Budapesti Mûszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

Szöveganalitikai megoldások cégek kapcsolódási hálójának felépítéséhez

TDK DOLGOZAT

Készítette
Pósfai Gergely

Konzulens
Gáspár Csaba
Távközlési és Médiainformatikai Tanszék

2013. október 25.

Tartalomjegyzék

Kivonat

A cégek, vállalatok és egyéb gazdasági szereplők életében nagy jelentőséggel bírnak a más vállalatokkal, szervezetekkel kialakított kapcsolataik. A cégek főként az üzleti partnereikkel kommunikálnak, velük kötnek stratégiai megállapodásokat, és általában velük bonyolítják le pénzügyi tranzakcióikat. Vagyis az üzleti partnerekkel kiépített viszonyok fontos információkat hordoznak a gazdasági szereplőkről, alapvetően meghatározzák működésüket, ezért érdekes és hasznos következtetéseket vonhatunk le belőlük. Azonban az üzleti megállapodások általában nem nyilvánosak, ennélfogva a kapcsolatokon alapuló következtetésekhez, elemzésekhez először azonosítani kell a vállalatközi kapcsolatokat.

A dolgozat a vállalatok és a vállalatokhoz köthető személyek közötti kapcsolati hálózat felépítésének egy lehetséges megközelítését mutatja be: a weben elérhető hírek elemzése alapján következtetünk a gazdasági szereplők közötti összeköttetésekre. Ehhez információkivonási eljárásokat hozunk létre, amelyek segítségével különböző hírportálok cikkeiben azonosítjuk a cégekre vonatkozó hivatkozásokat, ill. a cégekhez köthető alkalmazottakat, tisztségviselőket. Az így előállított információk felhasználásával építjük fel a gazdasági szereplők kapcsolati hálózatát. A munka során nagy mennyiségű, természetesen nyelven írt szövegek elemzésével állítunk elő információkat, így a részfeladatok során találkozhatunk adatbányászati, természetesnyelv-feldolgozási, ill. gépi tanulást igénylő folyamatokkal is.

A feltérképezett információk megjelenítéséhez egy webalkalmazást hoztunk létre. Az alkalmazás egy keresőfelületet nyújt a felhasználók számára, amely segítségével cégek és személyek között kereshetünk. Minden céghez és személyhez tartozik egy adatlap, amelyen megtekinthetjük a róla feltérképezett információkat: az adott gazdasági szereplő más cégekkel, ill. tisztségviselőkkel való kapcsolatait, valamint megtekinthetjük a webes híreknek azon részét, amely alapján egy adott tisztségviselőt azonosítottunk. A cégek és tisztségviselők böngészése mellett az alkalmazás lehetőséget biztosít a tévesen azonosított tisztségviselők korrigálására is. Az elvégzett korrekciókat a későbbi modellezés során felhasználjuk, amely így egyre pontosabb eredmények elérését teszi lehetővé.

Abstract

Business relationships play an important role in the operation of firms, companies and corporations. The companies mainly communicate with their partners, they also sign business agreements and administer financial transactions with them. That is, economic relations established between partners provide important information about companies and highly influence their operations, so interesting and useful conclusions can be drawn from them. However, business relationships are usually not public. In order to analyze the network of inter-company connections, each relation has to be individually identified.

This paper presents a possible approach for discovering business relations between companies and their employees. Connections between organizations and people related to them are determined based on the analysis of articles of online news portals. The references of firms, companies, organizations and office-holders are identified in the articles, then the assessed information is used to build the network of economic actors. Since an immense number of documents that were written in natural language are processed and analyzed, solving the subtasks requires the application of data mining, natural language processing and machine learning techniques.

In order to display the discovered information a web application is developed, which provides an interface for the users to search among corporations and people. Each corporation and person has a profile page, that aggregates all the information that was collected on the particular firm or person. This information includes the relationships with other organizations and employees, and also the segments of the articles that were used to identify the employees of the companies. In addition to browsing among corporations and office-holders, the application offers the possibility of correcting wrongly identified office-holders. These corrections are used in the subsequent modeling processes, which leads to more accurate results.

1. fejezet

Bevezetés

1.1. Motiváció

Minden vállalat igyekszik minél szélesebb körű üzleti kapcsolatokat kialakítani. A partnercégekkel való együttműködés növeli a vállalatok versenyképességét, stabilabbá és gazdaságosabbá teszi működésüket. Így a cégek közti kapcsolatok fontos szerepet játszanak a cégek fejlődésében, ezért a kapcsolatok azonosításával és elemzésével sok információt tudhatunk meg róluk. Ezek az információk komoly üzleti értéket jelenthetnek más gazdasági szereplők számára, akiknek fontos, hogy részletesen ismerjék a cégek működését és az azt befolyásoló tényezőket.

A cégek kapcsolatai nem érhetőek el struktúrált formában publikus adatbázisokon keresztül, így a kapcsolatokon alapuló következtetésekhez első lépésben fel kell térképezni a cégek közötti kapcsolati hálózatot. A dolgozatban ezzel a feladattal foglalkozunk.

A kapcsolati hálózat felépítésének egy lehetséges megközelítése, ha hírekben azonosítjuk a cégek említéseit, és az egy hírben közösen szereplő cégek között feltételezünk kapcsolatokat. Azonban ennél mélyebb, szorosabb kapcsolatokat találhatunk, ha azonosítjuk a cégek alkalmazottait. Ekkor a cégek közötti kapcsolati rendszer tovább finomítható. Ennek két lehetséges módja:

1. Ha az alapján kötünk össze vállalatokat, hogy van-e olyan személy, aki mindkét vállalattal kapcsolatban van.

2. Ill. ha az alkalmazottak személyes kapcsolati hálóját is bevonjuk az elemzésbe külső adatforrásként, és különböző közösségi oldalak kapcsolatain keresztül gazdagítjuk a hálózat struktúráját, mint pl.: Facebook, Twitter, LinkedIn stb.

A dolgozat során az 1. megközelítést alkalamzzuk a cégek kapcsolati hálózatának felépítéséhez, azaz akkor tekintünk kapcsolatot két cég között, ha egy alkalmazott átkerül az egyik cégtől a másikhoz, vagy mindkét cégnél egyidejűleg tölt be valamilyen pozíciót. Természetesen ez a módszer kombinálható a 2. eljárással, azonban ez már nem tárgya a dolgozatnak. A dolgozat folyamán kizárólag a cégek alkalmazottainak, tisztségviselőinek azonosításával foglalkozunk, és ez alapján építjük fel a cégközi kapcsolati hálózatot.

1.2. Alkalmazások

A nyilvánosan elérhető cégnyilvántartási adatbázisok általában csak néhány adatot tartalmaznak az egyes cégekről, mint pl.: cég neve, székhelye, tulajdonosok, éves költségvetés. Ennek segítségével felépíthetünk egy egyszerű kapcsolati hálózatot a cégekről, pl. közös tulajdonosok, azonos székhely alapján. Azonban a hírekben rejlő, informális vállalatközi kapcsolatokról nem találhatunk az adatbázisokban információkat, így a kapcsolati hálók ilyen módon történő felépítése és elemzése értékes eszköz lehet egy olyan vállalat kezében, amely számára fontos a cégek minél részletesebb ismerete. Ez különösen igaz a bankokra, amelyek számára több területen is hasznosak lehetnek ezek az információk. Ezek közül néhány példa a teljesség igénye nélkül:

- **Ügyfélakvizíció:** a bankok számára komoly bevételt jelent, ha a saját ügyfelei között jelentős pénzügyi tranzakciók zajlanak le, hiszen ekkor nem kell más bank szolgáltatásaiért fizetni, így a tranzakciók díjait nem kell megosztani. Ennélfogva a bankoknak érdemes olyan ügyfeleket keresni, amelynek partnerei már az adott bankhoz tartoznak, mivel feltehetőleg sok tranzakciót fognak egymás között lebonyolítani.
- **Ügyfélelvándorlás megelőzése:** az előző esethez hasonló okok miatt a bankok az ügyfélek elvándorlásának megelőzésére is alkalmazhatják a cégek kapcsolataira vonatkozó információkat. Ha egy bank látja, hogy egy ügyfelének szinte minden partnere egy másik bankkal áll kapcsolatban, akkor feltételezheti, hogy az ügyfélcég előbb-utóbb át fog menni a partnerei bankjához, hiszen így jóval olcsóbban tudná a partnerei felé irányuló átutalásait végrehajtani.
- **Csődkockázatbecslés:** a bankok a cégek közötti kapcsolatok alapján meg tudják becsülni, hogy egy adott cég csődje esetén milyen egyéb vállalatok kerülnek veszélyeztetett helyzetbe.

A bankok mellett természetesen más vállalatok számára is értékes információt jelenthet a cégek kapcsolatainak ismerete:

- A business-to-business világban dolgozó vállalatoknak a piacfelmérés vagy az értékesítés során hasznos adatokat nyújthat a kapcsolati gráf.
- Potenciális partnerek keresésében bármely vállalat számára hasznos lehet a vállalatok közötti kapcsolatok ismerete. Pl.: egy vállalat egy korábbi partnercéggel való együttműködés rossz tapasztalatai után feltehetőleg nem szeretne olyan új partnerekkel szerződést kötni, amelyek szoros kapcsolatban állnak az említett korábbi partnercéggel.

Ezeken az egyszerű példákon kívül számos egyéb alkalmazási lehetőséggel is találkozhatunk.

1.3. Feladatspecifikáció, részfeladatok

A feladatunk tehát a vállalatok kapcsolati hálózatának felépítése a vállalatok beosztottjainak azonosításával. A részfeladatokat négy tág, egymással szorosan összefüggő csoportra oszthatjuk:

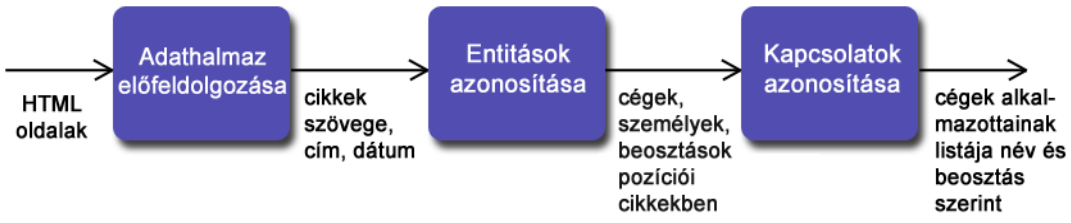
1. **Adathalmaz gyűjtése:** első lépésként szükségünk van egy nyers adathalmazra, amelyben valamilyen módon azonosítani lehet a cégek alkalmazottait. Ehhez online cikkeket használunk. Néhány hírportál összes 2011-ben megjelent cikke került letöltésre, amelyek a munka alapjául szolgáltak, így ezzel a részfeladattal nem foglalkozunk a dolgozatban.

2. **Adathalmaz előfeldolgozása:** az összegyűjtött nyers adatokban – HTML oldalakban – azonosítanunk kell a számunkra lényeges entitásokat (cégek, személyek, beosztások). Azonban a HTML dokumentumok számos lényegtelen elemet, hirdetéseket tartalmaznak, amelyek egyáltalán nem kapcsolódnak a cikkhez, így torzíthatják az eredményeket, és nagy mértékben lassíthatják a feldolgozást. Ezért az entitások azonosítása előtt meg kell keresnünk a HTML oldalakban a számunkra releváns részeket. A cikkek szövege mellett a cikkek címét és megjelenésének dátumát is azonosítjuk.

3. **Lényeges entitások azonosítása:** a cikkek szövegeiben azonosítanunk kell a cégeket, személyeket és beosztásokat, hogy a köztük levő kapcsolatokat vizsgálhassuk.

4. **Kapcsolatok azonosítása az entitások között:** ha rendelkezésünkre áll egy adathalmaz, amely az egyes cégek, személyek, beosztások hivatkozásainak tulajdonságait, paramétereit tartalmazza, akkor elkezdhetjük vizsgálni, hogy milyen összefüggések rejlenek az egyes előfordulások, hivatkozások között, mikor mondhatjuk, hogy egy személy egy adott pozícióban egy adott cég alkalmazottja.

A részfeladatok alapján a cikkek feldolgozását az 1.1 ábra foglalja össze.



1.1. ábra. *Cikkek feldolgozásának folyamata*

A dolgozat folyamán a második, harmadik és negyedik részfeladattal foglalkozunk, amelyek az információ kivonás („*information extraction*") területére tartoznak, így a harmadik fejezetben áttekintjük a területen alkalmazott főbb technikákat, eljárásokat a nemzetközi szakirodalom alapján.

A kapcsolatazonosítás feladatának segítésére létrehozunk egy webalkalmazást, amely felhasználói felületet biztosít tanító adathalmazok létrehozásához, továbbá az azonosított

kapcsolatok megjelenítésére is képes. Azonban az alkalmazás fejlesztése, implementációja nem tárgya a dolgozatnak, pusztán felhasználói szinten kerül bemutatásra.

1.4. A dolgozat felépítése

A bevezető fejezetet követően a felhasznált adathalmazok részletes leírása következik. Ez tartalmazza a kapcsolati hálózatok feltérképezésére szolgáló cikkek jellemzését, valamint a munka során felhasznált egyéb adatforrások bemutatását.

A harmadik fejezet az eddigi, információkivonás területén alkalmazott technikákat, eljárásokat foglalja össze a nemzetközi szakirodalom alapján.

Ezt követően kerülnek sorra az elvégzett munka lépései, eredményei, tanulságai. A negyedik fejezet az adatok előzetes feldolgozásával foglalkozik, ami a későbbi munka alapvető feltételeit teremti meg. Ide tartozik az előző alfejezetben említett második és harmadik részfeladat, vagyis a releváns részletek kiszűrése a HTML oldalakból, valamint a cégek, személyek és beosztások azonosítása a szűrt adatokban.

Az ötödik és hatodik fejezetben az azonosított elemek közötti kapcsolatok felderítésével foglalkozunk. Az ötödik fejezet egy felcímékezett adathalmaz létrehozását mutatja be, amely a hatékony fejlesztéshez nyújt alapot. A felcímkézett halmaz lehetővé teszi a különböző felügyelt tanulást alkalmazó modellépítési algoritmusok futtatását, valamint a létrehozott modellek gyors tesztelését.

A hatodik fejezet a cikkekben megtalált entitások (cégek, személyek, beosztások) közötti, alkalmazotti viszonyt jelentő kapcsolatok azonosításával foglalkozik a létrehozott adathalmaz felhasználásával. A fejezet ismerteti a felépített modellalapú megoldásokat: bemutatásra kerülnek az alkalmazott modellek típusai, az alkalmazott magyarázó változók, továbbá a modellek tesztelésének eredményei is.

A hetedik fejezetben az újabb címkézett adatok előállítására, ill. az eredmények megjelenítésére létrehozott webalkalmazás felhasználói szintű ismertetése olvasható.

A dolgozat befejező részében röviden összefoglaljuk az érintett feladatokat, az elért eredményeket és a munka tanulságait, valamint áttekintjük a továbbfejlesztés lehetséges irányait.

2. fejezet

Felhaszált adathalmazok

A munka alapjául négy különböző online hírportál összes 2011-ben megjelent cikkét használtuk. A cikkek nyers HTML formában kerültek archiválásra a következő weblapokról: Index, Origo, Világgazdaság és 168 óra.

2.1. Adathalmazok cégek, személyek, tisztségviselők azonosításához

Ahogy a 1.3 fejezetben írtuk, a cégek alkalmazottainak azonosításához előbb meg kell keresnünk a cikkekben a cégneveket, személyneveket és tisztségviselői beosztásokat. Ez a feladat a nemzetközi szakirodalomban „*named entity recognition*" [18] néven ismert problémakörbe tartozik. A dolgozatban egy saját fejlesztésű, magyar nyelvű cikkekre alkalmazható megoldás kerül bemutatásra.

A cégekre való hivatkozások megtalálásához elengedhetetlen, hogy rendelkezésre álljon a cégek neveiről valamilyen lista. A weboldalak cikkei mellett egy cégnyilvántartási adathalmazt is beszereztünk, amelyben a cégek nevei és néhány egyéb információ szerepel. A letöltött, HTML formátumú cikkek, ill. ez az adattábla alkotják a munka kiindulási adatait.

A személynevek és tisztségviselői pozíciók keresésében - a cégekhez hasonlóan - szintén segítséget jelentene valamilyen lista az azonosítandó elemekről, azonban itt bármilyen külső adatforrás nélkül is lehetséges különböző eljárásokat kidolgozni a problémára. Az internetről könnyen letölthetünk egy listát a magyar keresztnevekről, amely jó kiindulási alapot nyújthat a személynevek felismeréséhez. A tisztségviselői beosztások azonosítása már nagyobb gondot okozhat, azonban a későbbiekben bemutatásra kerül egy egyszerű eljárás, amely segítségével erre a problémára is találhatunk használható megoldást külső adatforrás használata nélkül.

2.2. Cégnyilvántartási adathalmaz kiegészítése

A felhasznált cégnyilvántartási adatbázis kizárólag cégeket tartalmaz, azonban a vállalatok nemcsak más vállalatokkal, cégekkel állnak kapcsolatban. Gyakran más típusú szervezetekhez is kötődnek, mint pl.: pártokhoz. Érdekes információkat jelenthet a cégekről, hogy milyen pártokkal állnak kapcsolatban, emiatt a cégnyilvántartási adatbázist kiegészítettük pártok neveivel is.

Az adatbázis pártokkal való bővítésének egy másik előnye, hogy a pártokhoz általában rendkívül változatos tisztségviselői beosztások tartoznak, mint pl.: frakcióvezető, elnök, szóvivő, képviselő stb. Ez a sokféle, cikkekben gyakran hivatkozott beosztás nagy mennyiségű adatot szolgáltathat a későbbi munka során a tisztségviselők azonosításának feladatához.

A kiegészített cégnyilvántartási adatbázisban végül **26604** cég, ill. egyéb szervezet volt elérhető, a személynevek azonosításához letöltött keresztnévlista **560** nevet tartalmazott, az archivált cikkhalmazból pedig összesen **55000** cikket dolgoztunk fel a munka során. A cikkek a következő módon oszlottak el az egyes portálok között: Index: 13347 cikk, Origo: 15412 cikk, Világgazdaság: 18678 cikk, 168 óra: 7563 cikk.

3. fejezet

Nemzetközi szakirodalom az információ kivonásról

3.1. Mi is az az információkivonás?

Az információkivonás vagy információkinyerés alatt olyan folyamatokat értünk, melyek során szöveges dokumentumok szűrésével, a dokumentumokban megtalálható információ alapján állítunk elő struktúrált adatokat [7]. Vagyis a feladat célja, hogy természetes nyelven írt dokumentumokban található információkat struktúrált, adatbázisszerű formába rendezzünk. Az információkivonás szorosan kapcsolódik a szövegbányászat területéhez. Számos hasonló technikát, eljárást találunk a két terület között, azonban a szövegbányászati folyamatok során nincs előre specifikálva a keresendő információ típusa. Ezzel ellentétben az információkivonás során előre meghatározott típusú elemeket keresünk, mint pl.: nevek, könycímek, városok stb. [24]

A nemzetközi szakirodalomban *„information extraction"* (IE) néven hivatkoznak az információkinyerés területére, így a dolgozat során is ezt a kifejezést alkalmazzuk.

3.2. IE alkalmazások

Napjainkban óriási mennyigű adatot állítanak elő a felhasználók az interneten. Az adatok nagy részét természetes nyelven írt dokumentumok képezik, így rengeteg információ kizárólag ilyen dokumentumokban található meg. Ezen információk kiszűrése értékes adatokat jelentene egyes vállalatok számára.

Az IE iránti érdeklődés elterjedésének egyik fő oka, hogy az IE alkalmazások segítségével automatikusan végezhetjük el a természetes nyelven írt dokumentumok elemzését, értékelését, összehasonlítását, így gyorsan, nagyon nagy mennyiségű szövegből szűrhetjük ki a számunkra érdekes információkat [12]. Ezek az előnyök teszik népszerűvé az IE alkalmazásokat különböző területeken, melyek közül az alábbiakban említünk meg néhányat:

- **Ügyfélszolgálati alkalmazások**: minden cég számára rendkívül fontosak az ügyfelek véleményei, visszajelzései. A visszajelzések automatikus elemzéséhez, értékeléséhez számos IE probléma kapcsolódik, mint pl.: terméknevek, termékjellemzők azono-

sítása ügyfelek által küldött emailekben, ügyfelektől érkező emailek hozzákapcsolása egy adatbázisbeli eladási tranzakcióhoz [6], ügyfél-hangulat kiszűrése ügyfélszolgálati telefonbeszélgetések átiratából [16] stb.

- **Alkalmazás molekuláris biológiában**: több tanulmány foglalkozik az IE technikák molekuláris biológiában való alkalmazásával [15, 5]. Ezen kutatások fő célja különböző enzimek és metabolikus utak, valamint proteinstruktúrák azonosítása tudományos publikációk alapján, majd a kinyert információkból egy tudományos adatbázis létrehozása.

- **Ár-összehasonlító oldalak**: az ár-összehasonlító weboldalak különböző áruházak termékárait hasonlítják össze. A szükséges információk összegyűjtésének egy lehetséges módja az egyes áruhazak weboldalainak bejárása, majd megfelelő IE eljárásokkal a terméknevek és árak azonosítása a weboldalak kódjában [8].

- **Hirdetéselhelyezés weboldalakon**: az IE módszereket hirdetések pozícionálására is alkalmazhatjuk. A hirdetések feltehetőleg több vásárlót győznek meg, ha olyan szövegrészlet mellett helyezkednek el, ami a reklámozott termékről szól, és pozitív véleményt formáz róla. Mind a termék-, mind a véleményazonosítás az IE feladatok egy-egy példája [4].

3.3. Az előállított információ típusa

Az előállított információ típusa alapján az IE alkalmazásokat négy tágabb csoportra oszthatjuk [22]:

- **Entitások**: valamilyen típusú, kategóriájú entitásokat keresünk. Gyakori példák a nevek, termékek, földrajzi helyek, szervezetek keresése [13, 18].

- **Kapcsolatok**: kapcsolatokat általában a korábban azonosított entitások között keresünk. A kapcsolatok típusa a legtöbb esetben előre definiált. Egy ilyen kapcsolattípus lehet pl. két cég között a felvásárlási reláció, amely azt jelzi, hogy az egyik cég felvásárolta a másikat.

- **Entitások jellemzői**: bizonyos alkalmazások esetén az egyes entitásokra vonatkozó jellemzőket, tulajdonságokat kell felderíteni, melyekre az entitás szövegkörnyezetének elemzéséből lehet következtetni. Ide tartoznak a véleményfelderítő („opinion mining”) alkalmazások, amelyek egy szövegrészletet jellemeznek pozitívként vagy negatívként a bennük megfogalmazott vélemény alapján [19].

- **Struktúrák**: az előző példák kiterjesztésével rengeteg újabb eredménytípust lehet definiálni, amelyek információkivonási folyamatok eredményeként előállhatnak [20, 9]. Ezek közé tartoznak pl.: listák, ontológiák, táblázatok stb.

3.4. IE eljárások

A különböző IE technikákat kétféle módon csoportosítjuk. Az információkivonás implementációja alapján megkülönböztetünk manuálisan kódolt, ill. modellalapú eljárásokat, míg az információkivonás során alkalmazott következtetések típusa szerint beszélhetünk szabályalapú, ill. statisztikai módszerekről [22]. A továbbiakban ezt a négy osztályt jellemezzük röviden.

3.4.1. IE eljárások implementációja

Manuálisan kódolt eljárások

Manuálisan kódolt eljárások esetén sajátkezűleg implementáljuk a következtetési lépéseket a szükséges szabályok, feltételek figyelembe vételével. Nem egy előre létrehozott eljárást, modellt hangolunk az aktuális problémára, hanem egy újat hozunk létre a feladat sajátosságainak kihasználásával. Az eljárások létrehozásához rendelkeznünk kell a megfelelő domain-specifikus tudással, vagy más erőforrásokkal, amelyek alapján meg tudjuk határozni a programunk működését.

Modellalapú eljárások

A modellalapú vagy tanulásalapú módszerek [3] esetén modellek segítségével nyerjük ki az információt a nyers szövegekből. A modellek létrehozásához szükséges valamilyen manuálisan felcímkézett mintahalmaz, amely alapján elvégezhetjük a modellek tanítását. A címkézés végrehajtásához sok esetben szintén domain-specifikus tudásra van szükség. A manuálisan kódolt és a modellalapú eljárások közül mindig az aktuális feladat jellemzői, feltételei és a rendelkezésre álló erőforrások alapján kell kiválasztani a megfelelőt.

3.4.2. IE eljárások során alkalmazott következtetések típusai

Szabályalapú következtetések

Szabályalapú következtetési módszerek [23, 1] esetén a lehetséges nyelvi összefüggések, szabályok előre rögzítettek. A szabályok általában két részből állnak: egy mintából, amely leírja, hogy mikor lép érvénybe az adott szabály, ill. egy utasítássorozatból, amelyet végre kell hajtani, ha a szöveg aktuális része illeszkedik a szabály mintájára. További fontos elem lehet a szabályok precedenciájának megadása: több szabály érvényre jutása esetén hogyan kezeljük a konfliktust, melyik szabály érvényesüljön. A szabályokat megadhatjuk manuálisan, vagy tanulhatjuk minták alapján.

A szabályalapú modellek implementációja viszonylag egyszerű, azonban meglehetősen rugalmatlanok, mivel csak az előre megadott szabályok alapján képesek következtetni, nem képesek újszerű összefüggéseket felderíteni.

Statisztikai következtetések

Statisztikai következtetések [3, 11] alkalmazása során különböző magyarázó változók, attribútumok alapján vonunk le következtetések a felderítendő információra vonatkozóan. A magyarázó változókkal az aktuálisan vizsgált szövegelemet, szövegrészletet, ill. annak környezetét jellemezzük. Olyan jellemzőket kell választani, amelyek a minták nagy részében feltehetőleg jól elkülönítik a számunkra érdekes eseteket. Vagyis a magyarázó változók megválasztása kulcskérdés, hiszen akár modellalapú, akár kézzel kódolt eljárást alkalmazunk, rossz magyarázó változók esetén az eljárás nem lesz képes releváns információkat szolgáltatni a keresett adatokról.

A statisztikai következtetési módszerek képesek újszerű, rejtett összefüggések felismerésére, így általában robosztusabbak a szabályalapúaknál, azonban az implementálásuk, fejlesztésük bonyolultabb, időigényesebb feladat.

3.4.3. A dolgozatban alkalmazott eljárások

A fenti csoportosítások mentén az 1.3 fejezetben említett IE feladatokra bemutatott megoldásokat a következő módon osztályozhatjuk. A HTML oldalak forrádkódjában a lényeges tartalmak megkeresésére, továbbá a cikkekben a cégek, személyek, beosztások azonosítására manuálisan kódolt szabályalapú megoldásokat mutatunk be, míg a megtalált entitások közötti kapcsolatok azonosítására egy statisztikai modellalapú eljárás kerül kidolgozásra.

4. fejezet

Cégek, személyek, tisztségviselők azonosítása hírportálok cikkeiben

4.1. Cikkek szűrése weboldalakból

Ahogy az 1.3 fejezetben említettük, a kapcsolati hálózatok feltérképezését nyers HTML formában elmentett online hírportálok cikkei alapján végezzük. Első lépésként ezekben a weboldalakban kell megtalálnunk a cikkek szövegét, címét és megjelenésének dátumát.

A weboldalak kódjában a cikkek szövegének megkeresésének egyik lehetősége, ha statisztikai módszert alkalmazunk, és a forráskód egyes részeire különböző jellemzőket határozunk meg, majd ezen jellemzők alapján döntjük el, hogy az adott rész számunkra fontos-e. A jellemzők közé tartozhatnak olyan mennyiségek, mint pl. speciális karakterek gyakorisága, forráskód sorainak karakterszáma stb. A módszer hátránya, hogy a jellemzők alapján általában nem lehetséges mindig pontosan azonosítani az adott kódrészletet. Emellett minden típusú keresett elemhez más és más jellemzőhalmazt kell előállítani – hiszen pl. egy dátumot más tulajdonságok jellemeznek, mint egy hosszabb szöveget. Újabb hátrányt jelent, hogy a cikkek szövege mellett más természetes nyelvű szövegek is lehetnek egy weboldalon, mint pl. szöveges hirdetések. Ez az eset különösen rossz hatással lehet a cikkek elemzésének eredményeire, ha a hirdetés szövege olyan elemeket is tartalmaz, amelyeket a cikkekben keresünk, mivel a hirdetés szemantikailag nem kapcsolódik a cikkhez.

A felsorolt hátrányok miatt az előző módszer helyett egy szabályalapú megközelítést alkalmazunk, és a HTML oldalak struktúrájának vizsgálatával szűrjük ki az érdekes tartalmakat. Egy-egy hírportál cikkeinek weblapjai általában egyféle struktúrát követnek. A cikkek megtekintésekor ez a struktúra töltődik fel dinamikusan az adott cikk tartalmával. Ha megtaláljuk azokat a HTML-elemeket, amelyek a számunkra fontos adatokat tartalmazzák, akkor a forráskódból elegendő ezen elemek tartalmát kikeresnünk. Ezek az elemek többnyire egy adott típusú HTML-taget jelentenek egy meghatározott azonosítóval. Az eljárás előnye, hogy ugyanazzal a módszerrel a hosszabb természetes nyelven írt cikkszövegek mellett tetszőleges más elemeket is megtalálhatunk. Így a cikkek szövegéhez hasonlóan kereshetjük meg a megjelenés dátumát, ill. a cikk címét is. Azonban hátránya a módszernek, hogy a weboldalak struktúrája csak egy adott hírportál cikkei esetén azonos, így az

egyes hírportálok esetén külön-külön meg kell vizsgálni az oldal struktúráját, és ez alapján elvégezni a szükséges elemek keresését. A munka során négy portál cikkeit használjuk fel, így erre a négy hírportálra valósítottuk meg a struktúra alapján történő adatkinyerést.

A szűrés végrehajtásával minden cikk esetén rendelkezésre áll a cikk címe, megjelenésének dátuma és a cikk szövege.

4.2. Cégek, szervezetek azonosítása cikkekben

Mivel a cégek, szervezetek alkalmazottait szeretnénk meghatározni a cikkek alapján, ezért mindenképp azonosítani kell a cégek, szervezetek említéseit, hivatkozásait a cikkekben. Az azonosítást a 2.1 fejezetben említett cégnyilvántartási adatbázis alapján végezzük el, amely a Magyarországon bejegyzett cégekről tartalmaz néhány adatot. Ezek közül a cégek nevét, ill. „rövid nevét" használjuk fel a hivatkozások keresésekor. Vállalatok, személyek, ill. egyéb elemek azonosítására léteznek statisztikai modellalapú módszerek [21], azonban ezekben az esetekben a modellek felépítéséhez mindig rendelkezésre áll egy címkézett adathalmaz, vagyis olyan szövegek, amelyekben kézzel be vannak jelölve a keresendő elemek. Esetünkben nem áll rendelkezésre hasonló címkézett adathalmaz, így szabályalapú, manuálisan kódolt eljárást kell kidolgoznunk, amelyben a cikkek szavait közvetlenül a cégek, szervezetek neveivel vetjük össze és bizonyos szabályok, feltételek alapján döntjük el, hogy egy adott szövegrészlet hivatkozást jelent-e valamely cégre vagy szervezetre.

Nehézségek

A cégek azonosításában a fő nehézséget az jelenti, hogy szinte egyetlen esetben sem a cég teljes nevével hivatkoznak egy cégre, hanem valamilyen rövidített formát használnak. Ezek a rövidített formák nagyon változatosak lehetnek akár egyetlen cég vagy szervezet esetén is. Sok cég, szervezet, párt esetén már egyetlen szó is nyilvánvalóvá teszi a cégre való hivatkozást. Ilyen vállalatok pl.: BKV, MÁV, Erste stb. Azonban más cégek, szervezetek esetén hiába találunk a cégnév első szavával megegyező szót egy cikkben, akkor sem állíthatjuk, hogy az egy adott cégre való hivatkozást jelentene. Pl. a cégnyilvántartásban szerepel egy vállalat, amelynek neve: *Antal és Társa Fuvarozó és Kereskedelmi Kft.* Ha egy cikkben megtaláljuk azt a szót, hogy *Antal*, akkor nyilvánvalóan nem állíthatjuk, hogy a cikk erre a vállalatra hivatkozik. Általában a nagyobb és ismertebb, vagyis a gyakrabban említett szervezetek, cégek esetén találkozhatunk egyszavas hivatkozásokkal. Ezért a szervezetekre, cégekre való hivatkozások megtalálásához először kiegészítettük a cégnyilvántartási adatbázist rövidített, egyszavas nevekkel, amelyek már önmagukban hivatkozást jelentenek egy adott szervezetre.

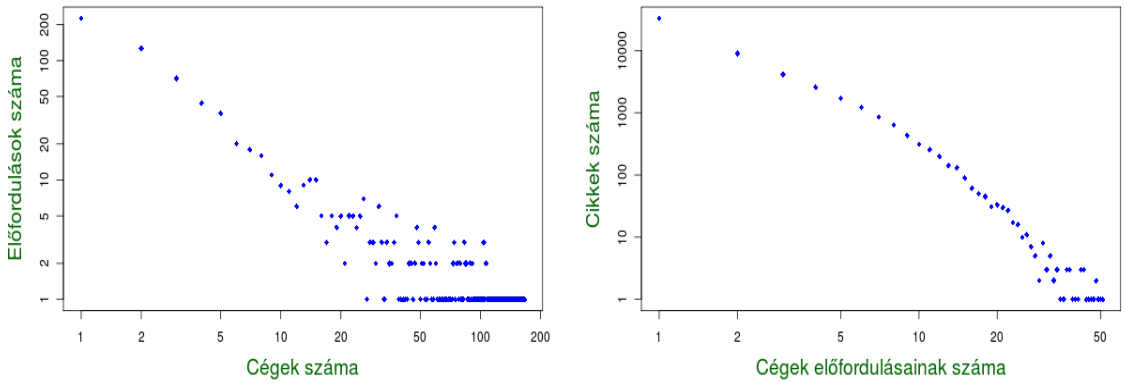
A rövidített nevek megtalálásához egy egyszerű algoritmust alkalmaztunk: ha egy cikkben egy nagybetűs szó megegyezik egy cég vagy szervezet nevének első szavával, akkor a szó-cég párt feljegyeztük. Sok cikket feldolgozva számos hivatkozást gyűjtöttünk. Ezek közül eldobásra került az összes olyan pár, amelyek előfordulási gyakorisága nem ért el egy küszöbértéket, így csak a legtöbbször előforduló párok maradtak meg, ami kb. 2000 név-cég párost jelentett. Ez a mennyiség már viszonylag gyorsan átnézhető manuálisan is.

A végső kézi szűrés után 142 rövidített név maradt az adathalmazban. Ezek közé tartoznak például a következők: Aegon, Auchan, Bosch, CBA, Citibank, Erste, FHB stb.

A létrehozott eljárás

A cégnyilvántartási adatbázist kiegészítettük a rövidített nevekkel, majd a bővített adatbázis felhasználásával hoztunk létre egy céghivatkozásokat kereső eljárást. Az eljárás az aktuális cikket tokenekre darabolja. Minden szó egy-egy token, de az írásjelek, szóközök is tokeneket alkotnak. A tokeneken végighaladva, ha egy nagybetűs szót találunk, akkor az algoritmus megvizsgálja, hogy itt vállalatot, szervezetet jelöl-e a cikk. A szóhoz hozzákapcsolja az utána következő nagybetűs szavakat, majd megvizsgálja, hogy milyen hosszan egyeznek meg a szavak a céges adatbázis cégneveivel. Ha a megtalált nagybetűs szót nem követi más nagybetűs szó a cikkben, akkor abban az esetben tekinti az algoritmus céges hivatkozásnak a szót, ha az megegyezik valamelyik cég, szervezet rövidített nevével, vagy megegyezik egy olyan cég nevének első szavával, amely cégre már találtunk korábban hivatkozást a cikkben. Ha több nagybetűs szó következik egymás után, akkor az utolsó szó kezdőbetűéig meg kell egyeznie a szövegbeli szavaknak egy vállalat nevével. Az így kiválasztott vállalatok közül a leghosszabb egyezéssel rendelkező vállalatot választja ki az algoritmus. Ha leghosszabb egyezés két különböző vállalat esetén is ugyanakkora volt, akkor az algoritmus nem tekinti hivatkozásnak a szövegrészletet.

Az egyes cégekre, szervezetekre vonatkozó hivatkozások számának eloszlását a 4.1a ábra mutatja. Az eloszlásokat bemutató ábrákon logaritmikus skálákat alkalmaztunk. Látható, hogy nehézfarkú eloszlást követ a cégek megjelenéseinek száma: egy-két cégre, szervezetre igen gyakran hivatkoznak, azonban a többségre egyáltalán nem, vagy csak nagyon ritkán. A 4.1b ábra a cégek előfordulásainak cikkek szerinti eloszlását mutatja. Ismét nehézfarkú eloszlást figyelhetünk meg: a cikkek nagy részében egyáltalán nem találhatunk cégekre, szervezetekre való hivatkozást, azonban néhány cikkben sok vállalat is előfordul.



(a) *Céghivatkozások számának eloszlása* **(b)** *Céghivatkozások eloszlása cikkek szerint*

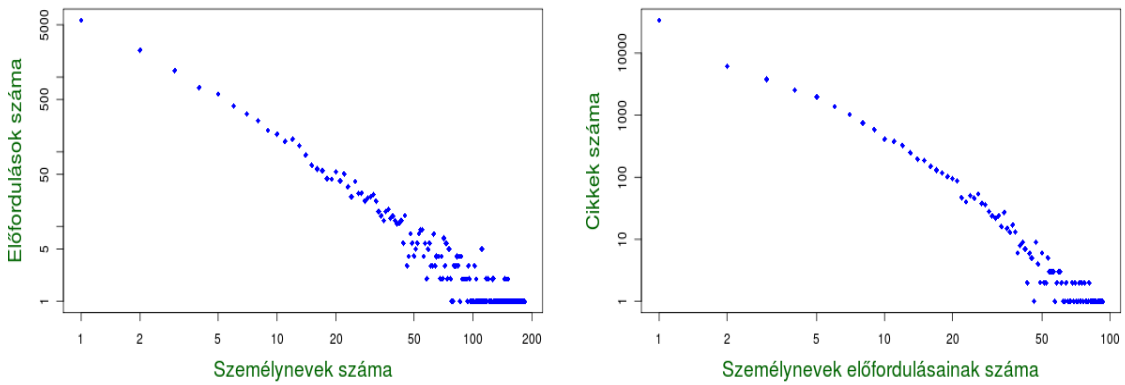
4.1. ábra. *Céghivatkozások eloszlása*

4.3. Személyek azonosítása cikkekben

A személynevek azonosításához egy Wikipédiáról letöltött magyar keresztnévlistát használtunk fel. A személynevek keresésében a következő egyszerűsítéseket tesszük: a személynevek mindig egy vezetéknévből és egy keresztnévből állnak, csak toldalék nélküli alakokat veszünk figyelembe, valamint feltételezzük, hogy a cikkekben egy személyre vonatkozó hivatkozások közül legalább az első esetben kiírják az adott személy teljes nevét, későbbi azonban lehetséges, hogy csak a vezetéknévvel hivatkoznak az adott személyre. Továbbá nem teszünk különbséget az azonos nevű emberek között, vagyis ha két különböző cikkben találunk egy-egy azonos vezetéknév-keresztnév párt, akkor a két hivatkozást minden esetben egyetlen személyhez rendeljük.

Ezek alapján a személynevek keresése az alábbi módon történik: a cikket tokenekre daraboljuk, majd nagybetűs szavakat keresünk. Ha egy nagybetűs szó megegyezik egy keresztnévvel, és előtte szintén egy nagybetűs szó található, akkor a két szót egy vezetéknév-keresztnév párosnak tekintjük, és az adott névhez hozzárendeljük a megtalált hivatkozást. Ezen kívül minden olyan nagybetűs szót hivatkozásnak tekintünk, amely megegyezik egy, a cikkben már korábban megtalált vezetéknévvel. Ehhez hasonló megoldással a szakirodalomban is találkozhatunk [10].

A 4.2a ábra mutatja az egyes személyekre vonatkozó hivatkozások számának eloszlását, míg a 4.2b ábra a személynevek cikkek szerinti eloszlását ábrázolja. Mindkét ábra nehézfarkú eloszlást követ, hasonlóan a cégek előfordulásainak eloszlásához.



(a) *Személynevek hivatkozásainak eloszlása* **(b)** *Személynevek eloszlása cikkekben*

4.2. ábra. *Személynevek eloszlása*

4.4. Tisztségviselők azonosítása cikkekben

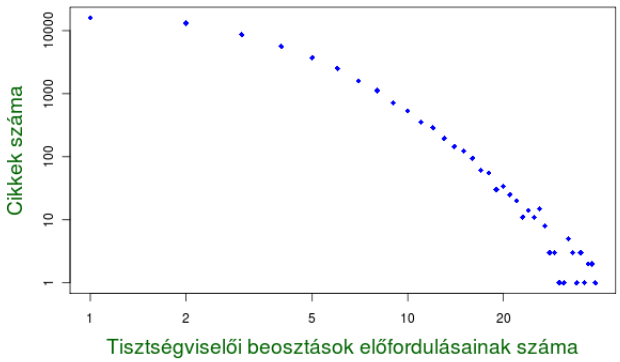
A tisztségviselői beosztások azonosításához szintén szükséges valamilyen lista a beosztások, pozíciók, titulusok neveiről. A személynevekkel ellentétben ilyen listát nem tudunk egyszerűen találni az interneten, ezért létre kell hoznunk egyet. Néhány cikkben megvizsgálva a beosztások megjelenését gyorsan feltűnik, hogy rengeteg esetben egy szervezet, vállalat neve után szerepelnek a beosztások birtokos személyjellel. Pl.:

„...a Raiffeisen Bank elemzője...”,
„...az OMSZ munkatársa...”,
„...a Magyar Nemzeti Bank elnöke...”

Ezt a megfigyelést felhasználva a cégneveket azonosító algoritmus segítségével generáltunk egy listát, amely a cikkekben megtalált cégnevek után következő, birtokos személyjellel rendelkező szavakat tartalmazta. Ezzel a lépéssel azt a feltételezést tettük, hogy a beosztások nevei egyetlen szóból állnak. A létrehozott listából kiszűrve a duplikációkat kb. 2000 szó maradt, amelyeket kézzel átnézve szűrtük ki a tisztségviselői beosztásokat. A végső listába 299 beosztás került toldalék nélküli alakban.

Ezek után a cikkek szavainak szótövezett alakját összehasonlítva a létrehozott lista elemeivel megkereshetjük a tisztségviselői beosztásokat a cikkekben. A szavak szótövezésére egy belső fejlesztésű szótövező programot használtunk fel.

A tisztségviselői beosztások cikkek szerinti eloszlását a 4.3 ábra mutatja. Hasonlóan a cégek és személynevek eloszlásához a beosztások előfordulásai is nehézfarkú eloszlást követnek.



4.3. ábra. *Tisztségviselői beosztások eloszlása cikkek szerint*

4.5. Néhány jellemző az adatokról

Kb. 55000 cikkben végeztük el a cégek, személyek és tisztségviselői beosztások azonosítását. A felhasznált cégnyilvántartási adatbázisban több, mint 30000 cég szerepel, így a cégek azonosítása ilyen nagy mennyiségű cikkben igen időigényes feladat. Emiatt célszerű a megtalált hivatkozásokat valamilyen formában elmenteni, és a későbbi munka során ebből az elmentett állományból betölteni a megtalált előfordulások adatait. A cégek hivatkozásai mellett a személynevek és tisztségviselői beosztások előfordulásait is elmentettük. Minden elmentett szövegelemhez rögzítésre került a cikk azonosítója, a megtalált entitás – cég, személy vagy beosztás – azonosítója, a megtalált entitás típusa (cég, személy vagy tisztségviselői beosztás), valamint a hivatkozás szövegbeli helyét leíró indexek. Az indexek lehetővé teszik, hogy a későbbi munka során egy adott hivatkozás cikkbeli szövegkörnyezetét is gyorsan megtaláljuk, ami alapján további jellemzőket határozhatunk meg a

hivatkozásról.

Az egyes híportálok cikkeiben előforduló cégek, személynevek, beosztások átlagos számát a 4.1 táblázat mutatja. Az azonosított entitások gyakoriságában nincs számottevő különbség az egyes portálok között.

	Cégek száma cikkenként	**Személynevek száma cikkenként**	**Beosztások száma cikkenként**
Index	1.26	1.74	2.13
Origo	1.22	2.67	2.43
Világgazdaság	1.38	1.07	2.21
168 óra	1.30	2.69	2.06

4.1. táblázat. *Cégek, személyek, beosztások előfordulásainak átlagos száma cikkenként*

5. fejezet

Egy adathalmaz előállítása

5.1. Adatreprezentáció és célváltozó meghatározása

A munka eddigi lépései során elértük, hogy a hírportálok cikkeiben rendelkezésre állnak a számunkra érdekes elemek előfordulásainak pozíciói. Ezek alapján elkezdhetjük vizsgálni, hogy milyen kapcsolatokat lehet ezek között az elemek között észrevenni, feltérképezni. A célunk, hogy megtaláljuk az egyes cégek tisztségviselőinek nevét, vagyis olyan hármasokat keresünk, amelyeknek eleme egy cég, egy személynév és egy tisztségviselői beosztás.

Mindenképp jó lenne, ha a feladatra a klasszikus adatbányászati módszerek alkalmazásával modellalapú megoldásokat tudnánk találni. Ahhoz, hogy modelleket építhessünk, olyan adatreprezentáció szükséges, amelyben esetek, minták szerepelnek és minden mintához tartozik egy célváltozó, amit a modell segítségével szeretnénk meghatározni. Az egyes minták a célváltozó mellett egyéb attribútumokat, magyarázó változókat is tartalmaznak. A felépített modellek a magyarázó változók és a célváltozó közötti összefüggések alapján következtetnek a célváltozó értékére.

Az előzőek alapján egy olyan adatreprezentációt választottunk, amelyben minden minta egy-egy cég-személy-beosztás hármasnak felel meg. A célváltozó egy bináris változó, amely azt jelzi, hogy a mintában azonosított hármas összekapcsolható-e. Azaz a célváltozó akkor vesz fel *igaz* értéket, ha a három hivatkozás a cikkben azt fejezi ki, hogy az adott személy valóban a megadott cég megadott tisztségviselője, egyébként a célváltozó értéke *hamis*. A reprezentáció mintáit, egyedeit az egy cikkben előforduló céghivatkozások, személynevek és tisztségviselői beosztások lehetséges kombinációi alkotják. Természetesen csak az egy cikken belüli előfordulások kombinációi kerülnek a minták közé, hiszen a külön cikkekben szereplő hivatkozások között nem lenne értelme ilyen összefüggéseket keresni.

Mivel a létrehozott célváltozó két értéket vehet fel, ezért a feladatunk egy bináris osztályozási probléma [14].

5.2. Adatok címkézése, tapasztalatok

A minták osztályozására alkalmas modellek építése felügyelt tanító eljárások segítségével történik. Ahhoz, hogy ilyen eljárásokat futtathassunk, szükséges ismernünk a célváltozó értékét a minták egy részén, azaz szükségünk van egy korpuszra. A korpusz alapján felépít-

hetjük modelleinket, amelyeket a későbbi, ismeretlen célváltozójú eseteken alkalmazunk.

A cikkekben előforduló cég-személy-beosztás hármasok közül feltehetőleg azokat lehet nagy valószínűséggel összekötni, amelyek egymáshoz viszonylag közel helyezkednek el a szövegben. Ezért kizárólag olyan mintákat címkéztünk fel, ahol a három hivatkozás egymástól vett távolsága egy bizonyos küszöbérték alatt volt. A későbbiekben is az itt alkalmazott küszöbértéket használjuk a minták szűrésére, így ez az érték nem változik az elemzések folyamán. A hivatkozások távolságának mérésére a három hivatkozás között a cikk szövegében megtalálható tokenek számát alkalmaztuk.

A címkézést egy egyszerű program segítségével végeztük el: a hivatkozások szövegbeli pozíciója alapján megjelenítettük a cikkek megfelelő részét, megjelölve a kérdéses cég, személy és beosztás pozícióját, majd két lehetséges érték közül egyet választva megadtuk a célváltozó értékét. Ezzel a módszerrel viszonylag gyorsan lehet mintákat címkézni. A program felületét az alábbi három példa, ill. a hozzájuk megadott válaszok mutatják (a kimenetben vastag betűk jelzik az aktuális céget, személyt és tisztségviselői beosztást):

> *„...A* **Generali-Providencia Biztosítót** *16 éve* **vezető**, *57 éves* **Pálvölgyi Mátyás** *2011. június...”*
> (F:false/T:true)
> T
> *„...és Kerék-Bárczi Szabolcs, az* **MDF** *korábbi* **szóvivője** *is.* **Gyurcsány** *a rendezvény elején...”*
> (F:false/T:true)
> F
> *„...fontosnak tartja.* **Szigeti Károly**, *a* **KÉSZ Kft.** *stratégiai* **igazgatója** *Egyeztetett elképzelések...”*
> (F:false/T:true)
> T

5.3. Adathalmazok

A címkézés eredményeként kb. 1100 cég-személy-beosztás hármas esetén került meghatározásra a célváltozó értéke. Ezeket a mintákat két halmazra osztjuk:

- A minták kb. 85%-ából egy tanuló adathalmazt hozunk létre. Ez az adathalmaz szolgál a felügyelt tanuló folyamatok futtatására, modellek építésére, valamint a modellek első értékelésére. Mivel a mintaszám viszonylag alacsony, ezért a modellek építése és tesztelése során keresztvalidációt alkalmazunk, így a modell teljesítményéről stabilabb, megbízhatóbb információt kapunk. A keresztvalidáció során valójában ezt az adathalmazt is több részre osztjuk, vagyis a modellek tanítása és tesztelése mindig diszjunkt halmazokon történik.

- A minták hátramaradó részéből, kb. 15%-ából áll a teszthalmaz, amely kizárólag a létrehozott modellek végső tesztelését, értékelését szolgálja.

Az adathalmazok szétválasztásánál - a keresztvalódiciós folyamatot is beleértve - ügyelnünk kell arra, hogy az egy cikkhez tartozó hármasok mindig ugyanabba a halmazba kerüljenek. Ha ez nem teljesülne, akkor előfordulnának olyan szövegbeli hivatkozások, amelyek – más és más minta részeként, de – mind a teszt, mind a tanítóhalmazban szerepelnek (és pusztán a hármasok többi tagja különbözne ezekben a mintákban). Ekkor ezen hivatkozások jellemzői mindkét halmazban szerepelnének, ami könnyen vezethetne túltanuláshoz, hamis szabályok rögzítéséhez.

Az adathalmazok szétválasztására a túltanulás elkerülése miatt van szükség, hiszen egy tanuló algoritmus eredménye alapján működő modell nyilvánvalóan jól teljesít azon az adathalmazon, amelyen tanítottuk. Azonban egy másik, független halmazon végzett tesztelés feltehetőleg már releváns mutatókat szolgáltat a modell teljesítményéről.

Ahhoz, hogy egy általános képet kapjunk az adatok karakterisztikájáról, vizsgáljuk meg a célváltozó eloszlását. A változó halmazonkénti eloszlását a 5.1 táblázat foglalja össze.

	Hamis esetek száma	**Igaz esetek száma**
Tanítóhalmaz	267	676
Teszthalmaz	49	101

5.1. táblázat. *A célváltozó eloszlása az adathalmazokon*

Láthatjuk, hogy a teszt- és a tanítóhalmazban is viszonylag nagy arányban találunk *igaz* címkéket, amelynek főként az lehet az oka, hogy olyan hármasokat címkéztünk, amelyek a szövegekben egymáshoz közeli hivatkozásokra vonatkoznak.

6. fejezet

Modellalapú megoldások entitások közötti kapcsolatok azonosítására

6.1. A modellek teljesítménymértéke

Miután összeállítottuk a felcímkézett adathalmazokat, elkezdhetünk modelleket építeni a célváltozó meghatározásához. A különböző modellek pontosságának, hatékonyságának összehasonlítására szükségünk van valamilyen teljesítménymértékre. Ahogy az 5.1 fejezetben írtuk, a célváltozónak két lehetséges értéke van, így a feladatunk egy bináris osztályozási feladat, ezért a modellek teljesítményének jellemzésére használhatjuk a ROC (Receiver Operating Characteristic) görbét, amellyel nemcsak a modell által meghatározott osztályokat vesszük figyelembe az értékeléskor, hanem az osztályozások konfidenciáját is. Az értékelés során elsősorban a görbe alatti terület nagyságát vesszük figyelembe, azonban a ROC görbe segítséget nyújt abban is, hogy miként határozzuk meg egy modell esetében az osztályozások konfidenciájának küszöbértékét, annak függvényében, hogy az első- vagy másodfajú hibák számát akarjuk minimalizálni.

6.2. Alkalmazott modelltípusok

Egy adott modellezési feladatban különböző modelltípusok igen eltérő eredményt érhetnek el, hiszen minden modelltípusnak vannak előnyei és hátrányai. Emiatt érdemes többféle modelltípust is kipróbálni egy probléma esetén, és az elért eredmények alapján kiválasztani az adott feladatra legalkalmasabb modellt. A modellezési vizsgálatok során az alábbi három modelltípust alkalmazzuk:

- **LogitBoost:** a logitboost eljárás egy olyan boosting eljárás, amely során sok, egyetlen csomópontból álló döntési fa kimenetének az átlaga adja a modell kimenetét. A tanítás egy többiterációs folyamat, melyben az egyes mintákat különböző súllyal vesszük figyelembe. Az iterációk során a minták súlyai folyamatosan változnak: a rosszul osztályozott minták súlya nő, míg a jól osztályozott mintáké csökken. A logitboost algoritmus a logisztikus veszteségfüggvény értékét minimalizálja. A modellépítéshez az R program *caTools* csomagját [25] alkalmazzuk.

- **Random forest:** a random forest eljárás során döntési fákat hozunk létre. A modell végső kimenete (osztályozás esetén) az egyes döntési fák kimenetének módusza lesz. Az eljárás segítségével gyakran pontosabb, túltanulásra kevésbé érzékeny, robosztusabb modelleket kaphatunk, mint ha csak egy döntési fát építenénk. A random forest modellek építéséhez az R program *randomForest* csomagját [17] alkalmazzuk.

- **SVM (Support Vector Machine):** egy SVM modell egy transzformáció segítségével a mintákhoz egy-egy pontot rendel a térben, majd a különböző osztályokhoz tartozó pontokat elválasztó hipersíkot keres, úgy hogy az egyes osztályok pontjainak a síktól való minimális távolságát maximalizálja. Az ismeretlen mintákat a tanulás során alkalmazott transzformációval szintén a tér egy pontjához rendeli a modell, majd az alapján sorolja a pontot egy osztályba, hogy az elválasztó sík melyik oldalára esik. Az SVM modellek képesek a változók közötti nemlináris kapcsolatokat is felismerni, amennyiben a bemeneti adatokhoz rendelt pontok meghatározásánál megfelelő kernelt használ. Az SVM modellek az úgynevezett kernel-trükk segítségével nagyon magas dimenziószámú térben is képesek jó elválasztó síkot keresni. A futás a bemenő pontok számával arányos ekkor, nem a vizsgált tér dimenziószámával, ami a mi esetünkben különösen előnyös tulajdonság. Az SVM modellek létrehozásához az R program *e1071* csomagját [2] alkalmazzuk.

6.3. Magyarázó változók előállítása

A modellek építése esetén a legfontosabb kérdés az, hogy milyen magyarázó változókat állítunk elő, hiszen azok alapján próbálja a modell meghatározni a célváltozó értékét.

Az első, kezdeti modellekhez a következő változókat használjuk:

- **Hivatkozások sorrendje:** kategorikus változó, amely hat lehetséges értéket vehet fel. Azt mutatja meg, hogy az adott minta esetén a cég-személy-beosztás hármas elemei milyen sorrendben követik egymást a cikkben.

- **Egy mondat-e:** bináris változó, amely azt jelzi, hogy a cég-személy-beosztás hármas elemei a cikkben egy mondaton belül vannak-e.

- **Egy tagmondat-e:** bináris változó, amely azt jelzi, hogy a cég-személy-beosztás hármas elemei a cikkben egy tagmondaton belül vannak-e. Természetesen, ha az előző változó értéke *hamis* volt, akkor ez a változó is *hamis*.

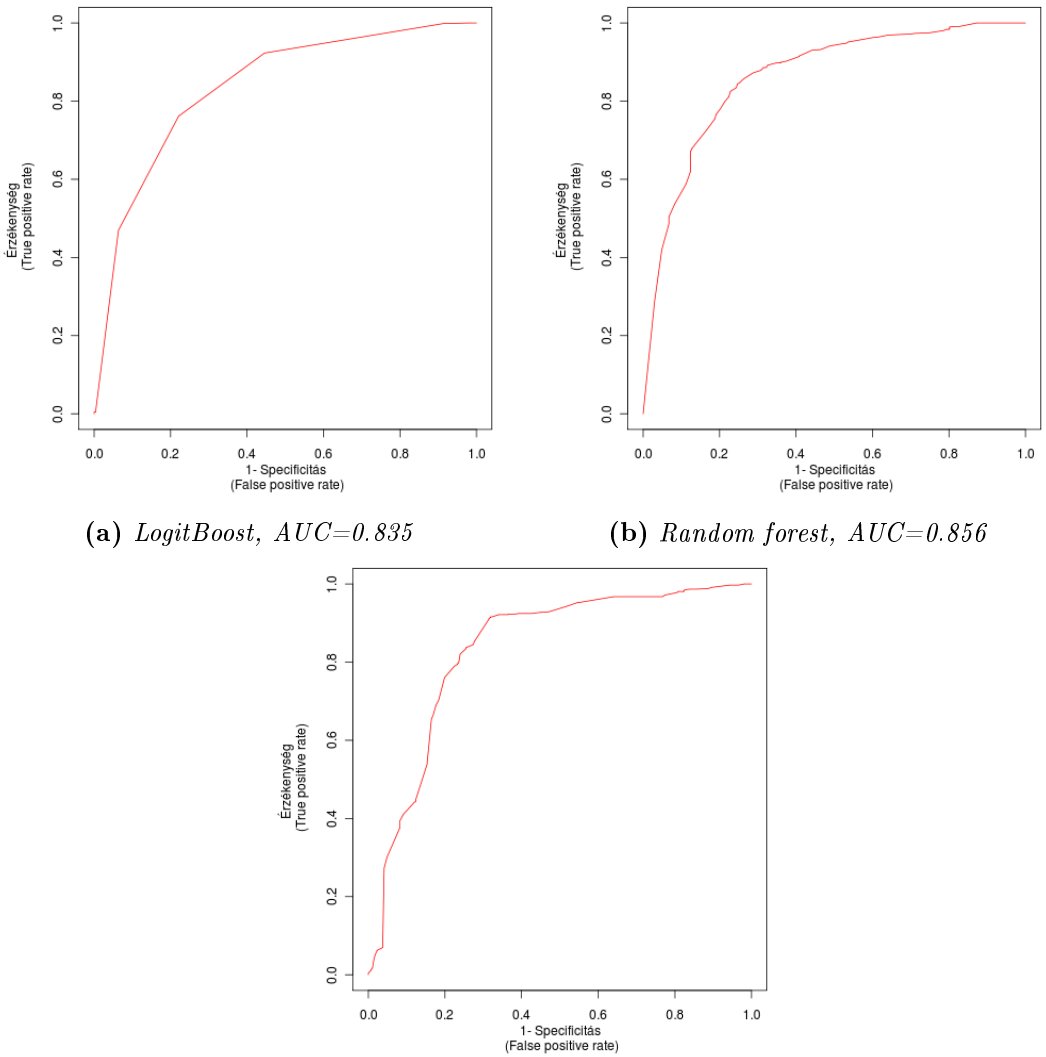
- **A személynév a cégnéven belül van-e:** szintén bináris változó. Azt mutatja meg, hogy a mintában szereplő személyhivatkozás a céghivatkozáson belül van-e a cikkben. Ez a változó azon esetek megkülönböztetésére szolgál, ahol a cég nevében személynév található, és emiatt a cikkben egyetlen szó egyaránt része egy cég- és egy személyhivatkozásnak is.

- **Első hivatkozás első szavának, és az utolsó hivatkozás utolsó szavának távolsága:** egy egész értékű változó, amely azt jelzi, hogy hány szó a terjedelme a

cikk azon részének, amely a cég-személy-beosztás hármas hivatkozásait tartalmazza. Vagyis azt mutatja meg, hogy a hármas első elemének első szava és utolsó elemének utolsó szava között hány szó található a cikkben.

6.4. Kezdeti modellek építése és értékelése

Az előállított magyarázó változók segítségével már létrehozhatunk modelleket a célváltozó meghatározásához. A modellek építéséhez, és első tesztelésükhöz a tanító adathalmazt használjuk fel. A halmazon hatszoros keresztvalidációt alkalmazunk, így minden iterációban az adatok kb. 17%-át használjuk az aktuális modell tesztelésére. Az előző fejezetben említett háromféle modelltípus az alábbi ROC görbéket eredményezte:



(a) *LogitBoost, AUC=0.835*

(b) *Random forest, AUC=0.856*

(c) *SVM, AUC=0.830*

6.1. ábra. *Kezdeti modellek ROC görbéje*

A ROC görbék alapján elmondhatjuk, hogy az összes modell viszonylag jó eredményt ért el, annak ellenére, hogy csak néhány magyarázó változót használtunk. A legjobb teljesítményt a random forest modell érte el, melynél a ROC görbe alatti terület **0.856** volt.

6.5. További magyarázó változók bevezetése

A kezdeti modellekben alkalmazott magyarázó változók az elemeknek pusztán néhány, egyszerű tulajdonságát jellemzik. Az eredmények javításához, pontosításához újabb változókat kell bevezetnünk, amelyek más szempontból jellemzik az elemeket. A modelleink ezeket felhasználva új összefüggéseket deríthetnek fel, amelyek segíthetnek az osztályozás továbbfejlesztésében.

6.5.1. Toldalékok figyelembe vétele

A magyar nyelv az agglutináló nyelvek közé tartozik, azaz a magyar nyelvben rengeteg dolgot toldalékokkal fejezünk ki, mint pl.: alanyt, igeidőt, birtokos viszonyt, többes számot stb. Emiatt érdemes lehet néhány változót bevezetni, amelyek az elemek hivatkozásain megtalálható toldalékokat jellemzik.

A címkézés során sok cégre, személyre és pozícióra vonatkozó hármast kellett megvizsgálni, így feltűnővé vált néhány gyakrabban előforduló eset, ahol a toldalékok figyelembe vétele segíthet az osztályozásban. Ezen esetek jellemzésére a következő változókat vezetjük be:

- **A tisztségviselői pozíció többes számban szerepel-e:** ha egy pozíció többes számban szerepel, akkor az több emberre vonatkozik. Ezekben az esetekben gyakran nem említik név szerint a pozíciót betöltő embereket, vagyis nem rendelhető össze az adott hármas, hiszen a név nem a pozíciót betöltő ember(eke)t jelöli. Ilyen eseteket mutatnak be az alábbi példák (a példákban a kérdéses céget, személyt és beosztást vastag betűvel emeltük ki):

  > *„ ...vélekednek az* **Erste közgazdászai**. **Hosszú Edmond**, *az MKB Bank elemzője... ”*

  > *„ ...alapvető magyar érdek.* **Orbán Viktor** *bemutatta az* **Opel vezetőinek** *a magyar felsőoktatás... ”*

  Ezen esetek jellemzésére bevezetünk egy bináris változót, amely azt jelzi, hogy a pozíciót leíró szó többes számban szerepel-e a cikkben.

- **Szerepel-e -nak, -nek toldalék a cég nevén:** rengeteg olyan esetben szerepel a cég nevén a -nak, -nek toldalék, ahol egy személy az említett cégnek nyilatkozik. Ekkor szintén nem szabad összerendelnünk a hármast, hiszen a személy nem a cég alkalmazottja. A következő cikkrészletek erre mutatnak példát:

  > *„...Az* **MTI-nek** *nyilatkozó* **Márton Levente**, *egy búvárbázis* **vezetője** *közölte...”*

„...mondta az **MTI-nek Juhász Attila**, *a Political Capital vezető* **elemzője**...*”*

- **A személynéven és a pozíción ugyanaz a toldalék szerepel-e:** sok esetben a személynév előtt vagy után magyarázatként olvasható, hogy az adott személy hol dolgozik milyen pozícióban. Ekkor ha a személynéven szerepel egy toldalék, akkor általában a pozíción is ugyanaz a toldalék szerepel. Vagyis ha a néven és a pozíción ugyanolyan toldalék található, akkor valószínűleg összerendelhető a hármas, egyébként pedig nem. Az alábbi példák ilyen eseteket mutatnak be:

  „...hozzájárulása nélkül - a **Csányi Sándorral**, *az* **OTP vezérigazgatójával** *folytatott...”*

  „...aki **Élő Gábornak**, *az* **MTI Híircentrum vezetőjének** *bizalmasa...”*

  „ ...48 óráig nem találkozhatnak az **ügyvédjükkel**. **Lázár János**, *a* **Fidesz** *frakcióvezetőjének... ”*

  A címkézés során ilyen esetekben különösen sokszor lehetett találkozni a -nak, -nek, ill. a -val, -vel toldalékokkal, ezért erre a két toldaléktípusra vezetünk be egy-egy magyarázó változót.

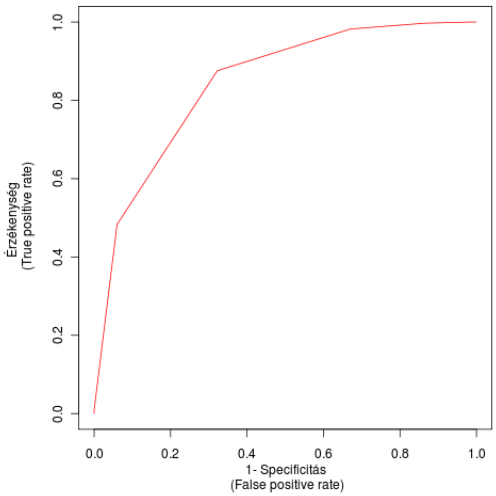
6.5.2. Más hivatkozások pozíciójának figyelembe vétele

A toldalékok mellett érdemes figyelembe venni, hogy egy elem esetén az elem három hivatkozása mellett milyen egyéb hivatkozások szerepelnek a cikkben, és azok hol helyezkednek el. Ez különösen akkor lehet hasznos, ha egy hármas egyes hivatkozásai között találhatunk egyéb cégre, személyre vagy beosztásra vonatkozó szövegelemet. Ekkor feltehetőleg nem rendelhető össze a hármas, hiszen közéjük van ékelődve egy másik hivatkozás. Az ilyen esetek megkülönböztetésre bevezetünk egy bináris változót, amely jelzi, hogy az adott minta esetén a három hivatkozás között találtunk-e egyéb cégre, személyre vagy beosztásra vonatkozó említést a cikkben.

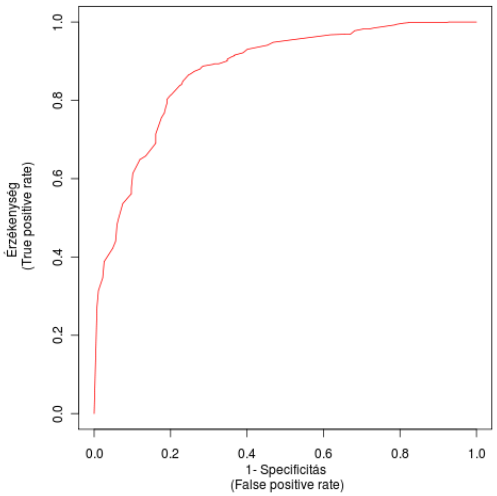
6.5.3. Eredmények

A magyarázó változók új, bővített halmazát felhasználva ismét lefuttattuk a három modellezési eljárást a tanítóhalmazon hatszoros keresztvalidációt alkalmazva. A kapott ROC görbéket a 6.2 ábra mutatja.

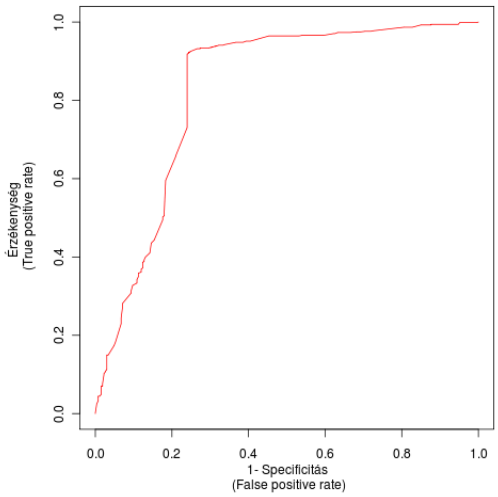
A grafikonok alapján láthatjuk, hogy az új változók létrehozásával a logitboost és a random forest modellek képesek voltak jobb teljesítményt elérni, ennek ellenére az SVM modell teljesítménye kicsit visszaesett a korábbiakhoz képest.



(a) *LogitBoost, AUC=0.846*



(b) *Random forest, AUC=0.878*



(c) *SVM, AUC=0.822*

6.2. ábra. *Bővített attribútumhalmazzal tanított modellek ROC görbéje*

A legnagyobb előrelépés a random forest modell teljesítményében következett be, amely már az első tesztek során is a legjobb teljesítményt nyújtotta. A görbe alatti területe: **0.878**.

6.6. Végső modell értékelése

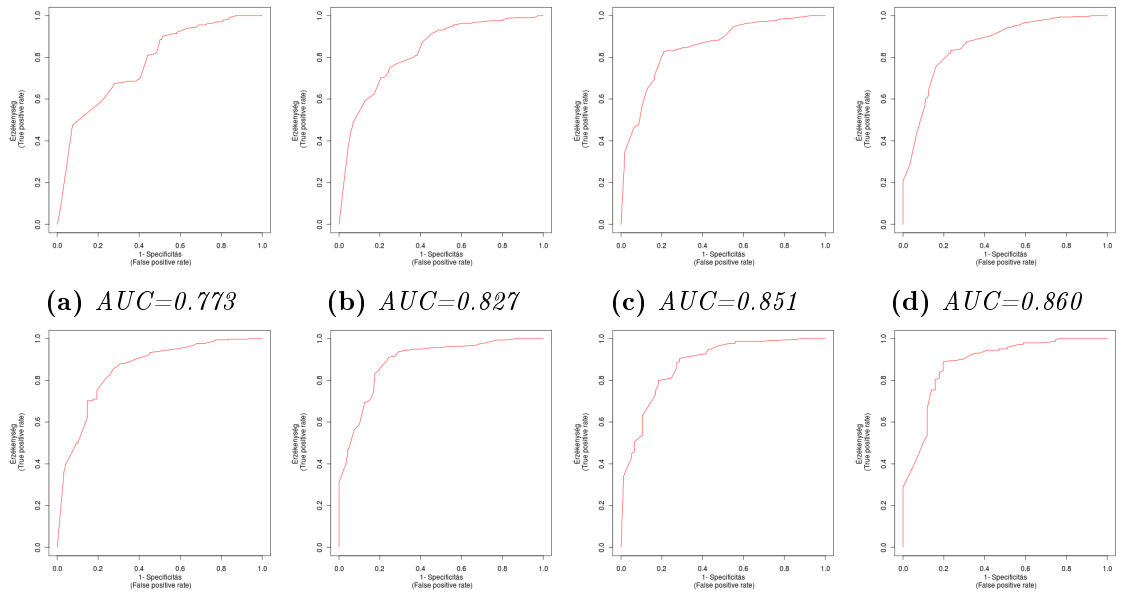
Ahogy az előző alfejezetben láthattuk, a három alkalmazott modelltípus közül a random forest teljesített a legjobban, így ezt a modellt alkalmazzuk a továbbiakban, és ennek a modellnek vizsgáljuk meg részletesebben a tulajdonságait, ill. az alkalmázásával járó következményeket.

6.6.1. További címkézés szükségessége

A címkézett adathalmaz mérete kb. 1100 elem (amelyből a tanítóhalmaz mérete kb. 950 elem.) Ez igen alacsonynak mondható, így könnyen lehet, hogy nagyobb tanító adathalmaz-

zal jóval nagyobb teljesítményt érhetnének el a modellek. Ennek vizsgálatára dinamikusan osszuk fel a tanítóhalmazunkat két részre: tanító- és teszthalmazra. Kezdetben a tanítóhalmazba kevés elemet sorolunk, így a teszthalmaz nagyobb lesz. Ezek után fokozatosan növeljük a tanítóhalmaz méretét, a teszthalmazét pedig csökkentjük. Minden felosztás esetén felépítünk egy random forest modellt, és értékeljük a teljesítményét az aktuális teszthalmazon. Azt kell megvizsgálnunk, hogy a teljesítmény hogyan változik a tanítóhalmaz méretének növelésével. Ha a teljesítmény folyamatosan növekedik, még az utolsó felosztásban is, akkor feltehetőleg érdemes lenne újabb címkézett elemeket előállítani. Azonban ha a teljesítmény növekedése megáll, vagy nagy mértékben lelassul, akkor valószínűleg elég nagy a címkézett adathalmazunk, és nem tudnánk sokkal jobb teljesítményt elérni nagyobb mintahalmaz esetén sem a jelenlegi magyarázó változókkal.

A tanító halmaz méretét kezdetben 50-nek választottuk, majd folyamatosan 100-zal növeltük, vagyis a teszthalmaz mérete 100-zal csökkent az iterációk során. Összesen 8 tanító-tesztelő ciklust hajtottunk végre. A ROC görbék változását az alábbi ábrák mutatják be:



(a) *AUC=0.773* **(b)** *AUC=0.827* **(c)** *AUC=0.851* **(d)** *AUC=0.860*

(e) *AUC=0.853* **(f)** *AUC=0.891* **(g)** *AUC=0.878* **(h)** *AUC=0.883*

6.3. ábra. *A ROC görbék változása a tanítóhalmaz méretének növelésével*

Láthatjuk, hogy a modell teljesítménye a hatodik iterációban érte el a maximumát, előtte – egyetlen kivétellel – folyamatosan nőtt a teljesítmény. A hetedik és nyolcadik iterációban a görbe alatti terület mérete kicsit visszaesett, de ez nem mondható számottevő romlásnak. Tehát elmondhatjuk, hogy a jelenlegi változóhalmaz használata mellett a címkézett adathalmaz mérete elegendő.

Azonban fontos megjegyeznünk, hogy ha a modellünket újabb változók bevezetésével szeretnénk javítani, akkor a szükséges mintahalmaz mérete könnyen megugorhat a dimenzió átka miatt [14], így ezeket a vizsgálatokat újra el kell végezni az új változóhalmaz felhasználásával.

Érdekesség, hogy már az első iterációban létrehozott modell is igen jó teljesítményt nyúj-

tott. Ennek oka, hogy a magyarázó változók egymástól függetlenül is hatékonyak tudnak lenni, már önmagukban nagy magyarázó erővel bírnak. Emiatt a random forest modell egyes döntési fái képesek a magyarázó változók és a célváltozó közötti összefüggéseket gyorsan, kis számú minta alapján felismerni.

6.6.2. Minimális konfidenciaérték meghatározása

Egy osztályozó modell alkalmazása esetén meg kell határoznunk, hogy mikor fogadjuk el a modell döntéseit. Ehhez általában a modell osztályozásának konfidenciájára vezetünk be egy küszöbértéket. A küszöbérték meghatározásakor körültekintően kell eljárnunk. Nincs általános szabály a küszöbérték megválasztására, hiszen különböző típusú feladatok esetén más és más módszer lehet célravezető.

Jelenlegi feladatunkban a célváltozó egy bináris változó, amelynek értéke akkor *igaz*, ha a cégre, személyre és pozícióra vonatkozó hivatkozás-hármast egymáshoz lehet rendelni, vagyis az adott személy az adott cég adott beosztottja. Azok az elemek hordoznak információt, ahol a célváltozó értéke *igaz*. Ha a célváltozó *hamis*, vagyis a hármas nem rendelhető össze, akkor abból semmilyen információt nem tudunk a későbbiekben felhasználni. Tehát azok az esetek érdekesek számunkra, ahol az osztályozó *igaz* címkét rendel az elemekhez. Azt szeretnénk, hogy az ilyen esetek között minél alacsonyabb legyen a hibák száma. Hiszen ha egy *hamis* esetet sorolunk az *igaz* címkéjű osztályba, akkor a későbbiekben ezt a hibás információt fogjuk felhasználni, azonban ha egy *igaz* esetet osztályozunk *hamisnak*, akkor pusztán nem használjuk fel az elemben rejlő információt, ami jóval kisebb kárt okoz, mint a helytelen információ felhasználása.

Esetünkben az *igaz* címkéjű elemek jelentik a pozitív mintákat, míg a *hamis* címkéjűek a negatívakat. Tehát a negatív elemek pozitívak közé sorolását szeretnénk minimalizálni. Az ilyen típusú hibát - ahol negatív esetet sorolunk a pozitívak közé - elsőfajú hibának (false positive errornak) nevezzük. Az elsőfajú hiba minimalizálásához a specificitást kell lehetőség szerint maximalizálni (vagyis a false positive rate-t minimalizálni). A ROC görbék vízszintes tengelye az (1-specificitás) értéket ábrázolja, tehát a ROC görbén egy olyan pontot kell keresnünk, amely közel helyezkedik a 0-hoz a vízszintes tengelyen, ugyanakkor a függőleges tengelyen minél közelebb van az 1-hez, azaz az érzékenység értéke is legyen minél nagyobb. Az utóbbi feltétel csak másodlagos, ugyanakkor mindenképp szükséges szem előtt tartanunk, hiszen választhatnánk a ROC görbén a (0,0) pontot, ahol a specificitás maximális, azonban ekkor az érzékenység 0 lenne, ami azt jelentené, hogy minden mintát a negatívak (*hamisak*) közé sorolnánk. Ahhoz, hogy pontosan meghatározzuk a kívánt konfidencia-küszöbértéket, számszerűsítsük a különböző típusú hibák költségeit. A következő hibafüggvény minimalizására törekszünk:

$$1.0 * FN + 4.0 * FP$$

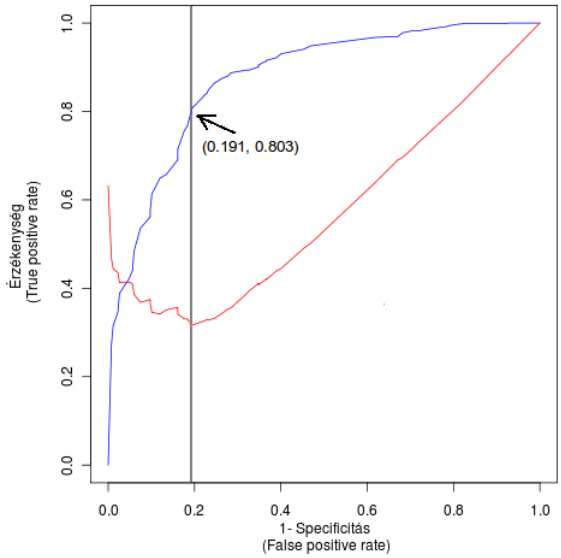
,ahol FN a false negative hibák száma, míg FP a false positive hibák száma. Tehát az elsőfajú hibákat 4-szer olyan súlyosnak tekintjük, mint a másodfajú hibákat.

Ezek után a specificitás függvényében meghatározhatjuk a hibafüggvény értékét. A ROC

görbére felrajzolva a hibafüggvény [0,1] intervallumba normalizált értékeit könnyen láthatjuk, hogy a ROC görbe melyik pontját kell kiválasztanunk. A kiválasztott pontot a 6.4 ábra mutatja, amelyen a hibaköltség normalizált értékeit piros színnel ábrázoltuk.

A pontban a hibaköltség **337** (a normalizált érték 0.316), a false positive rate – (1-specificitás) – **0.191**, míg az érzékenység **0.803**. A ponthoz tartozó konfidencia-küszöbértéket a következő módon kaphatjuk meg: meghatározzuk, hogy az adott pont választása esetén, mely pontokat sorolunk a pozitívak közé, majd ezen minták osztályozásának konfidenciaértékei közül kiválasztjuk a minimálisat. A kapott érték lesz a konfidencia-küszöbérték.

A tanítóhalmazon végzett tesztek esetén az így meghatározott konfidencia-küszöbérték **0.8**. Tehát ha a modellünk egy elemhez *igaz* címkét rendel, akkor csak abban az esetben fogadjuk el az osztályozást, ha a konfidencia értéke nagyobb, mint **0.8**. Ellenkező esetben a címkét *hamisnak* tekintjük.



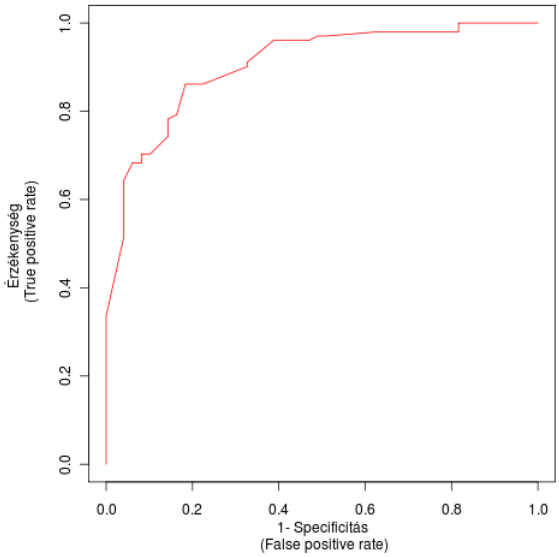
6.4. ábra. *Specificitás meghatározása a modell ROC görbéje és a hibaköltség alapján*

6.6.3. A teszthalmazon elért teljesítmény

Ahhoz, hogy a modellépítési eljárásunk eredményéről valóban releváns képet kapjunk, vizsgáljuk meg a modellünket egy eddig még nem használt, független adathalmazon. Ezt a célt szolgálja a teszthalmaz, amelyet a modellezési folyamat előtt különítettünk el a címkézett adathalmazból. A teszthalmaz független a tanítás során felhasznált adatoktól, így a modell számára teljesen ismeretlen mintákat tartalmaz.

Mivel már egy korábban létrehozott halmazon végezzük a modell tesztelését, ezért nincs szükség keresztvalidációra, így a modellünk tanítására felhasználhatjuk a teljes tanítóhalmazt. A létrehozott modell teszthalmazon elért eredményét a 6.5 ábrán látható ROC görbe mutatja. A görbe alatti terület értéke **0.905**, azaz a modell a teszthalmazon jobb eredményt ért el, mint a tanítóhalmazon a keresztvalidációs tesztek során, így megállapíthatjuk, hogy

a tanítóhalmazon való túltanulást sikerült elkerülnünk.



6.5. ábra. *A végső modell ROC görbéje a teszthalmazon, AUC=0.905*

A görbe alatti terület vizsgálata után nézzük meg mekkora első- és másodfajú hibát, ill. milyen pontosságot eredményez a modell, ha az előző fejezetben meghatározott konfidenciaküszöbértéket alkalmazzuk az osztályozás során. Ehhez vizsgáljuk meg az osztályozás konfúziós mátrixát:

Osztályozás \Valódi címke	**Hamis**	**Igaz**
Hamis	42	26
Igaz	7	75

6.1. táblázat. *A teszthalmazon végzett osztályozás konfúziós mátrixa*

A mátrix alapján felírhatjuk a helyesen osztályozott minták arányát:

$$(42 + 75)/(42 + 26 + 7 + 75) = 0.78$$

A helyesen osztályozott minták aránya kb. 78%, ami egész jó eredménynek mondható. Azonban fontosabb, hogy pusztán 7 *hamis* címkéjű elemet soroltunk az *igaz* osztályba, vagyis az elsőfajú hibák száma **7**, a false positive rate értéke $7/(7 + 42) = 0.14$, és emellett az *igaz* esetek kb. 74%-át sikerült azonosítanunk. Látható, hogy az első- és másodfajú hibák aránya meglehetősen pontosan tükrözi a hibafüggvényben megadott hibasúlyok arányát. Mivel az elsőfajú hibák számának minimalizálása volt a célunk, ezért a modellünk teljesítményét igen jónak értékelhetjük.

7. fejezet

A kapcsolatok megjelenítésere létrehozott webalkalmazás bemutatása

Az előző fejezetben létrehozott végső modell segítségével állítottuk elő a cégek alkalmazottainak teljes listáját. Ezúttal a modellt a teljes címkézett adathalmazon tanítottuk, majd minden olyan esetre lefuttattuk, ahol a hivatkozások közti távolság a cikkben a címkézésnél alkalmazott küszöbérték alatt volt. A modell konfidencia-határait a 6.6.2 fejezet eredményei szerint állítottuk be.

A modellt lefuttatva az adathalmazon összesen 1860 beosztottat azonosítottunk. Ezeket a kapcsolatok megjelenítésére létrehozott webalkalmazás adatbázisába importáltuk.

A webalkalmazás két fő funkcióval rendelkezik: lehetőség van további címkézett adatok előállítására, ill. kereshetünk a cégek és személyek között. Minden cég és személy rendelkezik egy adatlappal, amely aggregálja a róla összegyűjtött információkat.

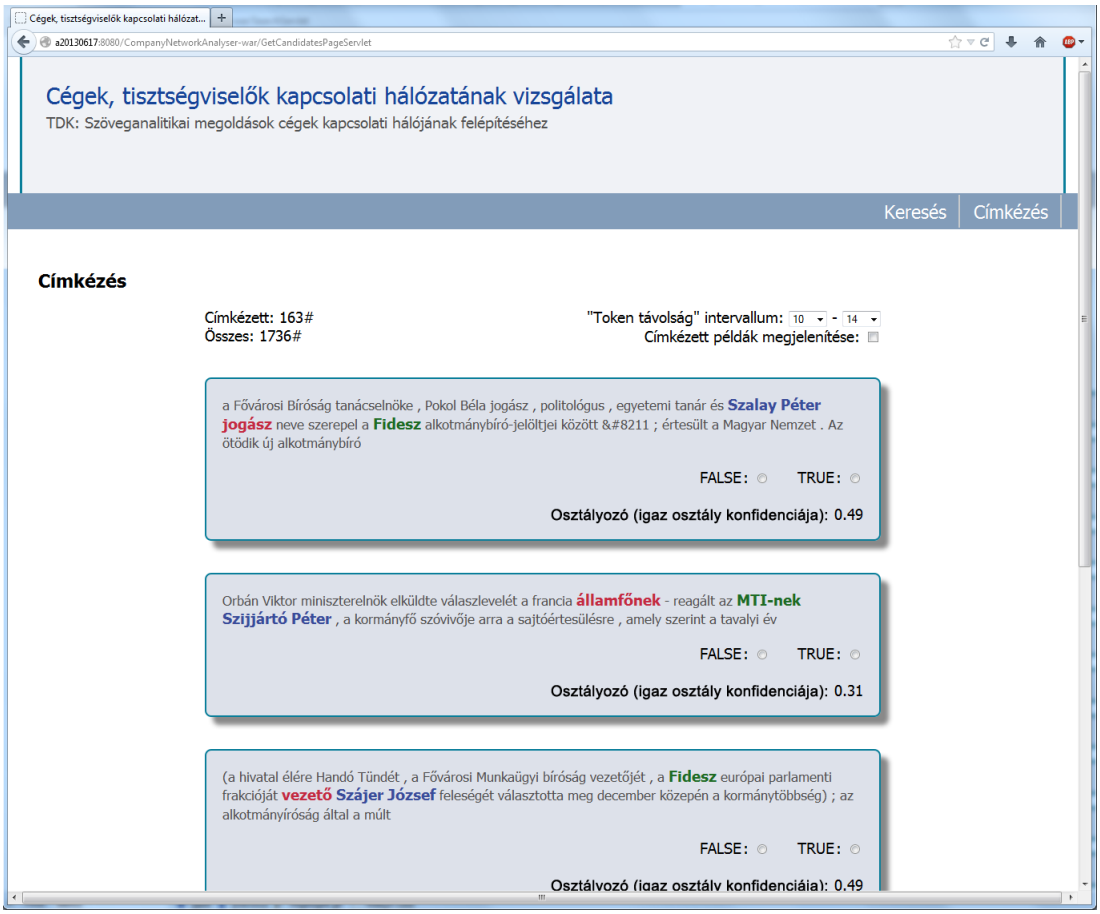
7.1. Adatok címkézése

A webalkalmazás „*Címkézés*” menüpontja alatt lehetőség van újabb címkézett minták előállítására, ill. a rosszul osztályozott elemek módosítására. A címkézésre szolgáló felhasználói felületet a 7.1 ábra mutatja.

Minden minta esetén láthatjuk a cikk megfelelő részletét, amiből a minta generálódott. A cikk szövegéből ki van emelve az azonosított cég, személy és beosztás, amelyekre a minta vonatkozik. A szöveg alapján eldönthetjük, hogy a hármas valóban egymáshoz kapcsolható-e és ez alapján megadhatjuk a címke értékét.

A listázott mintákat kétféle módon szűrhetjük: megadhatjuk a „tokentávolság” intervallumot, amely meghatározza, hogy mekkora lehet az elemekben a három hivatkozás minimális és maximális távolsága tokenekben mérve. A másik szűrési opció segítségével beállíthatjuk, hogy megjelenjenek-e a már korábban címkézett minták vagy sem. Ha az utóbbi beállítás segítségével nem jelenítjük meg a már címkézett eseteket, akkor egy minta felcímkézését követően a minta eltűnik, és automatikusan betöltődik egy újabb, még címke nélküli minta

felgyorsítva ezzel a címkézés folyamatát.



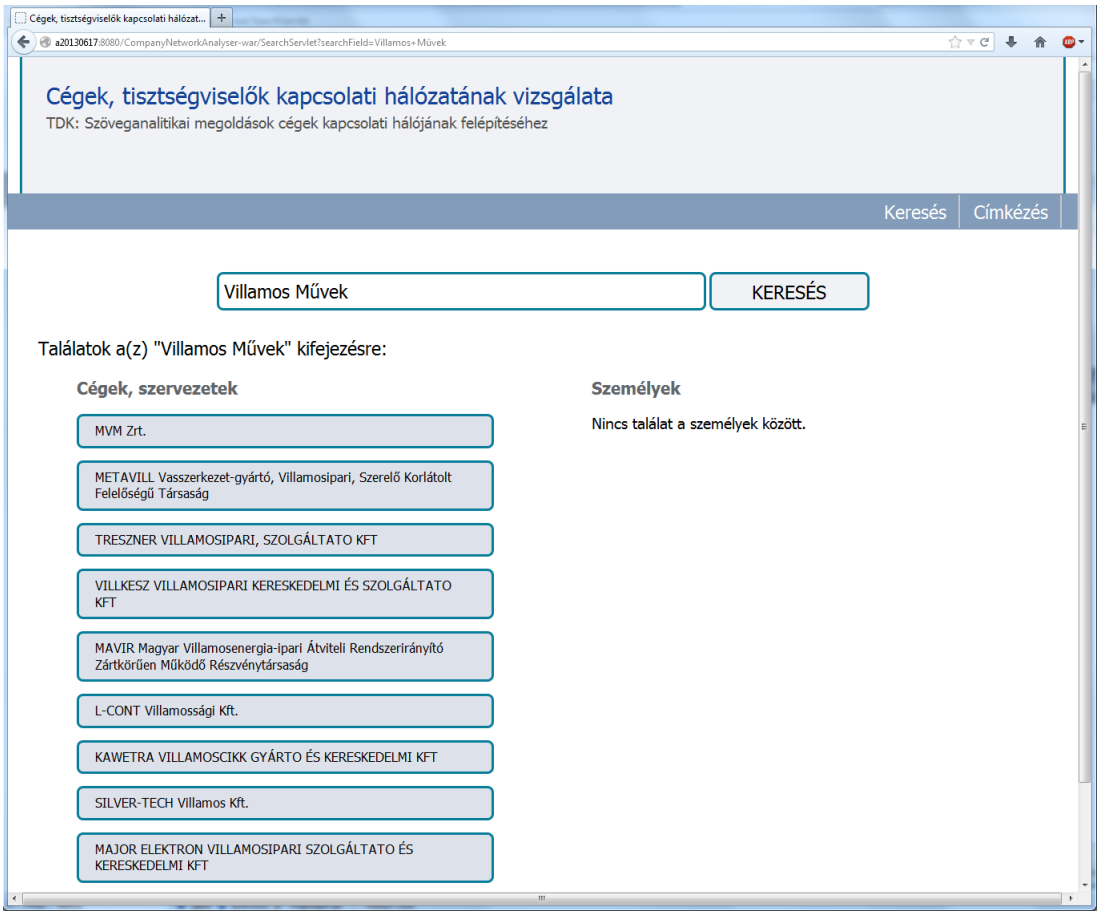
7.1. ábra. *Webalkalmazás felhasználói felülete: Címkézés*

A címkézési felületen az osztályozó modell által meghatározott kimenet is megjelenítésre kerül, így lehetőség nyílik, hogy a rosszul osztályozott mintákat átsoroljuk a megfelelő osztályba. Ezek a minták felhasználásra kerülnek a későbbi modellek tanítási folyamataiban, ezzel javítják az újabb modellek teljesítményét, ami a kapcsolatok pontosabb meghatározásához vezet.

7.2. Cégek, személyek keresése

A webalkalmazás másik fő funkciója a „*Keresés*” menüpont alatt érhető el. A keresési oldalt a 7.2 ábra mutatja. A keresőmezőben megadhatunk egy kifejezést, amellyel kereshetünk a cégek és személyek között. Minden kifejezés esetén külön találati listában láthatjuk a keresési kifejezésnek megfelelő cégeket és személyeket relevancia szerinti csökkenő sorrendbe rendezve. A relevancia értékének megállapításánál a keresési kifejezéssel való egyezés mértékét, ill. az adott elem kapcsolati hálójának méretét vesszük figyelembe.

A találati listából kiválasztva egy elemet az elem adatlapjára ugorhatunk. Az adatlap összegzi az adott cégről vagy személyről összegyűjtött információkat. A 7.3 ábra a Magyar Villamos Művek Zrt. adatlapját mutatja.

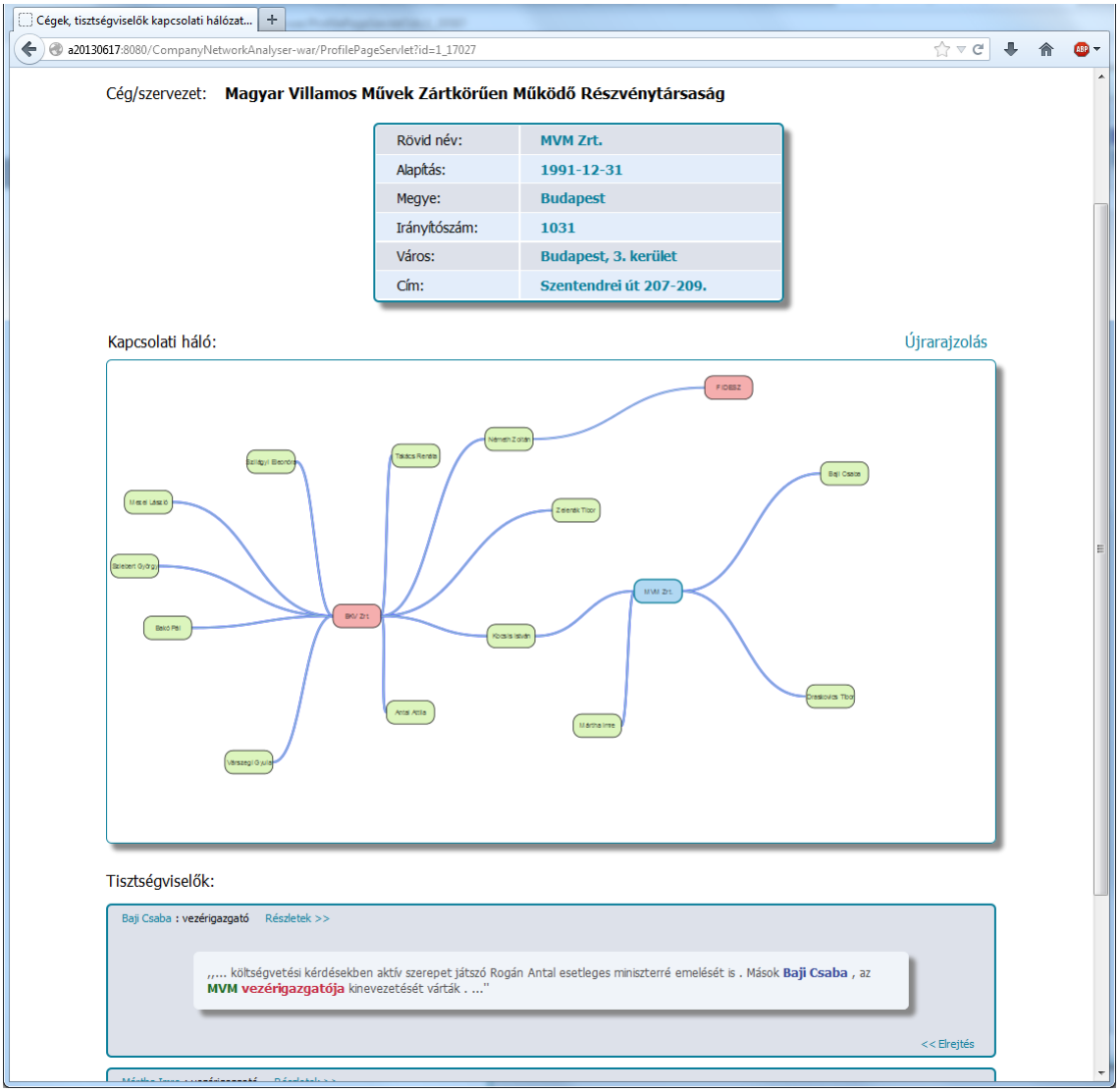


7.2. ábra. *Webalkalmazás felhasználói felülete: Keresés*

Az adatlapokon láthatóak az adott elemre vonatkozó alapvető adatok. Személyek esetén láthatjuk a személy nevét, a személy kapcsolati hálóját, és a személy által betöltött tisztségviselői pozíciókat a cég és a beosztás leírásával. Vállalat esetén megjelenítésre kerülnek a cégnyilvántartásbeli adatok, a vállalat kapcsolati hálója, ill. a vállalat azonosított alkalmazottai név és beosztás szerint.

A kapcsolati háló mind személyek, mind cégek esetén egy gráfként jelenik meg, ahol a csúcsok személyek és vállalatok, az élek pedig azon személyek és cégek között haladnak, amelyek kapcsolatban állnak egymással. A gráf elemeire kattintva átugorhatunk a kiválasztott elem adatlapjára.

A személyek által betöltött pozíciók, és a cégekhez tartozó alkalmazottak listájában minden elem esetén megtekinthetjük azokat a cikkrészleteket, amelyek alapján az adott kapcsolatot azonosítottuk.



7.3. ábra. *Webalkalmazás felhasználói felülete: az MVM Zrt. adatlapja*

8. fejezet

Összefoglalás

A munka során több információkivonási feladatot érintettünk, és oldottunk meg. Ezek közé tartoznak: megfelelő tartalmak kiszűrése HTML oldalakból, hírek szövegében cégek, személyek, beosztási pozíciók azonosítása, valamint a megtalált entitások közötti kapcsolatok felderítése. A részfeladatok közül több már önmagában is igen nehéz és szerteágazó feladat, amellyel komoly szakirodalom foglalkozik, így az általunk nyújtott megoldásokon természetesen még nagyon sokféle módon lehetne javítani.

A cikkek szövegében a cégek azonosítására szolgáló algoritmus sok esetben továbbra sem képes felismerni a cégekre történő hivatkozásokat, így érdemes lenne megvizsgálni milyen módon lehetne pontosítani ezen az algoritmuson. A személyek felismerésénél a középső nevek, ill. a toldalékolt alakok figyelembe vételével komoly javulást lehetne elérni, míg a pozíciók keresésében a többszavas titulusok felismerése jelenthetne nagy előrelépést. A cég, személy, beosztás hármasok kapcsolatainak keresésében a szövegkörnyezetben előforduló szavakat lenne érdemes figyelembe venni, de ezeken kívül még számtalan egyéb javítási lehetőség is elképzelhető.

A lehetséges javítási lépéseken túl rengeteg új, érdekes továbblépési irány is kínálkozik a munka folytatására. Érdekes lehet más típusú kapcsolatokat keresni a cégek között a cikkekben, esetleg a kapcsolatokat jellemezni a két cég közti viszony alapján. Emellett hasznos lehet a kapcsolatokhoz időskálát rendelni – pl. a cikkek dátumai alapján, amelyek kiszűrését már megvalósítottuk – és azt vizsgálni, hogy időben miként alakulnak, hogyan fejlődnek a cégek kapcsolatai vagy a személyek beosztásai.

A sok javítási lehetőség, és a lehetséges továbblépési irányok jó alapot nyújthatnak a további munkához, azonban az eddigi munka folyamán létrehozott eljárásokkal is már használható, értékelhető eredményeket sikerült elérnünk. Viszonylag nagy konfidenciával tudunk különböző cégekhez, szervezetekhez tartozó alkalmazottakat azonosítani hírportálok cikkeiben, továbbá a cégek és alkalmazottaik kapcsolati hálózata egy könnyen átlátható, egyszerű felületen érhető el külső felhasználók számára a létrehozott webalkalmazáson keresztül. Az alkalmazás hátránya, hogy egyelőre viszonylag kevés az azonosított kapcsolatok száma, azonban ezt növelhetjük a bemutatott javítási lehetőségek megvalósításával, vagy ennél jóval egyszerűbb módon: újabb cikkek letöltésével és feldolgozásával.

Összességében elmondható, hogy a munka folyamán létrehozott eljárással sikerült bebi-

zonyítanunk, hogy a weben elérhető magyar nyelvű cikkekben lehetséges a cégek, szervezetek alkalmazottainak azonosítása, és érdemes ezzel foglalkozni, hiszen fontos és értékes információkat vonhatunk le belőlük a cégekre, szervezetekre vonatkozóan.

Ábrák jegyzéke

Táblázatok jegyzéke

Irodalomjegyzék

[1] Douglas E Appelt, Jerry R Hobbs, John Bear, David Israel, and Mabry Tyson. Fastus: A finite-state processor for information extraction from real-world text. In *IJCAI*, volume 93, pages 1172–1178, 1993.

[2] David Meyer [aut, cre], Evgenia Dimitriadou [aut], Kurt Hornik [aut], Andreas Weingessel [aut], and Friedrich Leisch [aut]. *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*, 2012. R package version 1.6-1.

[3] Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201. Association for Computational Linguistics, 1997.

[4] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. A semantic approach to contextual advertising. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 559–566. ACM, 2007.

[5] Razvan Bunescu, Ruifang Ge, Rohit J Kate, Edward M Marcotte, Raymond J Mooney, Arun K Ramani, and Yuk Wah Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial intelligence in medicine*, 33(2):139–155, 2005.

[6] Venkatesan T Chakaravarthy, Himanshu Gupta, Prasan Roy, and Mukesh Mohania. Efficiently linking text documents with relevant structured information. In *Proceedings of the 32nd international conference on Very large data bases*, pages 667–678. VLDB Endowment, 2006.

[7] Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39(1):80–91, 1996.

[8] Robert B Doorenbos, Oren Etzioni, and Daniel S Weld. A scalable comparison-shopping agent for the world-wide web. In *Proceedings of the first international conference on Autonomous agents*, pages 39–48. ACM, 1997.

[9] David W Embley, Yuan Jiang, and Y-K Ng. Record-boundary discovery in web documents. *ACM SIGMOD Record*, 28(2):467–478, 1999.

[10] Ronen Feldman and James Sanger. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, 2007.

[11] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[12] Ralph Grishman. Information extraction: Techniques and challenges. In *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, pages 10–27. Springer, 1997.

[13] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *COLING*, volume 96, pages 466–471, 1996.

[14] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.

[15] Kevin Humphreys, George Demetriou, and Robert Gaizauskas. Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In *Pac Symp Biocomput*, volume 5, pages 505–516, 2000.

[16] Martin Jansche and Steven P Abney. Information extraction from voicemail transcripts. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 320–327. Association for Computational Linguistics, 2002.

[17] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[18] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[19] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[20] David Pinto, Andrew McCallum, Xing Wei, and W Bruce Croft. Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 235–242. ACM, 2003.

[21] Farkas Richárd and Szarvas György. Nyelvfüggetlen tulajdonnév-felismerő rendszer, és alkalmazása különböző domainekre. *MSZNY 2006*, pages 22–31, 2006.

[22] Sunita Sarawagi. Information extraction. *Foundations and trends in databases*, 1(3):261–377, 2008.

[23] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine learning*, 34(1-3):233–272, 1999.

[24] Anna Stavrianou, Periklis Andritsos, and Nicolas Nicoloyannis. Overview and semantic issues of text mining. *ACM Sigmod Record*, 36(3):23–34, 2007.

[25] Jarek Tuszynski. *caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc.*, 2012. R package version 1.14.