol. 39 No. 1, pp. 5–14, available at: https://doi.org/10.1609/aimag.v39i1.2775.

Hoeber, O. and Liu, H. (2011), “A Luhn-Inspired Vector Re-weighting Approach for Improving Personalized Web Search”, *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference On*, Vol. 3, IEEE Computer Society, Los Alamitos, CA, USA, pp. 301–305, available at: https://doi.org/10.1109/WI-IAT.2011.130.

Hunter, J.D. (2007), “Matplotlib: A 2D graphics environment”, *Computing in Science & Engineering*, IEEE COMPUTER SOC, Vol. 9 No. 3, pp. 90–95, available at: https://doi.org/10.1109/MCSE.2007.55.

Incze, L., Gyepesi, G. and Simon, E. (2005), “Jmorph dokumentáció [ONLINE]”, available at: http://jhunlang.sourceforge.net/jmorph/jmorph/index.html (accessed 15 December 2013).

Inmon, W.H. (2005), *Building the Data Warehouse*, John Wiley & Sons, Online, O’reilly, available at: https://www.oreilly.com/library/view/building-the-data/9780764599446/ (accessed 1 January 2021).

Inmon, W.H. and Linstedt, D. (2014), *Data Architecture: A Primer for the Data Scientist: Big Data, Data Warehouse and Data Vault*, Morgan Kaufmann.

Jarmul, K. and Lawson, R. (2017), *Python Web Scraping*, 2nd ed., Packt Publishing.

Jurafsky, D. and Martin, J.H. (2018), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3rd ed. draft., Draft, available at: https://web.stanford.edu/~jurafsky/slp3/.

Keil, J.M. (2019), “Efficient Bounded Jaro-Winkler Similarity Based Search”, in Grust, T., Naumann, F., Böhm, A., Lehner, W., Härder, T., Rahm, E., Heuer, A., et al. (Eds.), *BTW 2019*, Gesellschaft für Informatik, Bonn, pp. 205–214, available at: https://doi.org/10.18420/btw2019-13.

Kennedy, D., Hyland, Á. and Ryan, N. (2007), *Writing and Using Learning Outcomes: A Practical Guide*, University College Cork, Cork.

Khalid, M.A., Jijkoun, V. and de Rijke, M. (2008), "The Impact of Named Entity Normalization on Information Retrieval for Question Answering", in Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I. and White, R.W. (Eds.), *Advances in Information Retrieval*, Springer, Berlin, Heidelberg, pp. 705–710, available at: https://doi.org/10.1007/978-3-540-78646-7_83.

Khalil, S. and Fakir, M. (2017), "RCrawler: An R package for parallel web crawling and scraping", *SoftwareX*, Vol. 6, pp. 98–106, available at: https://doi.org/10.1016/j.softx.2017.04.004.

Khazaei, H., Fokaefs, M., Zareian, S., Beigi-Mohammadi, N., Ramprasad, B., Shtern, M., Gaikwad, P., *et al.* (2016), "How do I choose the right NoSQL solution? A comprehensive theoretical and experimental survey", *Big Data & Information Analytics*, Vol. 1 No. 2 & 3, p. 185, available at: https://doi.org/10.3934/bdia.2016004.

Kher, S. (2016), "Designing a Database for an Online Job Portal", *Vertabelo*, 15 November, available at: https://www.vertabelo.com/blog/technical-articles/designing-a-database-for-an-online-job-portal (accessed 13 May 2019).

Kher, S. (2017), "Improving Our Online Job Portal Data Model", *Vertabelo*, 11 January, available at: https://www.vertabelo.com/blog/technical-articles/improving-our-online-job-portal-data-model (accessed 13 May 2019).

Kimball, R. and Ross, M. (2013), *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd Edition., John Wiley & Sons, Online, O'reilly,

available at: https://www.oreilly.com/library/view/the-data-warehouse/9781118530801/ (accessed 2 January 2021).

King, I. (2020), "An Ancient Computer Language Is Slowing America's Giant Stimulus", *Bloomberg.Com*, 13 April, available at: https://www.bloomberg.com/news/articles/2020-04-13/an-ancient-computer-language-is-slowing-america-s-giant-stimulus (accessed 17 January 2021).

Kleppmann, M. (2015), "A Critique of the CAP Theorem", *ArXiv Preprint ArXiv:1509.05393 [Cs]*, available at: https://doi.org/10.17863/CAM.13083.

Kő, A. (2013), "Szövegbányászat és véleményfeltárás [PowerPoint prezentáció]".

Koper, R. and Tattersall, C. (2004), "New directions for lifelong learning using network technologies", *British Journal of Educational Technology*, Vol. 35 No. 6, pp. 689–700, available at: https://www.academia.edu/5128157/New_directions_for_lifelong_learning_using_network_technologies (accessed 1 May 2019).

Kovács, E. (2014), *Többváltozós Adatelemzés*, Typotex Kiadó, Budapest.

Krekó, P. (2018), *Tömegparanoia - Az összeesküvés-elméletek és álhírek szociálpszichológiája*, Athenaeum.

Laal, M. (2011), "Lifelong Learning: What does it Mean?", *Procedia - Social and Behavioral Sciences*, Vol. 28, pp. 470–474, available at: https://doi.org/10.1016/j.sbspro.2011.11.090.

Lane, H., Hapke, H. and Howard, C. (2019), *Natural Language Processing in Action: Understanding, Analyzing, and Generating Text with Python*, 1 edition., Manning Publications, Shelter Island, NY.

Lawson, R. (2015), *Web Scraping with Python*, Packt Publishing, Online, available at: https://learning.oreilly.com (accessed 11 August 2019).

Lee, K.-F. (2018), *How AI Can Save Our Humanity*, available at: https://www.ted.com/talks/kai_fu_lee_how_ai_can_save_our_humanity (accessed 14 September 2019).

Lemaître, G., Nogueira, F. and Aridas, C.K. (2017), "Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning", *The Journal of Machine Learning Research*, Vol. 18 No. 1, pp. 559–563, available at: https://dl.acm.org/doi/10.5555/3122009.3122026.

Li, B. and Han, L. (2013), "Distance Weighted Cosine Similarity Measure for Text Classification", in Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., et al. (Eds.), *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, Springer, Berlin, Heidelberg, pp. 611–618, available at: https://doi.org/10.1007/978-3-642-41278-3_74.

Liu, J., Shang, J., Wang, C., Ren, X. and Han, J. (2015), "Mining Quality Phrases from Massive Text Corpora", *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM, New York, NY, USA, pp. 1729–1744, available at: https://doi.org/10.1145/2723372.2751523.

Loper, E. and Bird, S. (2002), "NLTK: The Natural Language Toolkit", *ArXiv:Cs/0205028*, available at: https://doi.org/10.3115/1118108.1118117.

Maaten, L. van der and Hinton, G. (2008), "Visualizing Data using t-SNE", *Journal of Machine Learning Research*, Vol. 9 No. 86, pp. 2579–2605, available at: http://jmlr.org/papers/v9/vandermaaten08a.html (accessed 27 December 2020).

Manning, C.D., Raghavan, P. and Schütze, H. (2009), *Introduction to Information Retrieval*, Online edition., Cambridge University Press, New York.

Montalenti, A. (2012), “Web Crawling & Metadata Extraction in Python”, 27 October, available at: https://speakerdeck.com/amontalenti/web-crawling-and-metadata-extraction-in-python (accessed 9 August 2019).

Morville, P. and Rosenfeld, L. (2006), *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites*, 3rd ed., O’Reilly Media, Sebastopol, CA.

Myers, D. and McGuffee, J. (2015), “Choosing Scrapy”, *Journal of Computing Sciences in Colleges*, Vol. 31, pp. 83–89.

Naqvi, S.N.Z., Yfantidou, S. and Zimányi, E. (2017), *Time Series Databases and Influxdb, Advanced Databases Winter Semester 2017-2018*, Studienarbeit, Université Libre de Bruxelles, available at: https://cs.ulb.ac.be/public/_media/teaching/influxdb_2017.pdf.

Nasir, S.A.M., Yaacob, W.F.W. and Aziz, W.A.H.W. (2020), “Analysing Online Vacancy and Skills Demand using Text Mining”, *Journal of Physics: Conference Series*, IOP Publishing, Vol. 1496, p. 12, available at: https://doi.org/10.1088/1742-6596/1496/1/012011.

Navarro, G. (2001), “A guided tour to approximate string matching”, *ACM Computing Surveys*, Vol. 33 No. 1, pp. 31–88, available at: https://doi.org/10.1145/375360.375365.

Nayak, A., Poriya, A. and Poojary, D. (2013), “Type of NOSQL Databases and its Comparison with Relational Databases”, *International Journal of Applied Information Systems*, Vol. 5 No. 4, pp. 16–19.

Neches, R., Fikes, R.E., Finin, T., Gruber, T., Patil, R., Senator, T. and Swartout, W.R. (1991), “Enabling Technology for Knowledge Sharing”, *AI Magazine*, Vol. 12 No. 3, p. 36, available at:

https://www.aaai.org/ojs/index.php/aimagazine/article/view/902 (accessed 20 April 2017).

Neusch, G. (2014), *Folyamatokban rejlő tudás leképezése szemantikus technológiákkal*, Master thesis, Corvinus University of Budapest, Budapest.

Neusch, G. and Gábor, A. (2014), "ProKEX – Integrated Platform for Process-Based Knowledge Extraction", in Gómez Chova, L., López Martínez, A. and Candel Torres, I. (Eds.), *ICERI2014 Proceedings*, presented at the 7th International Conference on Education Research and Innovation, IATED Academy, Seville, Spain.

Nunes, B.P. (2014), *Towards a Well-Interlinked Web through Matching and Interlinking Approaches*, Doctoral Thesis, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Rio de Janeiro, available at: http://www.dbd.puc-rio.br/pergamum/tesesabertas/1012681_2014_completo.pdf (accessed 27 November 2020).

Oracle. (2019), "MySQL 8.0 Reference Manual", 9 0, available at: https://dev.mysql.com/doc/refman/8.0/en/ (accessed 11 August 2019).

Pataki, B. (2018), "A mintapéldákból tanuló számítógépes program (egyik lehetőség): döntési fák", Lecturepresented at the Intelligens orvosi műszerek VIMIA023, available at: http://www.mit.bme.hu/oktatas/targyak/vimia023 (accessed 21 December 2020).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., *et al.* (2011), "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, available at: https://dl.acm.org/doi/10.5555/1953048.2078195.

Pelkonen, T., Franklin, S., Teller, J., Cavallaro, P., Huang, Q., Meza, J. and Veeraraghavan, K. (2015), "Gorilla: a fast, scalable, in-memory time series database", *Proceedings of the VLDB Endowment*, Vol. 8 No. 12, pp. 1816–1827, available at: https://doi.org/10.14778/2824032.2824078.

Peterson, N.G., Mumford, M.D., Borman, W.C., Jeanneret, P.R., Fleishman, E.A., Levin, K.Y., Campion, M.A., *et al.* (2001), "Understanding Work Using the Occupational Information Network (o*net): Implications for Practice and Research", *Personnel Psychology*, Vol. 54 No. 2, pp. 451–492, available at: https://doi.org/10.1111/j.1744-6570.2001.tb00100.x.

Pitukhin, E., Varfolomeyev, A. and Tulaeva, A. (2016), "JOB ADVERTISEMENTS ANALYSIS FOR CURRICULA MANAGEMENT: THE COMPETENCY APPROACH", *ICERI2016 Proceedings*, presented at the 9th annual International Conference of Education, Research and Innovation, IATED, Seville, Spain, pp. 2026–2035, available at: https://doi.org/10.21125/iceri.2016.1456.

Porter, M.F. (1980), "An algorithm for suffix stripping", *Program: Electronic Library & Information Systems*, Vol. 40 No. 3, pp. 211–218.

Presser, M. (2017), *Data Warehousing with Greenplum*, O'Reilly Media, Inc., available at: https://www.oreilly.com/library/view/data-warehousing-with/9781491983539/ (accessed 11 May 2019).

Qi, P., Zhang, Y., Zhang, Y., Bolton, J. and Manning, C.D. (2020), "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, presented at the ACL, Association for

Computational Linguistics, Online, pp. 101–108, available at: https://doi.org/10.18653/v1/2020.acl-demos.14.

R. L. Cilibrasi and P. M. B. Vitanyi. (2007), “The Google Similarity Distance”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19 No. 3, pp. 370–383, available at: https://doi.org/10.1109/TKDE.2007.48.

Ratcliff, J.W. and Metzener, D.E. (1988), “Pattern-matching-the gestalt approach”, *Dr Dobbs Journal*, MILLER FREEMAN, INC 411 BOREL AVE, SAN MATEO, CA 94402-3522, Vol. 13 No. 7, p. 46.

Rohaidi, N. (2016), “IBM’s Watson Detected Rare Leukemia In Just 10 Minutes”, *AsianScientist*, available at: https://www.asianscientist.com/2016/08/topnews/ibm-watson-rare-leukemia-university-tokyo-artificial-intelligence/ (accessed 19 July 2019).

Russell, S. and Norvig, P. (2005), *Mesterséges Intelligencia Modern Megközelítésben*, Panem Kft., Budapest, available at: https://mialmanach.mit.bme.hu/aima/index (accessed 22 April 2017).

Sadalage, P.J. and Fowler, M. (2012), *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, Addison-Wesley, available at: https://dl.acm.org/doi/book/10.5555/2381014.

Sampson, S.E. (2018), “Professional Service Jobs: Highly Paid but Subject to Disruption?”, *Service Science*, Vol. 10 No. 4, pp. 457–475, available at: https://doi.org/10.1287/serv.2018.0227.

Sántáné-Tóth, E. (2001), “Ontológia - Oktatási segédlet”, available at: http://people.inf.elte.hu/santa/oktatasi-anyagok/segedletek-pdf/segedlet-5.doc (accessed 5 November 2013).

Sántáné-Tóth, E., Bíró, M., Gábor, A., Kő, A. and Lovrics, L. (2007), *Döntéstámogató Rendszerek*, Panem, Budapest.

Scorza, P., Araya, R., Wuermli, A.J. and Betancourt, T.S. (2016), "Towards Clarity in Research on 'Non-Cognitive' Skills: Linking Executive Functions, Self-Regulation, and Economic Development to Advance Life Outcomes for Children, Adolescents and Youth Globally", *Human Development*, Vol. 58 No. 6, pp. 313–317, available at: https://doi.org/10.1159/000443711.

Sebők, M., Kubik, B., Molnár, C., M. Balázs, Á., Vancso, A., Zorigt, B., Zágoni, B., *et al.* (2016), *Kvantitatív Szövegelemzés És Szövegbányászat a Politikatudományban*, L'Harmattan Kiadó, Budapest.

Seeger, M. (2009), "Key-value stores : A pracitical overview", *Computer Science and Media. Ultra-Large-Sites, 2009*, Vol. 9, pp. 1–21, available at: http://blog.marc-seeger.de/assets/papers/Ultra_Large_Sites_SS09-Seeger_Key_Value_Stores.pdf (accessed 3 January 2021).

Shannon, C.E. (1948), "A mathematical theory of communication", *The Bell System Technical Journal*, presented at the The Bell System Technical Journal, Vol. 27 No. 3, pp. 379–423, available at: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Sike, S. and Varga, L. (2003), *Szoftvertechnológia És UML*, ELTE Eötvös Kiadó, Budapest, available at: https://lexybunnyy.web.elte.hu/elte/felev5/szoftvertechnologia/progtechUMLkonyv.pdf (accessed 18 April 2017).

Singh, C. and Kumar, M. (2019), *Mastering Hadoop 3: Big Data Processing at Scale to Unlock Unique Business Insights*, 1st edition., Packt Publishing, available

at: https://www.oreilly.com/library/view/mastering-hadoop-3/9781788620444/ (accessed 10 January 2021).

Smedt, J.D., Vrang, M. le and Papantoniou, A. (2015), "ESCO: Towards a Semantic Web for the European Labor Market", presented at the www2015 workshop: Linked Data on the Web (LDOW2015), Florence, Italy.

Smith, N.A. (2011), "Linguistic Structure Prediction", *Synthesis Lectures on Human Language Technologies*, Vol. 4 No. 2, pp. 1–274, available at: https://doi.org/10.2200/S00361ED1V01Y201105HLT013.

Sneftel. (2013), "semantic web - Ontology vs vocabulary", *Stack Overflow*, 25 November, available at: https://stackoverflow.com/questions/20200270/ontology-vs-vocabulary) (accessed 26 June 2019).

Spear, A., Ceusters, W. and Smith, B. (2016), "Functions in Basic Formal Ontology", *Applied Ontology*, Vol. 11 No. 2, pp. 103–128, available at: https://doi.org/10.3233/AO-160164.

"Stack Overflow Developer Survey 2020". (2020), *Stack Overflow*, available at: https://insights.stackoverflow.com/survey/2020/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2020 (accessed 9 January 2021).

Studer, R., Benjamins, V.R. and Fensel, D. (1998), "Knowledge Engineering: Principles and Methods", *Data & Knowledge Engineering*, Vol. 25 No. 1–2, pp. 161–197, available at: https://doi.org/10.1016/S0169-023X(97)00056-6.

Szabó, I. and Neusch, G. (2015), "Dynamic Skill Gap Analysis Using Ontology Matching", in Kő, A. and Francesconi, E. (Eds.), *Electronic Government and the Information Systems Perspective*, Vol. 9265, Springer International

Publishing, pp. 231–242, available at: http://dx.doi.org/10.1007/978-3-319-22389-6_17.

Szabó Z. (2000), *A szervezeti információfeldolgozás strukturális és technológiai tényezőinek összerendelése*, phd, Budapesti Corvinus Egyetem, available at: http://phd.lib.uni-corvinus.hu/212/ (accessed 24 January 2021).

Szekeres, P. (2013), "Szövegbányászat és véleményelemzés [PowerPoint prezentáció]".

Szirmai M. (2005), *Bevezetés a korpusznyelvészetbe: a korpusznyelvészet alkalmazása az anyanyelv és az idegen nyelv tanulásában és tanításában*, Tinta, Budapest.

Tan, P.-N., Steinbach, M. and Kumar, V. (2011), *Bevezetés az adatbányászatba - elektronikus kiadás*, Panem Könyvkiadó Kft., Budapest, Magyarország, available at: https://regi.tankonyvtar.hu/hu/tartalom/tamop425/0046_adatbanyaszat/adatok.html (accessed 3 November 2020).

Thomas, R. (2019a), "What is NLP?", available at: https://github.com/fastai/course-nlp (accessed 14 September 2019).

Thomas, R. (2019b), "Topic Modeling with NMF and SVD", available at: https://github.com/fastai/course-nlp (accessed 15 September 2019).

Tikk, D., Farkas, R., Kardkovács, Z.T., Kovács, L., Répási, T., Szarvas, G., Szaszkó, S., *et al.* (2007), *Szövegbányászat*, edited by Tikk, D., Typotex.

Tordai, A. and de Rijke, M. (2005), "Four Stemmers and a Funeral: Stemming in Hungarian at CLEF 2005", in Peters, C. (Ed.), *Lecture Notes in Computer Science*, Vol. 4022, presented at the Conference: Accessing Multilingual Information Repositories, 6th Workshop of the Cross-Language Evalution

Forum, Springer, Berlin, Heidelberg, pp. 179–186, available at: https://doi.org/10.1007/11878773_20.

Török, M. (2014), *Organizational Knowledge Extraction from Business Process Models*, PhD thesis, Budapesti Corvinus Egyetem, available at: http://phd.lib.uni-corvinus.hu/817/ (accessed 13 September 2019).

Tudorica, B.G. and Bucur, C. (2011), "A comparison between several NoSQL databases with comments and notes", *2011 RoEduNet International Conference 10th Edition: Networking in Education and Research*, presented at the 2011 RoEduNet International Conference 10th Edition: Networking in Education and Research, pp. 1–5, available at: https://doi.org/10.1109/RoEduNet.2011.5993686.

Ukkonen, E. (1985), "Algorithms for approximate string matching", *Information and Control*, Vol. 64 No. 1, pp. 100–118, available at: https://doi.org/10.1016/S0019-9958(85)80046-2.

Ukkonen, E. (1992), "Approximate string-matching with q-grams and maximal matches", *Theoretical Computer Science*, Vol. 92 No. 1, pp. 191–211, available at: https://doi.org/10.1016/0304-3975(92)90143-4.

Vadász N. and Simon E. (2019), "Konverterek magyar morfológiai címkekészletek között", *Magyar Számítógépes Nyelvészeti Konferencia*, Vol. 13, presented at the Magyar Számítógépes Nyelvészeti Konferencia (15.) (2019) (Szeged), pp. 99–111, available at: http://acta.bibl.u-szeged.hu/59077/ (accessed 19 November 2020).

Varga, K. (2014), *A Szemantikus Folyamatmenedzsment Hasznosítási Lehetősége Az Üzleti Folyamatok Tudásalapú Fejlesztésében*, PhD thesis, Budapesti Corvinus

Egyetem, available at: http://phd.lib.uni-corvinus.hu/818/ (accessed 13 September 2019).

Vas, R. (2016), "STUDIO: Ontology-Centric Knowledge-Based System", in Gábor, A. and Kő, A. (Eds.), *Corporate Knowledge Discovery and Organizational Learning*, Springer International Publishing, pp. 83–103, available at: https://doi.org/10.1007/978-3-319-28917-5_4.

Vas, R., Kovacs, B. and Kismihok, G. (2009), "Ontology-based mobile learning and knowledge testing", *International Journal of Mobile Learning and Organisation*, Vol. 3 No. 2, p. 128, available at: https://doi.org/10.1504/IJMLO.2009.024423.

Vas, R.F. (2007), *Tudásfelmérést Támogató Oktatási Ontológia Szerepe És Alkalmazási Lehetőségei*, PhD thesis, Budapesti Corvinus Egyetem, available at: http://phd.lib.uni-corvinus.hu/258/.

Verma, V. and Aggarwal, R.K. (2020), "A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: empirical and theoretical perspective", *Social Network Analysis and Mining*, Vol. 10 No. 1, p. 43, available at: https://doi.org/10.1007/s13278-020-00660-9.

Vincze V., Szauter D., Almási A., Móra G., Alexin Z. and Csirik J. (2009), "A Szeged Treebank függőségi fa formátumban", *Magyar Számítógépes Nyelvészeti Konferencia*, presented at the Magyar Számítógépes Nyelvészeti Konferencia (6.) (2009) (Szeged), Vol. 4, pp. 127–138, available at: http://acta.bibl.u-szeged.hu/58703/ (accessed 19 November 2020).

Vincze, V., Szauter, D., Almási, A., Móra, G., Alexin, Z. and Csirik, J. (2010), "Hungarian Dependency Treebank", in Chair), N.C. (Conference, Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., et al. (Eds.),

Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), presented at the LREC 2010, Seventh International Conference on Language Resources and Evaluation, European Language Resources Association (ELRA), Valletta, Malta, pp. 19–21.

Vitányi, P.M., Balbach, F.J., Cilibrasi, R.L. and Li, M. (2009), "Normalized information distance", *Information Theory and Statistical Learning*, Springer, pp. 45–82.

Voronov, N. (2020), *TextDistance*, Python, Github Repository, available at: https://github.com/life4/textdistance.

Vrang, M. le, Papantoniou, A., Pauwels, E., Fannes, P., Vandensteen, D. and Smedt, J.D. (2014), "ESCO: Boosting Job Matching in Europe with Semantic Interoperability", *Computer*, Vol. 47 No. 10, pp. 57–64, available at: https://doi.org/10.1109/MC.2014.283.

Waldrop, M.M. (2016), "The chips are down for Moore's law", *Nature News*, Vol. 530 No. 7589, pp. 144–147, available at: https://doi.org/10.1038/530144a.

Weber, C. and Vas, R. (2015), "Studio: Ontology-Based Educational Self-Assessment", *Workshops Proceedings of EDM 2015 8th International Conference on Educational Data Mining, EDM 2015, Madrid, Spain, June 26-29, 2015.*, Vol. 1446, Madrid, Spain, pp. 33–40, available at: http://ceur-ws.org/Vol-1446/GEDM_2015_Submission_5.pdf.

Witten, I.H., Frank, E. and Hall, M.A. (2011), *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed., Elsevier Science, available at: http://books.google.hu/books?id=bDtLM8CODsQC.

Wolters Kluwer Kft. (2016), "18/2016. (VIII. 5.) EMMI rendelet - 1.oldal - Hatályos Jogszabályok Gyűjteménye", available at:

https://net.jogtar.hu/jogszabaly?docid=A1600018.EMM&timeshift=fffffff4&txtreferer=00000001.TXT (accessed 19 August 2018).

Wowczko, I.A. (2015), “Skills and Vacancy Analysis with Data Mining Techniques”, *Informatics*, Vol. 2 No. 4, pp. 31–49, available at: https://doi.org/10.3390/informatics2040031.

Zhao, M., Javed, F., Jacob, F. and McNair, M. (2015), “SKILL: a system for skill identification and normalization”, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI Press, Austin, Texas, pp. 4012–4017, available at: https://dl.acm.org/doi/10.5555/2888116.2888273.

10 Mellékletek

1. Melléklet: Adatbázis-séma jelmagyarázata

Az 12. ábrán bemutatott egyed-kapcsolat diagramm a *MySQL Workbench* alkalmazás 8.0 verziójával készült. A diagrammon látható legfontosabb jelölések a következők:

- Elsődleges kulcs
- Idegen kulcs – elsődleges kulcs
- Egyszerű kötelező attribútum
- Kötelező idegen kulcs
- Egyszerű opcionális attribútum
- Opcionális idegen kulcs

Bővebb információ az alkalmazás dokumentációjában található (Oracle, 2019).

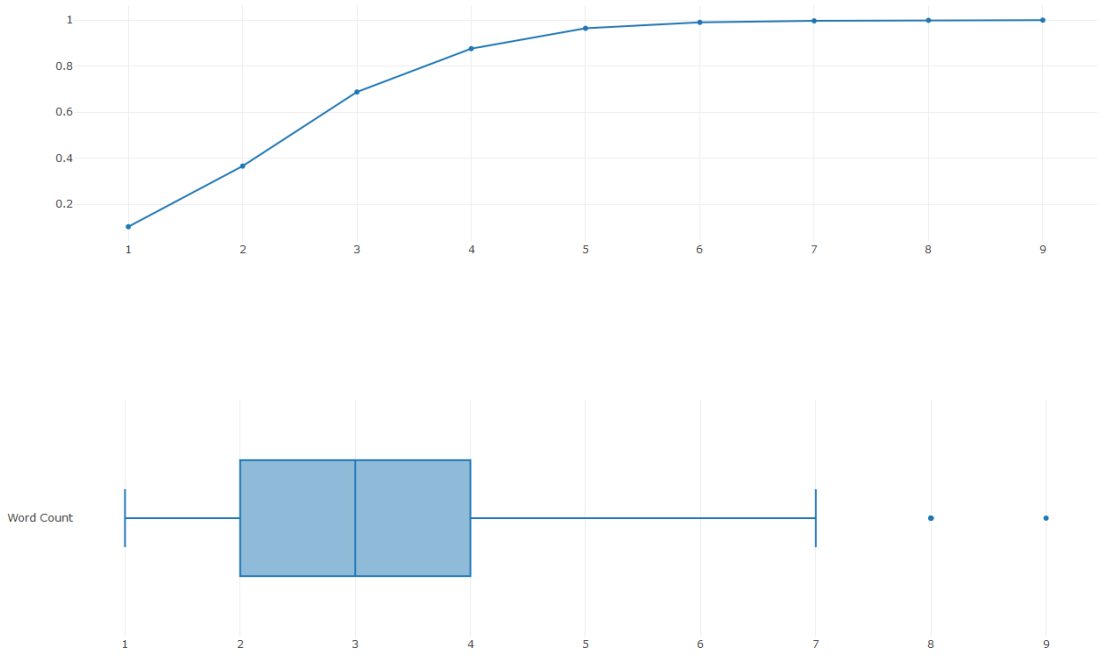
2. Melléklet: A kompetenciaszótár leíró statisztikái

Szavak száma	f_i	g_i	g'_i
1	226	0.107	0.107
2	535	0.254	0.361
3	640	0.303	0.664
4	303	0.144	0.808
5	195	0.092	0.9
6	114	0.054	0.954
7	72	0.034	0.988
8	16	0.008	0.996
9	2	0.001	0.997
10	2	0.001	0.998
11	3	0.001	0.999
12	2	0.001	1

13. táblázat: Kompetenciaszótár kifejezéshosszak gyakoriságai

Szavak száma	fi	gi	g'i
1	214	0.102	0.102
2	553	0.2636	0.3656
3	676	0.3222	0.6878
4	395	0.1883	0.8761
5	186	0.0887	0.9647
6	54	0.0257	0.9905
7	14	0.0067	0.9971
8	5	0.0024	0.9995
9	1	0.0005	1

14. táblázat: Kompetenciaszótár kifejezéshosszak gyakoriságai stopszavak eltávolítása után



33. ábra: Kompetenciaszótár elemhossz gyakoriságok vizuális ábrázolása a stopszavak eltávolítása után

3. Melléklet: Kompetenciaelemek beazonosítása szótár alapján

A 1. lista azon kifejezéseket tartalmazza – feldolgozás utáni formájukban – a kompetenciaszótárból, melyek a stopszavak eltávolítása után kerültek beazonosításra. A 2. listában egy további, lemmatizációs lépés után feltárt szókapcsolatok kaptak helyet.

1. lista: ["quality assurance arrangements", "use office systems", "computer using", "components hardware", "look data", "back data", "use handheld devices", "search database", "manage environmental impact", "service portfolio management", "financial management services", "document maintenance", "minimise environmental impact", "principles data protection", "utilise tools", "inspection data", "utilize tools", "implementing firewall", "use computer equipment", "research databases", "organizational structure", "use tools", "auditor", "audit process", "soft skills", "business systems", "audit", "regulations", "governance", "standard"]

2. lista: ["filemaker", "continuous improvement strategy", "manage product data", "kali linux", "use electronic service", "c sharp", "tqc", "failure testing", "aspx", "scan document", "ensure information privacy", "project closing", "use instant message", "use cmm", "file transfer protocol", "electronic business", "perform data mining", "use ict tool", "use cam software", "web 2.0 technology", "application layer protocol", "category media", "accessibility standard guideline", "look database", "e commerce organisation", "analyse big data", "infrastructure audit", "c language", "http cookie", "develop information security strategy", "analytical crm", "system development standard", "communicate online", "5 g", "use geographical information system", "burpsuite", "develop data set", "perform security vulnerability assessment", "operation audit", "testing level", "develop digital content", "implement information security program", "strategic performance indicator", "operate cmm", "keep stock control system", "technique handling", "system integration strategy", "php7", "define problem area", "service measurement", "ict security standard", "principle data privacy", "manage system problem", "use information communication technology", "sql server 2005", "diagnose system problem", "rdb", "nexpose", "repeater", "triplestore", "customer relationship management solution", "waterfall model", "system design procedure", "use query language", "quality management standard", "component engineering", "genetic algorithm", "sap data service", "able use database", "owasp zap", "control type", "cobol", "write database documentation", "java object oriented", "museum archive", "use communication equipment", "haskell", "sql server 2014", "office operating system", "ethereum", "routing table", "diagram circuit", "user requirement specification", "adobe illustrator", "implement vpn", "adobe photoshop", "nessus", "metasploit", "odi", "semantic technology", "pdi", "tqm", "network security standard", "scale network", "requirement prioritization", "business impact analysis", "python3", "information security process", "perform backup", "office programme", "use gis", "penetration testing tool", "abap", "project management standard", "sql server integration service", "cpi", "olap", "technical term", "ict security", "jboss", "sql server 2008", "sql server 2012", "ip address", "testing method", "scala", "testing procedure", "ict infrastructure", "etl tool", "sap erp", "project priority", "bcp", "ajax", "sem", "postgresql", "nlp", "cobit", "autocad", "gps", "ssis", "owasp", "ux design", "process control", "nosql", "matlab", "security compliance", "mysql", "iso27001", "iot", "etl", "gis", "crm system", "apis", "outlook", "erp", "ict"]

4. Melléklet: Kompetenciakifejezések valószínűség alapú beazonosítása

Correlations

		Cosine nts	Jaro nts	JaroWinkler nts	RatcliffObershelp nts
Cosine nts	Pearson Correlation	1	.327**	.233**	.918**
	Sig. (2-tailed)		0.000	0.000	0.000
	N	1724	1724	1724	1724
Jaro nts	Pearson Correlation	.327**	1	.981**	.322**
	Sig. (2-tailed)	0.000		0.000	0.000
	N	1724	1724	1724	1724
JaroWinkler nts	Pearson Correlation	.233**	.981**	1	.245**
	Sig. (2-tailed)	0.000	0.000		0.000
	N	1724	1724	1724	1724
RatcliffObershelp nts	Pearson Correlation	.918**	.322**	.245**	1
	Sig. (2-tailed)	0.000	0.000	0.000	
	N	1724	1724	1724	1724

**. Correlation is significant at the 0.01 level (2-tailed).

15. táblázat: Átfedő tartalmú hasonlósági mutatószámok

Area Under the Curve

Test Result Variable(s):

Area	Std. Error[a]	Asymptotic Sig.[b]	Asymptotic 95% Confidence Interval Lower Bound	Asymptotic 95% Confidence Interval Upper Bound
0.977	0.003	0.000	0.971	0.982

The test result variable(s): Predicted probability has at least one tie between the positive actual state group and the negative actual state group. Statistics may be biased.

a. Under the nonparametric assumption

b. Null hypothesis: true area = 0.5

16. táblázat: A ROC görbe alatti terület

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)	95% C.I.for EXP(B) Lower	Upper
Step 1[a]	NskillCat			38.727	3	0.000			
	NskillCat(1)	-3.912	0.708	30.537	1	0.000	0.020	0.005	0.080
	NskillCat(2)	-5.904	1.314	20.176	1	0.000	0.003	0.000	0.036
	NskillCat(3)	-3.600	0.686	27.533	1	0.000	0.027	0.007	0.105
	NpostingCat			37.457	3	0.000			
	NpostingCat(1)	7.750	2.516	9.486	1	0.002	2320.771	16.745	########
	NpostingCat(2)	5.453	2.538	4.618	1	0.032	233.513	1.615	33763.639
	NpostingCat(3)	8.443	2.839	8.847	1	0.003	4641.682	17.802	########
	Jaccard token nts	23.213	5.231	19.689	1	0.000	########	########	########
	Cosine nts	5.158	2.071	6.200	1	0.013	173.748	2.998	10070.391
	JaroWinkler nts	1.333	0.587	5.162	1	0.023	3.792	1.201	11.975
	Constant	-19.700	3.919	25.268	1	0.000	0.000		

a. Variable(s) entered on step 1: NskillCat, NpostingCat, Jaccard token nts, Cosine nts, JaroWinkler nts.

17. táblázat: A modell változói és a parciális illeszkedést jelző szignifikancia szintek

```python
def calculate_probability(jaccard_token, cosine, jarowinkler, skilllengthcat, postinglengthcat):
    constant = -19.700336

    b_skill_cat = {
        '1': -3.912355,
        '2': -5.904053,
        '3': -3.600454,
        '3+': 0
    }

    b_posting_cat = {
        '1': 0,
        '2': 7.749655,
        '3': 5.453237,
        '3+': 8.442832
    }

    log_odds = (
        constant +
        (23.212593 * jaccard_token) +
        (5.157607 * cosine) +
        (1.332940 * jarowinkler) +
        b_skill_cat[skilllengthcat] +
        b_posting_cat[postinglengthcat]
    )

    return 1 / (1 + math.exp(-log_odds))
```

34. ábra: A logit modell eredményei alapján a becslést végző függvény Python nyelvű implementációja

Chi-Square Tests

	Value	df	Asymptotic Significance (2-sided)	Exact Sig. (2-sided)	Exact Sig. (1-sided)
Pearson Chi-Square	15952.869[a]	1	0.000		
Continuity Correction[b]	15946.639	1	0.000		
Likelihood Ratio	11502.096	1	0.000		
Fisher's Exact Test				0.000	0.000
Linear-by-Linear Association	15952.311	1	0.000		
N of Valid Cases	28597				

a. 0 cells (0.0%) have expected count less than 5. The minimum expected count is 555.57.
b. Computed only for a 2x2 table

18. táblázat: A tesztadatok automatikus és a manuális besorolásait tartalmazó változók függetlenségét minden szignifikancia szinten elvethetjük

Symmetric Measures

		Value	Approximate Significance
Nominal by Nominal	Phi	0.747	0.000
	Cramer's V	0.747	0.000
	Contingency Coefficient	0.598	0.000
N of Valid Cases		28597	

19. táblázat: A tesztadatok automatikus és a manuális besorolásait tartalmazó változók között a közepesnél erősebb kapcsolat figyelhető meg

5. Melléklet: Foglalkozások beazonosítása reguláris kifejezésekkel

1. lista, prefixumok: ["Cheaf", "Lead", "Leader", "Junior", "Senior", "Medior", "Apprentice", "Sr.", "Sr", "Jr", "Jr.", "Graduate", "Principal", "Trainee", "Undergraduate", "Assistant", "Temporary", "Contract", "Contractor", "Director of", "Lecturer in", "Head of", "1st Line", "2nd Line", "3rd Line", "Level 1", "Level 2", "Level 3", "1st 2nd Line", "2nd 3rd Line"]

2. lista, szuffixumok: ["Administrator", "Advisor", "Analyst", "Architect", "Assistant", "Associate", "Consultant", "Co-ordinator", "Designer", "Developer", "Director", "Engineer", "Expert", "Helpdesk", "Intern", "Manager", "Officer", "Operator", "Planner", "Scientist", "Specialist", "Supervisor", "Technician","Technologist", "Tester", "Apprenticeship", "Apprentice", "Student"]

11 Publikációs lista

Referált szakmai folyóirat

Neusch, G. (2014), “Domain Ontology Tailoring Based on Business Processes in the Frame of the ProKEX Project”, *SEFBIS Journal*, Vol. I No. XI, pp. 51–59.

Szabó, I. and Neusch, G. (2015), “Dynamic Skill Gap Analysis Using Ontology Matching”, in Kő, A. and Francesconi, E. (Eds.), *Electronic Government and the Information Systems Perspective*, Vol. 9265, Springer International Publishing, pp. 231–242, available at: http://dx.doi.org/10.1007/978-3-319-22389-6_17.

Beel, J., Carevic, Z., Schaible, J. and Neusch, G. (2017), “RARD: The Related-Article Recommendation Dataset”, *D-Lib Magazine*, Vol. 23 No. 7/8, available at: https://doi.org/10.1045/july2017-beel.

Szabó, I., Neusch G., Vas R. (2021, megjelenés alatt), „Design Thinking based Ontology Development for Robo-Advisors”, *Proceedings of the 20th International Conference Intelligent Systems Design and Applications (ISDA 2020)*, Advances in Intelligent Systems and Computing, Springer Verlag

Lektorált konferenciakötetben megjelent tanulmányok

Neusch, G. and Gábor, A. (2014), “ProKEX – Integrated Platform for Process-Based Knowledge Extraction”, in Gómez Chova, L., López Martínez, A. and Candel Torres, I. (Eds.), *ICERI2014 Proceedings*, presented at the 7th International Conference on Education Research and Innovation, IATED Academy, Seville, Spain.

Castello, V., Mahajan, L., Flores, E., Gabor, M., Neusch, G., Szabo, I., Guerrero, J., *et al.* (2014), “THE SKILL MATCH CHALLENGE. EVIDENCES FROM THE

SMART PROJECT", in Gómez Chova, L., López Martínez, A. and Candel Torres, I. (Eds.), *ICERI2014 Proceedings*, IATED Academy.

Weber, C., Neusch, G. and Vas, R. (2016), "Studio: A Domain Ontology Based Solution for Knowledge Discovery in Learning and Assessment", *Proceedings of the 2016 AIS SIGED International Conference on Information Systems Education and Research*, pp. 1-13., available at: https://aisel.aisnet.org/siged2016/12.

Neusch, G. (2016), "Ontology Tailoring for Job Role Knowledge", in Gábor, A. and Kő, A. (Eds.), *Corporate Knowledge Discovery and Organizational Learning*, Springer International Publishing, pp. 105–130, available at: https://doi.org/10.1007/978-3-319-28917-5_5.

Egyéb szakmai teljesítmények

Neusch, G., 2015. 'Studio-User Help' saját számítógépi programalkotás, Szellemi Tulajdon Nemzeti Hivatala, Önkéntes műnyilvántartásba vételi szám: 003871