UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Tomaž Kuralt

Avtonomen sistem za združevanje podatkovnih omrežij

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: izr. prof. dr. Marko Bajec

Ljubljana, 2009

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01580/2009
Datum: 01.09.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **TOMAŽ KURALT**

Naslov: **AVTONOMEN SISTEM ZA ZDRUŽEVANJE PODATKOVNIH OMREŽIJ**
AUTONOMOUS SYSTEM FOR DATA NETWORKS INTEGRATION

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

V praksi nemalokrat naletimo na problem združevanja dveh ali več naborov podatkov, ki opisujejo isto domeno, v zgolj en nabor. Podatki so pogosto predstavljeni v obliki omrežij, saj so ta primerna za izvajanje različnih analiz. Tako običajen problem integracije podatkov dobi novo obliko, ki jo imenujemo združevanje podatkovnih omrežij.

V okviru diplomske naloge naj kandidat razvije sistem za reševanje omenjenega problema združevanja omrežij. Pri tem naj najprej razišče sorodno delo in kritično oceni primernost posameznih že znanih metod in pristopov. Pri samem razvoju sistema naj nato uporabi najbolj ustrezne in zanje predlaga morebitne izboljšave in prilagoditve. Na ta način naj odpravi zaznane pomanjkljivosti obstoječih pristopov in jih adaptira za konkreten problem združevanja omrežij. Na koncu naj kandidat predlagan sistem tudi ovrednoti in preuči njegovo delovanje na različnih domenah podatkov.

Mentor:

prof. dr. Marko Bajec

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Franc Solina

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Tomaž Kuralt,

z vpisno številko 63030307,

sem avtor diplomskega dela z naslovom:

Avtonomen sistem za združevanje podatkovnih omrežij

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
  izr. prof. dr. Marka Bajca
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 8.10.2009 Podpis avtorja:

Zahvala

Iskreno se zahvaljujem mentorju izr. prof. dr. Marku Bajcu za spodbudo in pomoč pri izdelavi tega diplomskega dela. Prav tako bi se rad zahvalil tudi Lovru Šublju za vse dane ideje, nasvete in komentarje. Nenazadnje pa bi se za podporo med celotnim študijem zahvalil tudi svoji mami in dekletu Jani.

Kazalo

Povzetek **1**

Abstract **2**

1 Uvod **3**
1.1 Motivacija . . . 3
1.2 Sorodno delo . . . 4
1.3 Cilj . . . 5

2 Teoretična podlaga **7**
2.1 Omrežja . . . 7
2.1.1 Definicija . . . 8
2.1.2 Omrežja realnega sveta . . . 9
2.1.3 Značilnosti omrežij . . . 11
2.1.4 Naključna omrežja . . . 14
2.2 Razločevanje entitet . . . 14
2.2.1 Atributne metrike . . . 15
2.2.2 Relacijske metrike . . . 17

3 Sistem za združevanje podatkovnih omrežij **24**
3.1 Predprocesiranje . . . 26
3.2 Grupiranje . . . 30
3.3 Vzpostavitev začetnega stanja . . . 31
3.3.1 Enostavna shema . . . 32
3.3.2 Relacijska shema . . . 32
3.4 Iterativno združevanje . . . 32
3.4.1 Odločitev o združitvi . . . 34

4 Eksperimentalni rezultati **38**
4.1 Podatki 38
4.1.1 Realni nabori podatkov 38
4.1.2 Naključni generator omrežij 40
4.2 Implementacija 41
4.3 Rezultati 43
4.3.1 Združevanje realnih omrežij 43
4.3.2 Združevanje sintetičnih omrežij 46
4.4 Diskusija 48

5 Zaključek **50**

Dodatki **52**

A Algoritmi **52**

Seznam slik **53**
Seznam tabel **54**

Seznam algoritmov **54**

Literatura **56**

Seznam uporabljenih kratic in simbolov

oznaka	opis
S	množica omrežij
N	omrežje
R	množica referenc
r_i	i-ta referenca oziroma i-ti gradnik omrežja
V	množica vozlišč
v_i	i-to vozlišče
E	množica povezav
e_i	i-ta povezava
$r.A$	množica atributov reference r
$r.a_i$	i-ti atribut reference r
T	množica entitet
t_i	i-ta entiteta
b	entitetni tip b
C	množica grozdov referenc
c_i	i-ti grozd
sim_i	izračunana podobnost para gradnikov i-te metrike
sim_A	izračunana podobnost para gradnikov atributne metrike
sim_R	izračunana podobnost para gradnikov relacijske metrike
D	vektor odločitev
d_i	odločitev i-te metrike o združitvi opazovanega para grozdov
F	vektor kontekstualnih lastnosti
f_i	i-ta kontekstualna značilnost
$Nbr(\cdot)$	soseščina opazovanega gradnika
$Amb(\cdot)$	stopnja dvoumnosti opazovanega gradnika
$IsAmb(\cdot)$	oznaka ali je opazovan gradnik dvoumen

Povzetek

V praksi pogosto naletimo na problem združevanja dveh ali več podatkovnih naborov v enoten nabor. Podatki so pogosto podani v obliki omrežij, saj nam ta omogočajo enostavno formulacijo odnosov med entitetami, hkrati pa so primerna tudi za izvajanje različnih analiz. V takih primerih običajni problem integracije podatkov dobi novo obliko, ki jo imenujemo združevanje podatkovnih omrežij.

V nalogi predstavimo avtonomen sistem za združevaje poljubnega števila omrežij v eno omrežje. Naloga sistema je torej odkriti morebitno redundanco znotraj množice podanih omrežij in vsako resnično entiteto v končnem omrežju predstaviti le z enim gradnikom. Uporabimo pristop skupinskega razločevanja, kar pomeni, da so posamezne odločitve sistema odvisne od prejšnjih. V ta namen uporabimo različne atributne in relacijske metrike. Naš model razločevanja je samoprilagodljiv glede na podan nabor podatkov in posledično sodelovanje domenskega eksperta pri določanju različnih parametrov modela ni potrebno, kot je to pogosto pri podobnih sistemih.

Sistem preizkusimo nad dvema realnima naboroma podatkov pri čemer so dobljeni rezultati zelo dobri. Izvedemo pa tudi različne eksperimente nad sintetičnimi podatki, kjer opazujemo kakovost združevanja v odvisnosti od različnih lastnosti podatkovnih naborov. Z delovanjem sistema smo sicer zadovoljni, vendar pa bi bilo potrebno sistem preizkusiti tudi na različnih drugih domenah podatkov, da bi lahko sklepali o uspešnosti delovanja sistema v splošnem.

Ključne besede:

integracija podatkov, združevanje omrežij, razločevanje entitet, omrežja

Abstract

In practice, we often face the problem of combining two or more data sets into a single one. Data is often presented in the form of networks, because they allow us an easy formulation of relations between entities. Networks also prove suitable for implementation of various analysis. In such cases, the usual problem of data integration obtains new form of combining data, which is called network integration.

In this work we present autonomous system for integration of any number of networks into a single network. Therefore, the task of the system is to identify possible redundancy within the set of given networks and to present every real entity as a single element in final network. We use a collective entity resolution approach which means, that every individual decision depends on previously made decisions in the system. For this purpose we use different attribute and relational metrics. Our entity resolution model is also self-adaptive according to the given data set. Consequently, participation of a domain expert is not needed at defining various model parameters, as this is common in similar systems.

The system was tested on two real world-data and obtained results are very good. We have also performed various experiments over synthetic data, where we have observed the quality of integration, regarding to different characteristics of input data sets. We are satisfied with the results of the system, but it would be necessary to perform additional tests over various other data domains, so we could estimate the efficiency of the system in general.

Key words:

data integration, network integration, entity resolution, networks

Poglavje 1

Uvod

1.1 Motivacija

Dandanes se pri mnogih poslovnih in tudi znanstvenih aplikacijah uporabljajo različne tehnike podatkovnega rudarjenja. Namen tovrstnih analiz je odkrivanje zakonitosti in novega znanja iz obstoječih podatkov, kar predvsem v poslovnem svetu lahko predstavlja konkurenčno prednost pred tekmeci. Podatke se za izvajanje analiz pogosto predstavi v obliki omrežja, saj nam ta omogočajo enostavno formulacijo odnosov med entitetami in tudi preglednejšo vizualizacijo. Pogost primer takšne predstavitve podatkov so socialna omrežja, v katerih obravnavane osebe ali skupine oseb predstavimo z vozlišči, povezave med vozlišči pa predstavljajo opazovane odnose med njimi.

Mnoge podatkovne baze in zbirke podatkov pa niso popolne, saj lahko vsebujejo nenatančne in redundantne podatke o entitetah realnega sveta. Posledično so tudi omrežja zgrajena nad takšnimi podatki nepopolna. Rezultati analiz nad takšnimi omrežji pa običajno niso dobri in lahko vodijo do napačnih zaključkov. Problem se še poveča, kadar je potrebno analizirati več naborov podatkov iste domene, ki so predstavljeni vsak s svojim omrežjem. Analiza vsakega omrežja posebej bi lahko vodila do slabih rezultatov, saj so vsako posebej ponavadi nepopolna in le združena skupaj dajo celotno predstavo o problemski domeni. V praksi tako pogosto v fazi priprave podatkov naletimo na problem združevanja omrežij, z namenom doseganja boljših in bolj relevantnih rezultatov analize.

V primeru, da je vsak gradnik obdelovanih omrežij opremljen z enoličnim identifikatorjem, je omenjen problem trivialen. Omrežja lahko enostavno zlijemo v eno tako, da združimo tiste gradnike, ki imajo enako vrednost enoličnega identifikatorja. V praksi je problem lahko kompleksnejši, saj podatki pogosto

niso opremljeni z enoličnim identifikatorjem ali pa so le-ti različni. Možno je tudi, da gradniki omrežij, ki predstavljajo isto entiteto niso predstavljeni z enakimi atributi in na enak način. To še dodatno oteži postopek združevanja omrežij, saj je prostor za primerjavo gradnikov močno okrnjen.

V nadaljevanju naloge najprej v razdelku 1.2 predstavimo sorodno delo, nato v 1.3 sledi natančen opis ciljev naloge. V razdelku 2 podamo teoretično podlago; teorijo omrežij opišemo v razdelku 2.1, v 2.2 pa predstavimo problem razločevanja entitet. Sistem za združevanje omrežij podrobno opišemo v razdelku 3, katerega preizkusimo na različnih naborih podatkov. Rezultate podamo v razdelku 4. Na koncu pa v razdelku 5 podamo še kritično oceno in predloge za nadaljnje delo.

1.2 Sorodno delo

Problem združevanja omrežij spada v širše problemsko področje razločevanja entitet. Prvo delo [29] s tega področja so Newcombe in sodelavci objavili že leta 1959. V njem so opisali metode za omejevanje števila potrebnih primerjav med referencami[1]. Njihovo delo sta nadaljevala Fellegi in Sunter [14], v katerem sta definirala verjetnostni model za razvrščanje parov referenc v dva možna razreda (pari referenc, ki *predstavljajo* in *ne predstavljajo* iste entitete). Avtorji v [18, 41] so preučevali različne načine avtomatske nastavitve potrebnih parametrov Fellegi-Sunterjevega modela. V [16] pa so se avtorji ukvarjali z razločevanjem entitet z uporabo omenjenega modela predvsem nad velikimi zbirkami podatkov. Svojo pozornost pri obravnavi problema so namenili predvsem hitrosti iskanja potencialnih duplikatov in ne toliko natančnosti. Vsem opisanim pristopom je skupno to, da za razločevanje entitet uporabljajo le njihove statične lastnosti, torej atribute.

Druga skupina pristopov [2, 5, 6, 8, 9, 12, 19, 22, 23, 32, 34, 35] za razločevanje, poleg atributov uporablja tudi relacije oziroma povezave, s katerimi modeliramo odnose med opazovanimi referencami. Naivni pristopi [2, 9] iz te skupine povezave obravnavajo le kot dodaten nabor atributov, katere se uporabi med samim procesom razločevanja entitet. Naprednejše metode razločevanja entitet pa relacije pri procesu razločevanja tudi eksplicitno uporabljajo. V [5, 6, 8] avtorji predstavijo, kako se pri razločevanju entitet kot dodaten dokaz lahko uporabi tudi primerjava soseščin primerjanih referenc. Chen s sodelavci v [12] in Kalashnikov s sodelavci v [22, 23] predlagajo med-

[1] Z izrazom referenca označujemo zapis v podatkovni bazi ali zbirki podatkov, ki predstavlja neko določeno resnično entiteto.

seboj podobne modele razločevanja entiet, ki so osnovani na merjenju moči povezanosti med opazovanimi referencami. Pri teh se za razločevanje torej ne uporablja primerjava soseščin, temveč množica poti, ki obstaja med obravnavanimi referencami. Na podlagi te se meri moč medsebojne povezanosti, ki služi kot mera za ocenjevanje podobnosti med opazovanimi referencami. Omenimo pa lahko še nekatere druge modele, ki za razločevanje prav tako uporabljajo relacije, osnovani pa so na različnih verjetnostnih modelih kot so Markovska logika [35] in Latentna Dirichletova alokacija - LDA [6, 34] (ang. *Latent Dirichlet Allocation*).

V zadnjem obdobju se veliko pozornost na področju razločevanja entitet namenja tudi problemu avtomatskega prilagajanja različnih modelov glede na vhodno množico podatkov. Izkazalo se je namreč, da je kakovost razločevanja pri številnih modelih močno odvisna od ustrezne nastavitve parametrov. Bilenko v svojem delu [3] predstavi metodo nadzorovanega učenja, s katero prilagaja delovanje predlagane atributne metrike glede na vhodno množico podatkov. Podobno tudi Chen s sodelavci [12] uporabi metodo nadzorovanega učenja za nastavitev uteži povezavam, katere uporabijo v predlaganem modelu razločevanja entitet. Isti avtorji pa v [11] predstavijo tudi model kombiniranja rezultatov več različnih in medseboj neodvisnih sistemov za razločevanje v skupen rezultat.

Zaključimo, da je pristope razločevanja entitet, ki temeljijo na primerjavi atributov, možno uporabiti tudi pri našem problemu združevanja omrežij. Vendar pa bi z uporabo omenjenih pristopov obravnavali le statične lastnosti gradnikov omrežij ne pa tudi strukturnih oziroma relacijskih. Pri obstoječih pristopih, ki za razločevanje uporabljajo tudi relacije, pa se pojavi problem, da so le-te osredotočene predvsem v razločevanje vozlišč, ne pa tudi povezav. Pri problemu združevanja omrežij pa je enako pomembno razločevanje tako povezav kot tudi vozlišč.

1.3 Cilj

Cilj naloge je razviti avtonomen sistem za združevanje poljubnega števila podatkovnih omrežij. Naloga sistema je iz množice podanih omrežij zgraditi eno omrežje, v katerem je vsaka resnična entiteta predstavljena le z enim gradnikom omrežja. V novonastalem omrežju torej ni redundance in je kot tako primerno za nadaljnjo uporabo in analiziranje. Dodatno želimo, da je sistem sposoben avtomatskega prilagajanja modela združevanja podanemu naboru podatkov, saj je kakovost delovanja podobnih sistemov v veliki meri odvisna

od ustrezne nastavitve parametrov. V okviru naloge predlagan sistem tudi ovrednotimo in preučimo njegovo delovanje na različnih naborih podatkov iz različnih podatkovnih domen.

Poglavje 2

Teoretična podlaga

V nadaljevanju najprej podamo teoretično podlago omrežij ter formalno definiramo naš problem in cilj združevanja omrežij. Opišemo tudi različna realna omrežja ter predstavimo nekatere skupne značilnosti različnih omrežij.

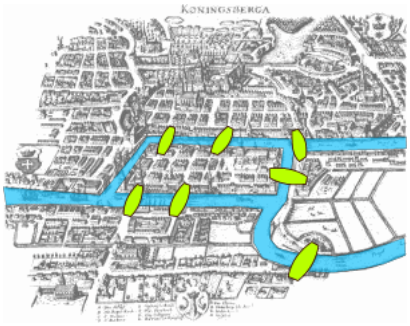
V drugem delu tega poglavja pa opišemo področje razločevanja entitet in predstavimo nekatere primerjalne metrike, ki jih uporabimo tudi pri izdelavi našega sistema za združevanje omrežij.

2.1 Omrežja

Teorija grafov je veda diskretne matematike, ki proučuje lastnosti omrežij. Implicitno je prisotna že od starih Grkov dalje (skeleti pravilnih in arhimedskih teles so grafi). Za njen začetek pa štejemo 18. stoletje, ko je švicarski matematik Leonhard Euler v delu *Seven Bridges of Königsberg* podal rešitev problema ali se je mogoče po mestu Königsberg sprehoditi tako, da vsak most prečkamo le enkrat (glej sliko 2.1).

Naslednje pomembno delo je leta 1852 izdal angleški matematik Francis Guthrie, v katerem je kot prvi podal problem štirih barv. V tem delu se sprašuje ali je možno poljuben zemljevid držav pobarvati le s štirimi barvami tako, da nobeni sosednji državi nista pobarvani z isto barvo. Na videz preprost problem je ostal nerazrešen več kot stoletje. Njegovo rešitev pa sta leta 1976 podala Kenneth Appel in Wolfgang Haken. Dokazala sta, da je mogoče vsako omrežje z določenimi omejitvami, pobarvati na omenjen način.

V zadnjem obdobju pa se pojavljajo nove smeri raziskovanja omrežij. Pri slednjih ni več v ospredju proučevanje lastnosti posameznih gradnikov v malih omrežjih, temveč se obširno proučujejo statistične lastnosti velikih omrežij, generatorji naključnih omrežij in različni modeli rasti. Vzroki za pojavljanje



Slika 2.1: Prikaz problema *Seven Bridges of Königsberg*: Problem rešimo tako, da narišemo graf, kjer povezave predstavljajo mostove in ugotovimo, da v njem ne obstaja Eulerjev obhod [43].

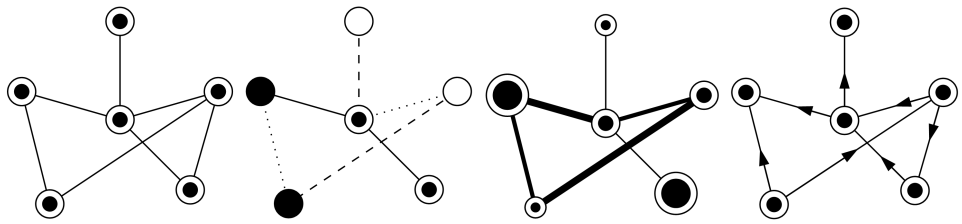
omenjenih novih smeri pa ležijo predvsem v zmožnostih procesiranja ogromnih količin podatkov.

2.1.1 Definicija

Omrežje N je podatkovna struktura, s katero modeliramo in prikazujemo relacije med opazovano množico objektov. V svoji najenostavnejši obliki je podano kot dvojka $N = \{V, E\}$, pri čemer z $V = \{v_i\}$ označimo množico vozlišč, z $E = \{e_i\}$ pa množico relacij oziroma povezav med vozlišči. Vozlišča in povezave so lahko obogatene še z atributi, s katerimi povečamo izrazno moč omrežij. Omenimo, da so tako vozlišča kot tudi povezave lahko različnih tipov. Množico entitetnih tipov pa označimo z $B = \{b_i\}$.

V omrežjih lahko povezave utežimo, s čimer lahko modeliramo stopnjo povezanosti med opazovanima vozliščema. Prav tako lahko povezave tudi usmerimo, s čimer prikažemo enostransko povezanost med dvema vozliščema. Omrežja s tovrstnimi povezavami imenujemo usmerjena omrežja. Poznamo pa tudi hiper-omrežja pri katerih povezava oziroma hiper-povezava lahko naenkrat povezuje tudi več kot dve različni vozlišči. Za povezave pa lahko tudi velja, da se lahko pričnejo in končajo v istem vozlišču, kar imenujemo zanke. Nekatera izmed opisanih omrežij so na enostaven način prikazana na sliki 2.2.

Pri problemu združevanja omrežij imamo torej podano množico omrežij $S = \{N_i\}$. Gradniki (vozlišča in povezave) enega ali večih omrežij lahko opisujejo neko isto entiteto, zato množico gradnikov označujemo s skupnim imenom, množica referenc $R = V \cup E = \{r_i\}$. Atribute referenc pa označimo z $r.a_1, r.a_2, r.a_3, \ldots, r.a_n$.



Slika 2.2: Prikaz različnih tipov omrežij [31]: (a) neusmerjeno omrežje z le enim tipom vozlišč in z le enim tipom povezav; (b) neusmerjeno omrežje z različnimi tipi vozlišč in različnimi tipi povezav; (c) omrežje z različno uteženimi povezavami; (d) usmerjeno omrežje.

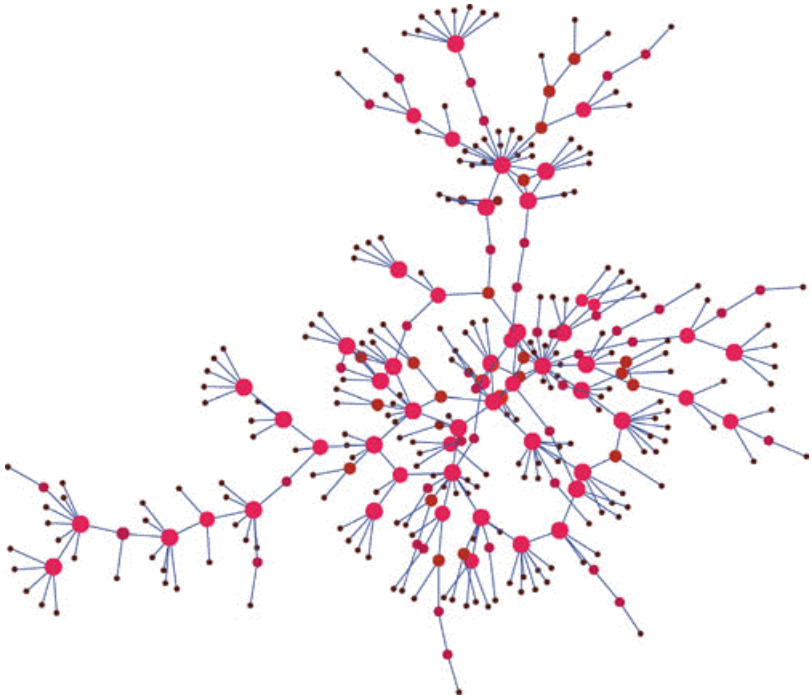
Cilj sistema za združevanje omrežij je iz množice podanih omrežij skonstruirati zgolj eno omrežje, v katerem je vsaka resnična entiteta predstavljena natanko z enim gradnikom. Potrebno je torej določiti neznano množico entitet $T = \{t_i\}$, pri tem pa je za vsako entiteto t potrebno poiskati množico referenc $t.R = \{r_i\}$, ki opazovano entiteto dejansko predstavljajo.

2.1.2 Omrežja realnega sveta

V nadaljevanju podajamo ohlapno razčlenitev realnih omrežij v štiri različne kategorije: socialna omrežja, informacijska omrežja, tehnološka omrežja in biološka omrežja [31].

Socialna omrežja je skupina omrežij med katera prištevamo vsa tista, s katerimi opisujemo določena razmerja oziroma interkacije med ljudmi ali skupinami ljudi. Primer takega omrežja je omrežje prijateljstva med posamezniki. Socialna omrežja so bila v preteklosti tudi najbolj raziskovana skupina omrežij, vendar ne s strani matematikov, temveč s strani sociologov. Predlagan sistem za združevanje omrežij smo testirali nad realnimi omrežji iz te skupine.

Informacijska omrežja v literaturi včasih poimenujejo tudi *omrežja znanja*. Njihov tipičen predstavnik je omrežje citiranj med akademskimi članki, kjer povezava med opazovanima člankoma obstaja, če avtor prvega članka citira avtorja drugega. Struktura tovrstnih omrežij nazorno prikazuje strukturo znanja shranjenega v vozliščih. Od tod izvira tudi ime informacijska omrežja.



Slika 2.3: Primer socialnega omrežja spolnih odnosov med ljudmi z virusom HIV [33].

Tehnološka omrežja predstavljajo kategorijo omrežij, kamor prištevamo vsa umetno izdelana omrežja, ki so tipično namenjena distribuciji nekega vira ali blaga. Tipični primeri tovrstnih omrežij so električna in vodovodna omrežja, omrežja fizičnih povezav med računalniki, omrežja zračnih, cestnih in železniških povezav, telefonska omrežja... Pri tovrstnih omrežjih pa je običajno pomembna tudi sama geografska lokacija gradnikov, kar je pri drugih tipih omrežij manj pogosto.

Biološka omrežja pa predstavljajo zadnjo kategorijo omrežij iz opisane delitve. V to kategorijo prištevamo vsa omrežja, s katerimi predstavljamo različne biološke sisteme in opisujemo interakcije ali razmerja med elementi teh sistemov.

2.1.3 Značilnosti omrežij

V nadaljevanju podamo opis nekaterih značilnosti, ki so skupne večini realnih omrežij ne glede na področje in namen uporabe. Nekatere izmed opisanih značilnosti uporabimo tudi pri razvoju predlaganega sistema za združevanje omrežij.

Učinek majhnega sveta

Stanley Milgram [28] je izvedel eksperiment, pri katerem sicer ni bilo konstruirano nobeno omrežje, vseeno pa je z njim pokazal na pomembno značilnost topologije omrežij. Naključno izbranim osebam je naročil, naj preko pisem navežejo stik z neko drugo osebo. Ključno navodilo pri tem poskusu je bilo, da mora vsak prejemnik pisma le-to poslati naprej tisti osebi, za katero meni, da bolj verjetno pozna končnega prejemnika. Večina pisem je na cilj prišla v le približno šestih korakih. S tem je nakazal, da je razdalja med poljubnima vozliščema v omrežju majhna oziroma veliko manjša, kot bi pričakovali glede na velikost omrežja. Učinek majhnega sveta običajno merimo s povprečno razdaljo med vsakim parom vozlišč. V neusmerjenih omrežjih tako definiramo oceno l:

$$l = \frac{1}{\frac{1}{2}n(n-1)} \sum_{i \geq j} d_{v_i,v_j}, \tag{2.1}$$

kjer je d_{v_i,v_j} dolžina geodetke[1] med vozliščema v_i in v_j. Omenjena značilnost omrežij je bila kasneje preizkušana in tudi dokazana v številnih drugih realnih omrežjih in ne le v socialnih [30].

Tranzitivnost

Ena izmed osnovnih metrik, s katerimi ugotavljamo značilnosti omrežij, je tudi tranzitivnost. Stopnjo tranzitivnosti ocenjujemo s koeficientom združevanja (ang. *clustering coefficient*), ki je bil dolgo časa predmet zanimanja tako empiričnih raziskovalcev kot tudi teoretikov. V teoriji omrežij tranzitivnost pomeni, da se verjetnost povezave med vozliščema v_i in v_k poveča, če je vozlišče v_i povezano z vozliščem v_j, v_j pa z v_k. V smislu socialnih omrežij to lahko pomeni, da je prijatelj mojega prijatelja tudi moj prijatelj, v smislu topologije omrežij pa se tranzitivnost kaže v povečanem številu trikotnikov[2] v omrežju.

[1] Pot med vozliščema v in u sestavlja zaporedje povezav, preko katerih smo prišli od v do u. Geodetka pa je najkrajša pot med opazovanima vozliščema v smislu, da jo sestavlja najmanjše število povezav.

[2] Trikotnik označuje trojico vozlišč, ki so povezana vsako z vsakim.

Stopnjo tranzitivnosti lahko ocenjujemo z globalnim koeficientom združevanja, ki je definiran kot:

$$C = \frac{3 \times \text{število trikotnikov}}{\text{število povezanih trojic vozlišč}}. \tag{2.2}$$

V številcu je faktor 3, saj pojav vsakega trikotnika zadosti trem trojicam vozlišč. Posledično to pomeni, da velja $0 \leq C \leq 1$. Alternativno definicijo koeficienta združevanja, ki je prav tako pogosto uporabljana, sta podala Watts in Strogatz v [39]. Lokalno vrednost koeficienta združevanja sta definirala kot:

$$C_i = \frac{\text{število trikotnikov, ki vsebujejo } v_i}{\text{število trojic vozlišč, ki vsebujejo } v_i}. \tag{2.3}$$

Za vozlišča, ki imajo stopnjo enako 0 ali 1, je tako števec kot tudi imenovalec enak 0. Zato za ta vozlišča navadno določimo $C_i = 0$. Globalna vrednost koeficienta združevanja pa je nato podana kot povprečje lokalnih vrednosti:

$$C = \frac{1}{n} \sum_i C_i. \tag{2.4}$$

Prispevek vozlišč, ki so povezana z manj povezavami, se pri slednji definiciji koeficienta združevanja upošteva močneje. To pa pomeni, da so rezultati obeh metrik pri določenih omrežjih lahko tudi različni [31].

Centralnost

Pri analizi omrežij pogostokrat želimo odgovoriti na vprašanje: *"Kateri gradnik omrežja je najpomembnejši?"*. Na to vprašanje sicer obstaja veliko odgovorov, odvisno predvsem od tega kako definiramo pojem *najpomembnejši*. V splošnem pa pomembnost gradnikov v omrežjih merimo z metrikami centralnosti. V nadaljevanju predstavimo štiri različne metrike.

Najenostavnejša metrika za opazovanje centralnosti je stopnja vozlišča (ang. *degree centrality*). Stopnja vozlišča C_D je podana s številom njegovih povezav. To je absolutna mera centralnosti vozlišča, ki pa je ne moremo uporabiti za primerjavo centralnosti pri obravnavanju različnih omrežij. Zato mero običajno normaliziramo in jo podamo kot:

$$C'_D(v) = \frac{C_D(v)}{|V| - 1}, \tag{2.5}$$

kjer z $|V|$ označimo število vseh vozlišč v omrežju. Navkljub svoji preprostosti se je uporaba stopnje vozlišča, kot mere za merjenje centralnosti, izkazala za

odlično v socialnih omrežjih, kjer so z njo merili moč oziroma ugled. Ljudje z več povezavami imajo običajno tudi večjo moč. Omenjeno mero centralnosti pa lahko uporabimo tudi v usmerjenih omrežjih, le da pri tem ločimo dve možnosti opazovanja stopenj. Z vhodno stopnjo označujemo število vhodnih povezav, izhodna stopnja pa predstavlja število izhodnih povezav. Na prejšnjem primeru socialnih omrežij prva lahko pomeni podporo, druga pa vplivnost.

Izboljšana metrika za centralnost (ang. *eigenvector centrality*), prav tako kot prejšnja, temelji na stopnji vozlišča. Pri tem pa upošteva, da niso vse povezave enako pomembne. Povezave, ki neko opazovano vozlišče povezujejo z vozlišči z večjo centralnostjo, bodo temu vozlišču doprinesle več kot tiste povezave, ki se navezujejo na vozlišča z manjšo. Če to ponazorimo na prejšnjem primeru socialnih mrež to pomeni, da bodo bolj vplivni ljudje svojo moč oziroma ugled posredovali svoji okolici, medtem ko ljudje brez moči tega ne morejo. Naslednji dve metriki za opazovanje centralnosti pa sta osnovani na konceptu poti v omrežju.

Sabidussi je v [36] predlagal naslednjo mero centralnosti (ang. *closeness centrality*):

$$C_C(v) = \frac{|V| - 1}{\sum_{v' \in V} d_{v,v'}}, \tag{2.6}$$

ki je osnovana na konceptu dostopnosti. Po tej metriki so najbolj centralna tista vozlišča, ki so 'dovolj blizu' vsem ostalim. Ta mera centralnosti je boljša od prejšnjih, ker upošteva tudi posredne sosede in ne le neposrednih. Pri usmerjenih omrežjih pa imamo pri tej metriki dve možnosti in centralnost lahko izračunavamo glede na usmerjene in neusmerjene povezave posebej.

Pri omrežjih pa ni pomembna le oddaljenost gradnika omrežja od ostalih, ampak tudi, kateri gradniki ležijo na najkrajših poteh. Takšni gradniki so namreč pomembni za pretok informacij, saj se v primeru njihove odstranitve, najkrajša pot večini parov lahko poveča. Freeman je v [15] definiral centralnost, ki slednje upošteva. Vmesna centralnost (ang. *betweenes centrality*) gradnika r je definirana kot vsota verjetnosti preko vseh parov možnih gradnikov omrežja, da bo najkrajša pot med pari gradnikov potekala skozi r:

$$C_B(r) = \sum_{i<j} \frac{\sigma_{r_i,r_j}(r)}{\sigma_{r_i,r_j}}. \tag{2.7}$$

S $\sigma_{r_i,r_j}(r)$ označimo število najkrajših poti med r_i in r_j skozi r, s σ_{r_i,r_j} pa označimo število vseh najkrajših poti med gradnikoma r_i in r_j. Pomembna lastnost vmesne centralnosti je tudi to, da jo lahko uporabimo tako nad vozlišči kot tudi povezavami. Zato smo namenoma v formuli uporabili oznako r, s katero označimo množico vseh gradnikov omrežja.

2.1.4 Naključna omrežja

Raziskovalci na področju analize omrežij posvečajo veliko pozornost tudi proučevanju naključnih omrežij. Ta so pomembna predvsem z vidika proučevanja obnašanja omrežij. V našem primeru pa jih izkoristimo kot alternativni vir izgradnje naključnih naborov podatkov.

Poissonova omrežja so neodvisno odkrili Solomonoff in Rapoport [38] ter Erdös in Renyi [4]. Predlagali so enostaven model gradnje naključnih omrežij, pri katerem med vsakim parom vozlišč postavimo povezavo z verjetnostjo p. Erdos in Renyi sta omenjen model poimenovala $G_{n,p}$, kjer je n število vozlišč, p pa verjetnost pojavljanja povezave med vsakim parom vozlišč. Verjetnost, da ima naključno izbrano vozlišče stopnjo k je tako:

$$p_k = \binom{n}{k} p^k (1-p)^{n-k} \simeq \frac{z^k e^{-z}}{k!}, \tag{2.8}$$

kjer je $z = p(n-1)$ povprečna stopnja vozlišč. Porazdelitev stopenj je torej binomska oziroma Poissonova v limiti, ko $n \to \infty$ od koder omrežjem izvira tudi ime [31].

Poissonov model naključnih omrežij dobro posnema učinek majhnega sveta (opisan v začetku razdelka 2.1.3). Izkaže pa se, da se porazdelitev stopenj močno razlikuje od porazdelitve stopenj pri realnih mrežah. Pri slednjih je ta močno raztegnjena v desno (ang. *right-skewed*) [31]. Zaradi omenjene pomanjkljivosti se veliko pozornosti posveča omrežjem s porazdelitvijo stopenj po potenčnem zakonu, saj je bila ta opažena pri velikem številu proučevanih realnih omrežij. Eden izmed modelov izgradnje naključnih omrežij po potenčnem zakonu je Eppsteinov model [13], ki je osnovan na načelu prednostne povezanosti. To pomeni, da imajo vozlišča z večjo stopnjo tudi večjo verjetnost, da jim bo dodana nova povezava.

2.2 Razločevanje entitet

S problemom določanja ali različni opisi določajo isto entiteto, se ukvarjajo metode razločevanja entitet. Dandanes je to pogost problem, ki se v različnih oblikah pojavlja na mnogih področjih. Najpogosteje se s tem problemom srečujemo v informatiki, tako pri čiščenju, kot tudi integraciji podatkov. Prav tako pa lahko problem v drugih oblikah prepoznamo tudi na drugih področjih. Osnova razločevanja entitet leži v medsebojnem primerjanju razpoložljivih podatkov o referencah. Obstajajo različne primerjalne metrike in nekatere od njih v nadaljevanju tudi podrobneje predstavimo.

2.2.1 Atributne metrike

V prvo skupino primerjalnih metrik sodijo tiste, ki omogočajo primerjavo referenc na osnovi njihovih statičnih lastnosti - atributov.

Levenshteinova primerjava

Levenshteinova primerjava [24] je mera za merjenje podobnosti med dvema nizoma. Podana je kot minimalno število potrebnih operacij za prenos enega niza v drugega. Pri tem je operacija definirana kot vnos, izbris ali zamenjava enega znaka. Manjše kot je število potrebnih operacij za prenos enega niza v drugega, bolj sta si primerjana niza med seboj podobna. Če sta niza enaka je število potrebnih operacij 0 in posledično je podobnost med njima največja, torej 1.

Obstajajo pa tudi različne posplošitve te metode, ki operacijo definirajo drugače. Za primer lahko navedemo *Damerau-Levenshteinovo primerjavo*, ki poleg omenjenih definira dodatno operacijo kot zamenjavo dveh znakov.

Jaro in Jaro-Winklerjeva primerjava

Podobno kot prejšnja, je tudi Jaro [20] primerjava metrika za ugotavljanje podobnosti med nizoma znakov. Podobnost po Jaro metriki je za podani vrednosti dveh atributov a_i in a_j definirana kot:

$$d_{jaro}(a_i, a_j) = \frac{1}{3}(\frac{m}{|a_i|} + \frac{m}{|a_j|} + \frac{m-t}{m}), \tag{2.9}$$

kjer je

- m število ujemajočih se znakov. Primerjana znaka veljata za med seboj ujemajoča, če sta enaka in če razdalja med njima ni več kot $\lfloor \frac{\min(|a_i|,|a_j|)}{2} \rfloor$.
- t število transpozicij $t = \frac{m'}{2}$. Z m' označimo število ujemajočih se znakov za katere pa eksplicitno velja, da niso na istih mestih.
- $\frac{1}{3}$ je faktor, s katerim (v tem primeru) enakomerno utežimo ujemanje znakov tako prvega in drugega niza, kot tudi transpozicij. Obstajajo pa tudi različne variacije, ki posamezne komponente primerjave utežijo različno.

Če slednje ponazorimo na primeru vrednosti *avto* in *atvo* je $m = 4$, $|a_1| = |a_2| = 4$ in $t = \frac{2}{2}$, saj je $m' = 2$ (črki v in t). Tako je podobnost po opisani metriki enaka $\frac{11}{12}$.

Jaro-Winklerjeva primerjava [40, 41] pa je razširitev opisane Jaro metrike in je prav tako namenjena primerjanju nizov znakov. Ta poveča oceno Jaro na način, da poišče največjo dolžino skupne predpone za oba opazovana niza in osnovno oceno poveča po formuli:

$$d_{jaro-winkler}(a_i, a_j) = d_{jaro}(a_i, a_j) + (l \times p(1 - d_{jaro}(a_i, a_j))). \tag{2.10}$$

Dolžino skupne predpone nizov označimo z l, p pa je konstanta, s katero utežimo pomembnost ujemanja predpon. Nastavitev omenjenih parametrov je sicer prepuščena uporabniku, vendar pa se v veliki večini primerov uporabijo vrednosti parametrov, ki jih je v svojih aplikacijah uporabljal tudi Winkler; $p = 0.1$ in $l \leq l_{max} = 4$.

Cos TF-IDF primerjava

Atribute referenc lahko alternativno obravnavamo tudi kot vrečo besed (ang. *bag of words*). Vrstni red besed v atributu tedaj ne igra nobene vloge. V nadaljevanju predstavimo metriko Cos TF-IDF [10, 26], ki je osnovana na statistični metodi TF-IDF (ang. *Term Frequency - Inverse document Frequency*) [37], s katero ocenjujemo pomembnost pojavljanja besede v atributu glede na celotno množico istopomenskih atributov. TF-IDF utež besede w v atributu a je definirana kot:

$$TFIDF(w, a) = TF(w, a) * IDF(w), \tag{2.11}$$

kjer s $TF(w, a)$ označimo frekvenco pojavljanja besede w v atributu a, z $IDF(w)$ pa označimo obratno vrednost pojavljanja besede w glede na množico vseh opazovanih atributov (glej enačbo 2.12).

$$TF(w, a) = \frac{n_{w,a}}{\sum_{w' \in a} n_{w',a}}, IDF(w) = \log \frac{|R|}{|a_i \in R.a \wedge w \in a_i|}. \tag{2.12}$$

Z $n_{w,a}$ označimo število pojavitev besede w v atributu a, $|R|$ pa je velikost množice vseh opazovanih referenc. Zaradi enostavnejšega primerjanja definiramo normalizirano vrednost uteži posamezne besede w v atributu a kot:

$$weight(w, a) = \frac{TFIDF(w, a)}{\sqrt{\sum_{w' \in a} TFIDF(w', a)^2}}. \tag{2.13}$$

Na podlagi tega pa lahko definiramo metriko za primerjanje podobnosti atributov Cos TF-IDF [10, 26] kot:

$$CosTFIDF(a_i, a_j) = \sum_{w \in a_i \cap a_j} weight(w, a_i) \times weight(w, a_j). \tag{2.14}$$

Soft TF-IDF primerjava

Slabost primerjalne metrike Cos TF-IDF je v tem, da v primerjavo všteje le besede, ki so identične v obeh primerjanih atributih. Problem nastane takrat, ko primerjamo atributa, ki sicer nimata nobene besede popolnoma enake, je pa večina besed zelo podobnih. Če bi tak par atributov primerjali z metriko Cos TF-IDF bi dobili podobnostno vrednost 0, kar pa v tem primeru ni ravno ustrezno. Določena stopnja podobnosti med opazovanima atributoma vendar vseeno obstaja. V ta namen so Cohen in njegovi sodelavci [10] vpeljali izboljšano metriko imenovano Soft TF-IDF, ki je predstavljena v nadaljevanju. Bistvo metrike je, da v primerjavo ne všteje le identičnih besed, temveč tudi pare besed, med katerimi obstaja določena podobnost. Množico parov podobnih besed primerjanih atributov definiramo kot:

$$W(\theta, a_i, a_j) = \{(w_1, w_2) \mid w_1 \in a_i \ \wedge \ w_2 \in a_j : sim_A(w_1, w_2) > \theta \wedge sim_A(w_1, w_2) = max(\{sim_A(w_1, w') \mid w' \in a_j\})\}, \quad (2.15)$$

kjer s θ označimo mejo nad katero primerjani besedi smatramo za medsebojno podobni. Podobnost med besedami izračunamo z uporabo katerekoli sekundarne atributne metrike sim_A, ki je primerna za primerjavo dveh znakovnih nizov. Tako opisano množico sestavljajo medseboj najbolj podobni pari besed, katerih podobnost je nad določeno mejo θ. Hkrati pa mora veljati tudi, da je poljubna beseda naenkrat del največ enega para. Na ta način preprečimo, da bi določeno besedo pri izračunu primerjave upoštevali večkrat in umetno povečevali podobnost med primerjanima atributoma. Soft TF-IDF je tako na podlagi definirane množice parov podobnih besed podana kot:

$$SoftTFIDF(a_i, a_j) = \sum_{(w_1, w_2)} weight(w_1, a_j) \times weight(w_2, a_i) \times sim_A(w_1, w_2). \quad (2.16)$$

Od metrike Cos TF-IDF se torej razlikuje v tem, da v primerjavo všteje vse pare podobnih besed in da vsako vrednost primerjave para besed uteži z njuno medsebojno podobnostjo. Vkolikor primerjamo atribute z enakimi besedami, bo ocena metrike Soft TF-IDF kar enaka oceni metrike Cos TF-IDF. V nasprotnem primeru pa bo ocena zmanjšana za faktor podobnosti neenakih besed, ki so v množici parov.

2.2.2 Relacijske metrike

V primerih, ko imamo med referencami podane relacije, lahko tudi te učinkovito uporabimo pri problemu razločevanja entitet. V ta namen je bilo razvitih

mnogo relacijskih metrik. V nadaljevanju nekatere od njih opišemo in tudi uporabimo pri razvoju sistema za združevanje omrežij.

Nekaj izmed opisanih metrik za ugotavljanje podobnosti med gradniki izkorišča soseščino. Pri našem primeru združevanja omrežij pa razločevanje izvajamo nad vsemi tipi gradnikov omrežja; tako vozlišči kot tudi povezavami. Zato najprej podajmo definicijo soseščine opazovanega gradnika r:

$$Nbr(r) = \begin{cases} \cup_{e \in r.E}\{e.V\}, & r \in V \\ r.V, & r \in E \end{cases} \tag{2.17}$$

kjer z $e.V$ označimo množico krajišč (vozlišč) povezave e. Z $v.E$ pa označimo množico povezav, ki imajo krajišče v vozlišču v.

V primeru, ko je r vozlišče, njegovo soseščino torej sestavljajo kar vsa vozlišča, ki so z opazovanim gradnikom r povezana preko neke povezave. Takšno pojmovanje soseščine za vozlišča je standardno in na podoben način bi lahko definirali tudi soseščino povezav. Vendar smo se v našem primeru odločili za drugačen pristop. Soseščino opazovane povezave definiramo kot množico vozlišč, ki jih ta povezuje. S tako definicijo soseščine povezave je primerjava slednjih tudi bolj smiselna. V primeru primerjave dveh povezav nas namreč zanima predvsem ali primerjani povezavi povezujeta ista vozlišča in ne, ali so povezave, ki izhajajo iz krajišč opazovanih povezav iste.

Skupni sosedi

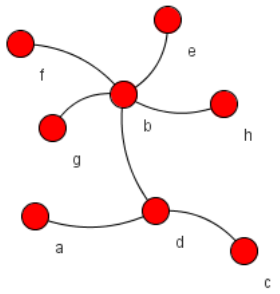
Najenostavnejši pristop za merjenje relacijske podobnosti med dvema gradnikoma omrežja predstavlja primerjava soseščin opazovanih referenc. Ideja tega pristopa je v tem, da imajo reference, ki predstavljajo iste entitete, podobne soseščine. Podobnost le-te pa ugotovimo s preštevanjem skupnih elementov v obeh soseščinah. Metrika je tako formalno podana kot:

$$CommonNeighbours(r_i, r_j) = \frac{1}{K} \times |Nbr(r_i) \bigcap Nbr(r_j)|. \tag{2.18}$$

Paziti moramo, da je parameter K dovolj velika konstanta, saj je le tako lahko ocena vedno manjša od 1 za vse možne primerjave gradnikov. To nam omogoča, da rezultat te metrike lahko primerjamo z rezultati katerih koli drugih metrik, ki vrnejo vrednosti z intervala $[0, 1]$.

Jaccardov koeficient

Pomanjkljivost metrike Skupni sosedi je normalizacijska konstanta K, ki je enaka za vsa možna primerjanja parov. Za primer poglejmo situacijo, ki je



Slika 2.4: Primer slabega delovanja metrike *Skupni sosedi*.

prikazana na sliki 2.4. Opazujemo gradnika a in b. Oba imata enako število skupnih sosedov z gradnikom c; to je d. Potemtakem sta oba gradnika a in b enako podobna z gradnikom c glede na metriko Skupni sosedi. Konkreten prikazan primer pa prikazuje situacijo, kjer so vsi sosedje gradnika a skupni s sosedi gradnika c, medtem ko ima gradnik b zelo veliko soseščino in je skupnih sosedov z gradnikom c relativno malo. V takem primeru želimo, da je podobnost med a in c večja kot med b in c, saj je verjetnost, da najdemo skupne sosede pri gradnikih z veliko soseščino večja.

V ta namen sta avtorja [5] opisala metriko, ki je osnovana na osnovi *Jaccardovega indeksa* za merjenje podobnosti in diverzitete med množicami. Metrika je formalno podana kot:

$$JaccardCoeff(r_i, r_j) = \frac{|Nbr(r_i) \bigcap Nbr(r_j)|}{|Nbr(r_i) \bigcup Nbr(r_j)|}. \tag{2.19}$$

Pri tej metriki normalizacijsko konstanto K nadomesti unija soseščin opazovanih gradnikov. Na ta način se odpravi opisana pomanjkljivost prejšnje metrike, hkrati pa se ohrani značilnost, da metrika vrne normaliziran rezultat.

Adamic-Adarjeva primerjava

Obe opisani relacijski metriki, Skupni sosedi in Jaccardov koeficient, reference v soseščinah dojemata kot enako pomembne. To pa, kot bomo videli na naslednjem primeru, ni vedno zaželeno. Če je opazovana referenca povezana z veliko različnimi referencami, potem njena prisotnost v neki soseščini ni tako pomembna kot prisotnost reference, ki je povezana z malo različnimi referencami. Ta ideja je zelo podobna oceni IDF v TF-IDF metriki (glej razdelek 2.2.1). Opisano idejo sta uporabila tudi Adamic in Adar v [1], kjer sta definirala

metriko za ugotavljanje podobnosti med spletnimi stranmi na osnovi skupnih lastnosti:

$$\sum_{skupna\ lastnost\ f} \frac{1}{log(\text{pogostost pojavljanja } f)}. \tag{2.20}$$

Bhattacharya in Getoor v [5] sta njuno metriko adaptirala za problem razločevanja entitet. Metrika je tako formalno definirana kot:

$$AdamicAdar(r_i, r_j) = \frac{\sum_{r \in Nbr(r_i) \bigcap Nbr(r_j)} u(r)}{\sum_{r \in Nbr(r_i) \bigcup Nbr(r_j)} u(r)}, \tag{2.21}$$

kjer z $u(r)$ označimo edinstvenost reference in je definirana kot:

$$u(r) = \frac{1}{log(|Nbr(r)|)}. \tag{2.22}$$

Velikost soseščine opazovanega elementa se torej uporabi kot mera za ugotavljanje edinstvenosti reference, pri čemer večja soseščina pomeni manjšo edinstvenost. Tudi ta metrika vrne normalizirano vrednost med 0 in 1. Omenimo še, da so prilagoditve osnovne Adamic-Adarjeve metrike uporabljene tudi na drugih področjih. Odlično se je na primer izkazala v aplikacijah predvidevanja povezav (ang. *link prediction*) v socialnih omrežjih.

Adamic-Adarjeva primerjava z oceno dvoumnosti

Avtorja v [5] navajata, da se uporaba velikosti soseščine, kot sredstva za merjenje edinstvenosti, ni izkazala za dobro. Pri problemu razločevanja entitet namreč ne poznamo resnične soseščine za vsako referenco. Resnično sosedstvo je možno pridobiti šele, ko je proces razločevanja uspešno izveden do konca. To pomeni, da je uporaba velikosti soseščine, kot mere za napovedovanje edinstvenosti reference, v aplikacijah razločevanja entitet pristranska v vseh vmesnih fazah razen na koncu.

Kot alternativo zato predlagata, da se v sistemih razločevanja entitet za merjenje edinstvenosti reference uporabi ocena dvoumnosti le-te:

$$u(r) = \frac{1}{log(Amb(r))}, \tag{2.23}$$

kjer je $Amb(r)$ ocena dvoumnosti opazovane reference r. Kot dvoumne reference označujemo tiste, ki se ujemajo v vseh atributih, vseeno pa predstavljajo različne entitete. Dvoumnost se sicer lahko ocenjuje na različne načine,

v splošnem pa ta ocena pomeni verjetnost, da dve naključno izbrani referenci z enakimi atributi, predstavljata različni entiteti. Bolj podrobno problem dvoumnosti opišemo v razdelku 3, kjer predstavimo tudi možen pristop ocenjevanja dvoumnosti.

Sosedstva višjega reda

Podobnost dveh referenc v omrežju lahko opazujemo tudi z vidika primerjanja sosedstev višjega reda. Sosedstvo reda n je definirano kot množica vseh tistih referenc, ki so od opazovane reference oddaljeni največ n korakov. Glede na učinek majhnega sveta (glej razdelek 2.1.3) bi pri $n = 6$ dobili soseščino, ki bi vsebovala vse gradnike[3] celotnega omrežja. Zato se običajno pri problemu razločevanja entitet omejimo na $n = 2$; torej sosedstva drugega reda in kot navajata avtorja [5], so se le-ta izkazala za dobra pri omenjenem problemu. Metrika je formalno podana kot:

$$Path2Sim(r_i, r_j) = sim_R(Nbr(r_i)^2, Nbr(r_j)^2), \tag{2.24}$$

kjer z $Nbr(r)^2$ označimo soseščino drugega reda, s sim_R pa označimo katerokoli relacijsko metriko, ki za primerjavo uporablja soseščini podanih referenc.

Metrika naključni sprehodi

Naključni sprehodi so vrsta gibanja, pri katerem enakomerno izberemo dolžino našega koraka in smer, v katero se bomo pomaknili. V literaturi jih pogosto poimenujejo tudi '*drunkard's walk*' ali pa '*Lévyev polet*' po francoskem matematiku Paulu Pierreu Lévyu. Pri slednjih dveh gre dejansko za posebni različici naključnega sprehoda po grafu.

Metriko naključni sprehodi označimo z $RndWalk(r_i, r_j)$. Definiramo pa jo na naslednji način. Izvedemo M naključnih sprehodov po omrežju v največ K korakih. Vsak sprehod začnemo v vozlišču r_i. Nato pa na vsakem koraku enakomerno izberemo naslednji gradnik. V primeru, da sta primerjana gradnika vozlišči, izberemo vozlišče iz množice njegovih sosedov. V primeru, da sta primerjana gradnika povezavi, pa izberemo eno izmed povezav, ki se začenjajo v krajiščih trenutne povezave. Postopek ponavljamo dokler, bodisi ne dospemo v r_j, bodisi presežemo število dovoljenih korakov K. V primeru, da gradnika r_j ne dosežemo v K korakih, je podobnost med opazovanima gradnikoma 0 za trenuten sprehod. V nasprotnem primeru pa je podobnost enaka produktu

[3]Soseščino bi glede na našo opredelitev soseščine sestavljala kar vsa vozlišča omrežja ne glede na to ali bi bili primerjani referenci povezavi ali vozlišči.

uteži tipov povezav, po katerih smo prispeli od r_i do r_j. Ko izvedemo vse naključne sprehode, rezultate posameznih sprehodov seštejemo in povprečimo, kar predstavlja končno podobnost med opazovanima gradnikoma:

$$RndWalk(r_i, r_j) = \frac{RndWalk_1(r_i, r_j) + \cdots + RndWalk_m(r_i, r_j)}{M}, \qquad (2.25)$$

kjer je $RndWalk_l(r_i, r_j)$ rezultat podobnosti med opazovanima gradnikoma iz l-tega sprehoda. Tipe povezav običajno utežimo z vrednostmi med 0 in $\frac{1}{2}$ in rezultat vsakega sprehoda tudi delimo z maksimalno vrednostjo uteži, saj tako zagotovimo, da je rezultat metrike normaliziran in primerljiv z rezultati ostalih metrik.

Predlagana metrika torej upošteva tudi različne tipe povezav, medtem ko ostale opisane relacijske metrike tega ne upoštevajo. To je v določenih primerih lahko pomembno, kar prikažemo na naslednjem primeru. Poglejmo omrežje oseb, ki so med seboj povezane z dvema tipoma povezav. Prvi tip povezav označuje 'prijateljstvo', drugi tip povezav pa označuje relacijo 'poslovni partner'. Intuicija nam pravi, da opazovani referenci bolj verjetno predstavljata isto entiteto, če sta povezani prek povezav 'prijateljstvo', kot pa prek povezav 'poslovni partner'.

Prav tako pomembna lastnost opisane metrike pa je, da za primerjavo ne izkorišča le neposredne soseščine opazovanih gradnikov, temveč celotno omrežje. Tako s to metriko lahko merimo podobnost tudi med referencami, ki so narazen več kot tri povezave in nimajo deljene soseščine.

Negativne omejitve iz relacij

Do sedaj opisanim metrikam za ugotavljanje podobnosti med dvema gradnikoma v omrežju je skupno to, da merijo le dokaze, da dve opazovani referenci pripadata isti entiteti. Nobena izmed opisanih metrik pa ne vključuje preverjanja, ali morda primerjani referenci ne pripadata isti entiteti. Iz relacijskih podatkov namreč lahko izhajajo tudi negativne omejitve za združevanje dveh referenc v isto entiteto. V mnogih domenah na primer velja, da reference, ki jih povezuje ista povezava oziroma hiper-povezava, ne morejo pripadati isti entiteti.

Kot realen zgled lahko navedemo primer iz domene znanstvenih člankov, kjer so članki predstavljeni s hiper-povezavami, avtorji pa z vozlišči teh hiper-povezav. *M. Faloutsos*, *P. Faloutsos* in *C. Faloutsos* so avtorji nekega istega članka in kljub temu, da so vrednosti atributov teh referenc enake, vsaka od teh referenc opisuje drugo resnično entiteto. Za bibliografske domene podatkov in

domene s podobnimi lastnostmi lahko dodamo omenjeno omejitev. V splošnem pa imamo seveda lahko poljubno veliko množico omejitev, ki izhajajo iz relacij in katerim je potrebno zadostiti pri razločevanju entitet. Pri tem pa je potrebno omeniti še, da so tovrstne omejitve odvisne od načina predstavitve podatkov z omrežjem. Zato je pri določanju tovrstnih omejitev nujna prisotnost poznavalca obravnavane domene.

Poglavje 3

Sistem za združevanje podatkovnih omrežij

V razdelku 2 smo predstavili teoretično podlago problema združevanja omrežij; opisali smo nekatere značilnosti omrežij in predstavili metrike, ki se pogosto uporabljajo pri razločevanju entitet. V naslednjem razdelku pa opišemo predlagan sistem za reševanje omenjenega problema. Sestavljen je iz štirih glavnih komponent (slika 3.1):

Predprocesiranje je komponenta sistema, v kateri nastavimo potrebne parametre sistema. Le-te določimo na podlagi izračunanih statistik iz vhodnega nabora podatkov.

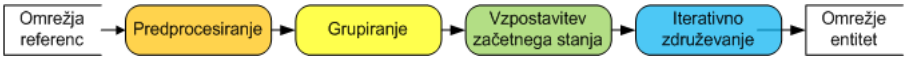
Grupiranje zmanjša časovno zahtevnost sistema. Množico referenc vseh obdelovanih omrežij razvrstimo v skupine med seboj podobnih referenc. Na ta način omejimo število vseh primerjav v sistemu in posledično zmanjšamo časovno zahtevnost.

Vzpostavitev začetnega stanja je komponenta sistema, v kateri obdelovana omrežja medsebojno povežemo tako, da združimo tiste gradnike za katere predpostavimo, da očitno predstavljajo isto entiteto.

Iterativno združevanje pa je kompononenta v kateri se odvija glavni proces zlivanja obdelovanih omrežij v enotno omrežje.

Vsaka izmed naštetih komponent sistema je podrobno opisana v nadaljevanju.

Omenili smo tudi (glej razdelek 1.3), da je eden od zadanih ciljev tega dela, narediti sistem samoprilagodljiv glede na podan nabor podatkov. To se izkaže za zahtevno nalogo, saj na kakovost same izvedbe združevanja vplivajo različni dejavniki:



Slika 3.1: Komponente sistema za združevanje omrežij.

Dvoumnost

Z izrazom dvoumne označujemo tiste reference, ki se ujemajo v vseh atributih, vseeno pa predstavljajo različne entitete. Za boljšo predstavo si poglejmo realen primer, ki nazorno predstavlja opisan problem. Podani imamo referenci 'A. Novak' in 'A. Novak' z istim naslovom bivališca. Denimo, da označujeta zakonski par Andreja in Ano Novak. Atributi omenjenih referenc so torej enaki, pa vendar le-ti predstavljata različni resnični entiteti. Zbirke podatkov, kjer je frekvenca pojavljanja podobnih primerkov velika, pa označimo kot dvoumne zbirke podatkov.

Sistemi za razločevanje entitet, v katerih razločevanje temelji na primerjavi atributov referenc, imajo nad zbirkami podatkov z visoko dvoumnostjo lahko velike težave. Združujejo namreč reference z enakimi atributi, ki pa vseeno pripadajo različnim entitetam. V našem sistemu torej potrebujemo mehanizem, ki bo omogočal dobro oceniti stopnjo dvoumnosti v podanih podatkih in bo znal to vrednost tudi ustrezno uporabiti. S tem mislimo predvsem to, da se bo model združevanja sposoben ustrezno prilagoditi podanim podatkom.

Izbira in utežitev primerjalnih metrik

V razdelkih 2.2.1 in 2.2.2 so opisane nekatere primerjalne metrike, ki se pogosto uporabljajo pri problemih razločevanja entitet. Če za trenutek vzamemo pod drobnogled atributne primerjalne metrike lahko ugotovimo, da nekatere bolje delujejo nad določenimi tipi podatkov, druge nad drugimi. Tako sta metriki Cos TF-IDF in Soft TF-IDF bolj primerni za primerjanje daljših, večbesednih atributov (recimo naslovov ali opisov), medtem ko so metrike Levenshtein, Jaro in Jaro-Winkler bolj primerne za primerjanje krajših eno- ali dvo-besednih atributov (recimo imen in priimkov) [10]. Do podobnega zaključka lahko pridemo tudi, če med seboj primerjamo značilnosti relacijskih metrik (glej razdelek 2.2.2), kjer smo pomanjkljivosti in prednosti vsake metrike navedli že ob njihovih opisih.

Običajno metrike za primerjavo referenc določi domenski ekspert. Če je izbranih več različnih metrik, pa je potrebno pomembnost vsake tudi utežiti. V literaturi [5, 8, 9, 12] zasledimo medseboj podobne modele primerjave, ki so

definirani kot linearna kombinacija poljubne atributne in relacijske metrike:

$$sim(r_i, r_j) = \alpha \times sim_A(r_i, r_j) + \beta \times sim_R(r_i, r_j). \quad (3.1)$$

V splošnem bi enačbo lahko razširili na linearno kombinacijo poljubnega števila metrik:

$$sim(r_i, r_j) = \alpha \times sim_1(r_i, r_j) + \beta \times sim_2(r_i, r_j) + \gamma \times sim_3(r_i, r_j) + \ldots \quad (3.2)$$

V takih modelih bi potrebovali odločitev domenskega eksperta, najprej o izbiri posameznih metrik in nato še o določitvi parametrov α, β, $\gamma \ldots$, s katerimi bi utežili pomembnost posamezne metrike. Zaradi velikega števila omenjenih parametrov, ki so lahko med seboj odvisni, je to težak problem, saj običajno ne poznamo korelacije med njimi. Tako se običajno utežitev metrik nastavlja le na podlagi intuicije, kar pa lahko vodi v slabe rezultate razločevanja.

V našem sistemu želimo, da sistem sam uteži metrike za razločevanje, glede na podane podatke.

Določitev meje združevanja

V sistemih za razločevanje entitet obstaja še en pomemben parameter, ki lahko močno vpliva na kakovost združevanja. To je parameter meja (ang. *threshold*), s katerim določimo najmanjšo vrednost podobnosti para referenc, nad katero ju še združimo v enoten gradnik (predpostavimo, da predstavljata isto entiteto). Ta parameter je neposredno odvisen od izbire in utežitve metrik, zato je nastavljanje le-tega prav tako problematično kot nastavitev parametrov iz prejšnje skupine.

Pri razvoju predlaganega sistema omenjene dejavnike upoštevamo in razvijemo sistem, ki se je sposoben podanim podatkom prilagajati. Na ta način želimo doseči boljše rezultate in razbremeniti domenskega eksperta pri nastavljanju parametrov sistema. Na tem mestu je potrebno tudi omeniti, da zato predlagan sistem za delovanje potrebuje manjši označen začetni nabor podatkov.

3.1 Predprocesiranje

Predprocesiranje je komponenta sistema, v kateri iz podanega nabora podatkov izračunamo ustrezne statistike, na podlagi katerih določimo vrednosti parametrov sistema. Za delovanje sistema je potrebno:

- vsakemu entitetnemu tipu določiti atribut, na podlagi katerega, bo izvedeno grupiranje referenc v skupine,
- izbrati utežitev atributov za vsak entitetni tip in
- oceniti dvoumnost referenc.

Izbira atributa za grupiranje

Grupiranje je komponenta sistema, katere namen je zmanjšati časovno zahtevnost celotnega procesa združevanja omrežij. To izvedemo tako, da množico referenc vseh omrežij razvrstimo v skupine podobnih referenc, na podlagi izbranega atributa za grupiranje. V nadaljevanju se primerjave izvajajo le med tistimi referencami, ki so v istih skupinah. Bolj natančen opis grupiranja sledi v razdelku 3.2, tu pa pozornost posvetimo predvsem izbiri atributa, na podlagi katerega reference razvrščamo v omenjene skupine.

Vsakemu entitetnemu tipu za grupiranje določimo tisti atribut, ki ima največ različnih vrednosti. Na ta način zagotovimo, da bo število skupin po končanem grupiranju čim večje, v skupinah pa bo čim manjše število referenc. Če bi alternativno za grupiranje izbrali atribut, ki ima malo različnih vrednosti, bi posledično dobili manjše število skupin z velikim številom referenc v vsaki skupini. Sistem bi v nadaljevanju opravil ogromno število primerjav in časovna zahtevnost celotnega postopka bi ostala relativno nespremenjena. Smiselnost takega pristopa leži v dejstvu, da je običajno le 1% primerjav takih parov referenc, ki dejansko predstavljajo isto entiteto.

Slabost omenjenega pristopa pa je ta, da reference razvrščamo le na podlagi atributov. V primerih, ko združujemo omrežja, kjer je veliko šuma, je ta pristop lahko problematičen. Tedaj atributi namreč niso dovolj informativni in posledično lahko reference, ki predstavljajo iste entitete, razvrščamo v različne skupine.

Utežitev atributov

Vsakemu entitetnemu tipu je potrebno določiti tudi utežitev atributov. Pri primerjavi referenc na podlagi atributov je namreč pomembno, v katerih atributih se referenci ujemata. Za ponazoritev si poglejmo naslednji primer. Če se poljubni referenci ujemata denimo v atributu ‘spol’, ki lahko zavzame le dve vrednost (‘moški’ in ‘ženska’), je to veliko manj informativno kot, če se referenci ujemata denimo v atributu ‘ime’. Za slednjega je očitno, da lahko zavzame znatno več vrednosti kot ‘spol’. To je razlog, zakaj je potrebno atribute utežiti.

V predlaganem sistemu tako atribute z več različnimi vrednostmi smatramo kot bolj pomembne in jim zato dodelimo tudi večjo utež.

Utež atributu dodelimo na podlagi števila različnih vrednosti, ki jih ta zavzame. Utež atributa a_i je tako definirana na naslednji način:

$$attrWeight(a_i) = \frac{\text{število različnih vrednosti } a_i}{\text{število vseh vrednosti } a_i}. \tag{3.3}$$

Po izračunu uteži atributov slednje še normaliziramo tako, da je vsota uteži atributov določenega entitetnega tipa 1. Normalizirane vrednosti uteži nato uporabimo pri izračunu podobnosti med dvema referencama kot

$$\begin{aligned} sim_A(r_1, r_2) = {} & attrWeight(a_1) \times sim_A(r_1.a_1, r_2.a_1) + \\ & + attrWeight(a_2) \times sim_A(r_1.a_2, r_2.a_2) + \\ & + \cdots + \\ & + attrWeight(a_n) \times sim_A(r_1.a_n, r_2.a_n), \end{aligned} \tag{3.4}$$

kjer je sim_A poljubna metrika za primerjavo atributov (glej razdelek 2.2.1).

Ocena dvoumnosti

Že v poglavju 2.2.1 smo omenili, da gre pri določanju stopnje dvoumnosti v splošnem dejansko za ugotavljanje verjetnosti, da opazovani referenci z enakimi atributi pripadata različnim entitetam. Obstaja sicer veliko modelov za ocenjevanje slednjega, Bhattacharya in Getoor pa v [5] predlagata naslednji model:

$$Amb(r) = \frac{|\sigma_{R_b.a_i = r.a_i}(R_b)|}{|R_b|}, \tag{3.5}$$

kjer je $|\sigma_{R_b.a_1 = r.a_1}(R_b)|$ število referenc, ki imajo vrednost opazovanega atributa a_i enako vrednosti atributa opazovane reference r. Seveda pri oceni upoštevamo le tiste reference, ki so enakega tipa kot opazovana referenca, saj bi bilo nesmiselno upoštevati celotno množico referenc. Za omenjeno shemo ocenjevanja dvoumnosti pa predlagata tudi izboljšavo. Ocena se namreč močno izboljša, če imamo na voljo dodaten atribut a_j. Tako definiramo novo formulo za oceno dvoumnosti

$$Amb(r) = \frac{|\sigma_{R_b.a_i = r.a_i \wedge R_b.a_j = r.a_j}(R_b))|}{|R_b|}, \tag{3.6}$$

kjer je $|\sigma_{R_b.a_i = r.a_i \wedge R_b.a_j = r.a_j}(R_b))|$ število referenc, ki se z opazovano referenco r ujemajo v izbranih atributih a_i in a_j. To nam da boljšo oceno dvoumnosti

opazovane reference kot prvi pristop, saj je ocena bolj natančna. Če slednji model ponazorimo na primeru atributov 'ime' in 'priimek' to pomeni, da dvoumnost opazovane reference ocenjujemo s štetjem referenc, ki imajo enako vrednost imena in priimka kot opazovana referenca. Kadar imamo na voljo več medseboj neodvisnih atributov lahko ta pristop seveda razširimo na več kot dva atributa za pridobivanje še boljše ocene dvoumnosti.

Kadar imamo na voljo dva atributa, dvoumnost v našem sistemu ocenjujemo z uporabo slednjega pristopa, sicer pa po prvem (glej enačbo 3.5). Za atributa a_i in a_j izberemo atributa z največjima utežema. Tako si zagotovimo, da je ocena kar najbolj natančna. Če bi dvoumnost ocenjevali z uporabo atributov z nizkimi utežmi, bi bila ocena zelo nenatančna oziroma lahko že kar nepravilna. To lahko ponazorimo na primeru atributa 'spol' in ugotovimo, da ta ni primeren za ugotavljanje dvoumnosti. Taka ocena dvoumnosti bi bila namreč za vse reference enaka[1], kar pa je očitno slabo.

Z omenjenim modelom pa dobimo le numerično oceno dvoumnosti. Še vedno je namreč potrebno določiti oznako ali je obravnava referenca dvoumna ali ne. Za določitev slednjega pa predlagamo naslednji model:

$$IsAmb(r) = \begin{cases} 1, & \text{če je } Amb(r) > \frac{1}{|T_b|} \\ 0, & \text{sicer} \end{cases} . \tag{3.7}$$

Izračunano vrednost dvoumnosti $Amb(r)$ torej primerjajmo z obratno vrednostjo števila entitet (določenega tipa). V primeru, da je ocena dvoumnosti večja od opazovanega razmerja, referenco razglasimo za dvoumno. Za določanje dvoumnosti torej potrebujemo število resničnih entitet v podanem naboru podatkov. V našem primeru to ocenimo kar na podlagi označenega dela nabora podatkov oziroma podane učne množice.

Za ugotavljanje števila resničnih entitet pa obstajajo tudi drugi pristopi. Za primer lahko navedemo metodo Latentna Dirichletova alokacija - LDA (ang. *Latent Dirichlet Allocation*) [6], katera omogoča ocenitev števila resničnih entitet brez označenega začetnega nabora podatkov; torej neposredno iz podane množice podatkov. Omenjene metode v našem primeru nismo uporabili, saj je ocena pridobljena iz označenih podatkov bolj enostavna za implementacijo in verjetno tudi bolj točna.

[1] Enaka bi bila za vse reference moškega spola in vse reference ženskega spola.

3.2 Grupiranje

Grupiranje je komponenta sistema, katere namen je zmanjšati časovno zahtevnost celotnega procesa združevanja omrežij. To dosežemo tako, da zmanjšamo število primerjav referenc v celotnem procesu združevanja. V ta namen reference obdelovanih omrežij razvrstimo v skupine. V vsaki skupini so le tiste reference, ki so si med seboj podobne in so dejansko potrebne nadaljnjega analiziranja. Primerjave v nadaljevanju izvajamo le med temi, ki so v istih skupinah in ne med vsemi pari referenc. Tako znatno zmanjšamo število primerjav in posledično tudi časovno zahtevnost celotnega procesa združevanja. Nepraktično in zamudno je izvajati primerjave med vsemi pari referenc predvsem zato, ker večina parov gotovo ne predstavlja iste entitete. V razdelku 3.1 smo opisali, na kakšen način določimo atribut za grupiranje, v nadaljevanju pa pojasnimo, kako ta atribut uporabimo pri samem razvrščanju referenc v skupine.

Za razdelitev referenc v skupine izvedemo enostaven sprehod skozi množico referenc. Vsako referenco primerjamo s predstavnikom vsake skupine tako, da primerjamo njuna atributa za grupiranje s poljubno atributno metriko. Če je podobnost nad določeno mejo ($meja_{grup}$), referenco dodelimo v trenutno opazovano skupino. Predstavnik skupine je referenca, ki je bila prva dodana v skupino. Pomembno je, da je primerjava reference s skupino hitra, zato izvedemo primerjavo le z eno (predstavnikom skupine) in ne z vsemi referencami v skupini. Če reference ne dodamo v nobeno skupino, kreiramo novo in kot predstavnika novonastale skupine določimo trenutno opazovano referenco. Časovna zahtevnost opisanega pristopa je *O(nm)*, kjer je n število referenc in m število skupin.

Obstajajo pa tudi drugi možni pristopi. Referenco bi lahko na primer primerjali s povprečjem vseh referenc v skupini. Pri tem bi bilo potrebno pri vsaki primerjavi izračunavati povprečje le-teh. Težava pa bi nastala tudi pri izračunavanju povprečja različno dolgih nizov. Za ta pristop se tako nismo odločili.

Potrebno je opozoriti, da pri postopku grupiranja ne delamo nikakršnih razlik med gradniki posameznih omrežij, kar pomeni, da se po končanem grupiranju lahko v isti skupini nahajajo tudi reference istega omrežja. Na ta način dosežemo, da bodo med celotnim procesom združevanja omrežij združeni tudi duplikati v posameznih omrežjih. Slednje bi lahko napravili tudi v ločenem procesu, vendar bi bila taka rešitev časovno potratnejša.

Opisan pristop grupiranja lahko močno vpliva tudi na kakovost celotnega procesa združevanja omrežij. Če dve referenci, ki predstavljata isto entiteto,

razporedimo v različni skupini, to vpliva na končni rezultat. Opazovanega para referenc namreč nikoli ne primerjamo in posledično tudi ne združimo.

3.3 Vzpostavitev začetnega stanja

Posebno pozornost je potrebno nameniti inicializaciji začetnega stanja za algoritem iterativnega združevanja, ki je opisan v razdelku 3.4. Bistvo tega algoritma je, da odločitve o združitvi posameznih parov referenc niso neodvisne ena od druge, temveč so skupinske. To pomeni, da za primerjavo uporabljamo tako atributne kot tudi relacijske metrike. Pri problemu združevanja omrežij v eno omrežje se pojavi specifična težava - omrežja so namreč nepovezana. Primerjani pari referenc iz različnih omrežij tako nimajo deljenega sosedstva, niti med njimi ne obstaja nobena pot. Uporaba relacijskih metrik, opisanih v razdelku 2.2.2, bi bila zato nesmiselna in združevanje bi tako potekalo le na osnovi atributnih metrik (vsaj v začetnih iteracijah iterativnega združevanja). Takšno združevanje pa je lahko slabo, še posebej v dvoumnih zbirkah podatkov.

Potrebno je torej vzpostaviti povezave med posameznimi omrežji, da dobimo enotno in predvsem povezano začetno omrežje. Na ta način zagotovimo, da med vsakim parom referenc, ki potencialno opisuje isto entiteto, obstaja neka pot in posledično morda tudi soseščina. Tako imamo že v začetnih ponovitvah iterativnega združevanja, možnost uporabiti tudi relacijske metrike pri primerjavi referenc in na ta način združevanje ni odvisno le od primerjave atributov.

Omrežja med seboj povežemo v enotno omrežje tako, da izvedemo primerjavo vseh parov referenc, ki se nahajajo v istih skupinah (glej razdelek 3.2). Vsak par referenc, ki *zagotovo* predstavlja isto entiteto, združimo v skupen gradnik, imenovan grozd referenc. Ostalih referenc ne združujemo, temveč vsako izmed njih dodelimo v svoj grozd. Omenimo, da od tu dalje ne govorimo več o referencah v omrežjih, temveč o grozdih referenc, kljub temu, da določeni grozdi vsebujejo le po eno referenco. Grozd označimo s c, množico referenc v grozdu pa s $c.R = \{r_i\}$.

Potrebno pa je še pojasniti, na kakšen način ugotavljamo, ali opazovani referenci *zagotovo* opisujeta isto entiteto. Pri tem procesu je ključna natančnost združevanja, saj odločitev o združitvi referenc v skupne grozde ni mogoče razveljaviti, hkrati pa te v nadaljevanju lahko vplivajo tudi na ostale.

3.3.1 Enostavna shema

Pri enostavni shemi vsak par referenc med seboj primerjamo na osnovi atributov. Če se referenci natančno ujemata v vseh istopomenskih atributih, ju združimo v skupen grozd. Vendar pa ima ta shema veliko pomanjkljivost - dobro deluje le nad podatkovnimi zbirkami z majhno stopnjo dvoumnosti. Pri zbirkah z visoko stopnjo dvoumnosti, bi namreč v skupne grozde združevali tudi reference, ki dejansko predstavljajo različne entitete. To pa posledično lahko pomeni slabo kakovost delovanja celotnega sistema. Zato shemo lahko obogatimo tako, da v skupne grozde združujemo tiste pare referenc, ki niso dvoumne in se natančno ujemajo v vseh istopomenskih atributih. Vendar tudi tu nastane težava. Pri zbirkah podatkov z visoko dvoumnostjo bo malo parov referenc takih, ki ne bodo označene za dvoumne in bo posledično večina referenc dodeljena različnim grozdom. Slednje pomeni, da s tem pristopom ne bomo zadostili nalogi inicializacije začetnega stanja, saj bodo omrežja še vedno lahko nepovezana.

3.3.2 Relacijska shema

Problem prejšnje sheme lahko na enostaven način rešimo z uporabo primerjanja soseščin obravnavanih referenc. Skupne soseščine primerjani referenci gotovo ne bosta imeli, saj se nahajata v različnih omrežjih. Zato presek soseščin sestavimo tako, da kot skupne sosede upoštevamo vse tiste gradnike, ki se natančno ujemajo v vseh istopomenskih atributih in so hkrati sosedi ene izmed obravnavanih referenc. Referenci torej združimo v skupen grozd natanko takrat, ko se ujemata v vseh istopomenskih atributih in imata k skupnih sosedov glede na opisano definicijo. Parameter k pa pri tem definiramo kot:

$$k = \lfloor \frac{max(Amb(r_1), Amb(r_2))}{\frac{1}{|T_b|}} \rfloor. \tag{3.8}$$

Ta je torej odvisen od stopnje dvoumnosti opazovanih referenc - večja kot je stopnja dvoumnosti opazovanega para, več skupnih sosedov zahtevamo za združitev. Omenjena definicija parametra k pa na drugi strani omogoča tudi, da nedvoumne reference združujemo tudi, če skupne soseščine nimata.

3.4 Iterativno združevanje

Po končani inicializaciji začetnega stanja pride na vrsto glavni proces, v katerem iterativno združujemo podobne pare grozdov referenc in na ta način

podana omrežja dejansko zlivamo v enotno omrežje. Postopek iterativnega združevanja je prikazan v algoritmu 1.

Algoritem 1 Algoritem iterativnega združevanja

```
for all c_i, c_j that are similar do
    push {c_i, c_j, sim(c_i, c_j)} into Q based on descending order of sim(c_i, c_j)
end for

while Q not empty do
    simpair ← pull {c_i, c_j, sim(c_i, c_j)} from Q
    if c_i and c_j from simpair are same entity then
        join c_i and c_j
        Q ← update sim(·, ·) in Q
    end if
end while
```

Najprej identificiramo vse pare grozdov referenc, ki potencialno predstavljajo isto entiteto. Tak par grozdov imenujemo podobnostni par in je v algoritmu 1 označen s *simpair*. V postopku kot podobne dejansko označimo le tiste pare grozdov, katerih reference se nahajajo v istih skupinah (glej 3.2). Vsakemu paru pa izračunamo tudi njuno podobnostno vrednost kot:

$$sim(c_i, c_j) = sim_1(c_i, c_j) + sim_2(c_i, c_j) + sim_3(c_i, c_j) + \ldots, \tag{3.9}$$

pri čemer sta c_i in c_j grozda iz opazovanega para, $sim_i(c_i, c_j)$ pa je rezultat primerjave grozdov z i-to metriko. Pomembno je, da uporabimo tako atributne kot tudi relacijske metrike, saj uporaba različnih tipov metrik izboljša kakovost združevanja. Nato sledi iterativno združevanje parov grozdov.

Na vsakem koraku iterativnega združevanja iz prioritetne vrste vzamemo podobnostni par grozdov z najvišjo vrednostjo podobnosti. Če *presodimo* (pojem je razložen v razdelku 3.4.1), da grozda podobnostnega para predstavljata isto entiteto, ju združimo v enoten grozd. Nato popravimo še podobnostne vrednosti ostalim parom, saj je združitev opazovanega para grozdov lahko spremenila njihove podobnostne vrednosti. Ker bi bilo popravljanje podobnosti vsem parom na vsakem koraku zamudno, v naši implementaciji podobnostno vrednost ponovno izračunamo le sosedom pravkar združenega para, na katere ima združevanje tudi največji vpliv. Slabost te poenostavitve lahko pride do izraza predvsem pri uporabi metrik, ki za primerjavo ne izkoriščajo le najožje soseščine temveč tudi širšo. Za primer lahko navedemo metriki Sosedstva višjega reda in Metrika naključni sprehodi, kateri smo opisali že v razdelku 2.2.2.

Bistvo opisanega pristopa je v tem, da vsakega para grozdov ne obravnavamo neodvisno, temveč skupinsko, kar zagotovi uporaba relacijskih metrik. To pomeni, da pri kasnejših odločitvah upoštevamo tudi predhodne odločitve o združitvah posameznih grozdov. Zato je pomembno, da so predhodne odločitve pravilne, saj v nasprotnem primeru lahko napačne odločitve slabo vplivajo na kasnejše. To je razlog, da na vsakem koraku iterativnega združevanja obravnavamo par z največjo medsebojno podobnostjo. Za tega velja, da grozda najverjetneje predstavljata isto entiteto in združitev v enoten grozd bo v pomoč pri izdelavi odločitev v naslednjih iteracijah združevanja. Opisan pristop v literaturi [5, 12] zasledimo pod imenom skupinsko razločevanje (ang. *collective resolution*).

Časovna zahtevnost opisanega algoritma je $O(mk \log m)$, kjer je m število podobnostnih parov v prioritetni vrsti in k maksimalno število sosedov, ki jim je treba popraviti podobnostno vrednost na vsakem koraku. Pri tem smo predpostavili, da ima implementacija operacije *pull* nad prioritetno vrsto, časovno zahtevnost $O(\log m)$. Časovno zahtevnost izvedbe posodobitve podobnostne vrednosti nekega para in same primerjave grozdov pa smo pri tej oceni zanemarili, saj sta odvisni od izbire primerjalnih metrik in samega postopka odločanja o združitvi posameznih parov.

3.4.1 Odločitev o združitvi

Potrebno je še pojasniti na kakšen način v predlaganem sistemu izdelamo odločitev o združitvi posameznega para grozdov v skupen grozd. Kot smo opisali že v uvodnem delu razdelka 3, različni avtorji predlagajo pristop, pri katerem referenci ali grozda referenc združimo, kadar je njuna podobnostna vrednost nad določeno mejo. Le-ta pa se običajno izračuna kot utežena linearna kombinacija različnih izbranih metrik. Omenjen pristop utegne biti problematičen zaradi nastavljanja uteži posameznim metrikam ter določanja meje, saj parametri omenjenega modela niso neodvisni. Tako je kakovost razločevanja pri omenjenih modelih močno odvisna od primerne nastavitve omenjenih parametrov.

V našem sistemu namesto opisanega pristopa predlagamo model, pri katerem rezultat vsake metrike obravnavamo neodvisno od drugih. Definirajmo torej najprej funkcijo, ki iz rezultata vsake metrike določi ali opazovana grozda predstavljata isto entiteto ali ne:

$$d_i(c_1, c_2) = \begin{cases} +1, & sim_i(c_1, c_2) > meja_i \\ -1, & \text{sicer} \end{cases}. \tag{3.10}$$

Za vsako metriko posebej je tako potrebno določiti parameter $meja_i$. Le-te sicer lahko nastavi domenski ekspert, lahko pa jih nastavimo z učenjem nad učno množico podatkov. Opazujemo delovanje vsake izmed metrik in mejo nastavimo tako, da je število pravilnih napovedi nad učno množico največje. S tem pristopom torej metrike obravnavamo neodvisno eno od druge in ne kot linearno kombinacijo. Potrebujemo pa še način za izdelavo skupne odločitve o združitvi na podlagi množice napovedi vseh metrik $D = \{d_i\}$.

Najbolj osnoven pristop za kombiniranje napovedi večih različnih metrik v skupno odločitev je *večinsko glasovanje*. Za odločitev o paru grozdov torej seštejemo napovedi $d_i = \{-1, +1\}$ vseh obravnavnih metrik. Če je rezultat pozitiven, je torej večina metrik odločila, da obravnavan par grozdov resnično predstavlja isto entiteto. Slabost tega modela pa je, da napovedi vseh obravnavanih metrik smatra kot enako pomembne.

Slednje sicer lahko izboljšamo in namesto večinskega glasovanja uporabimo *uteženo glasovanje*, vendar tu naletimo na enak problem nastavljanja uteži kot prej. Bistveni problem glasovalnih sistemov pa je, da so neprilagodljivi glede na kontekst in podatke, s katerimi imamo opravka.

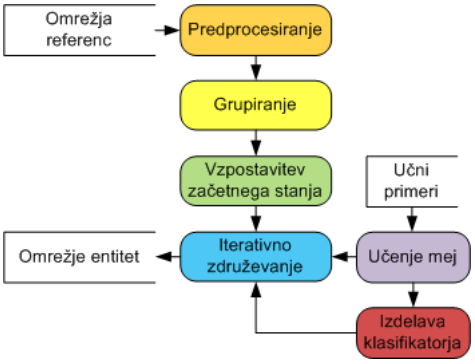
Kontekstualne značilnosti

Namesto glasovalnih pristopov v predlaganem sistemu raje uporabimo klasifikacijo, vendar pa vektorju odločitev metrik $D = [d_1, d_2, d_3 \dots]$, dodamo še vektor kontekstualnih značilnosti $F = [f_1, f_2, f_3 \dots]$. Izbrane kontekstualne značilnosti so klasifikatorju v pomoč pri napovedovanju skupne odločitve o primerjanem paru. Za naš problem združevanja omrežij izberemo naslednje značilnosti:

- dvoumnost primerjanega para,
- dolžina geodetke med grozdoma,
- razmerje vmesne centralnosti grozdov in
- razmerje med velikostjo soseščin grozdov.

Že v uvodu razdelka 3 smo opisali kako problematična je lahko dvoumnost referenc pri problemu razločevanja entitet. V ta namen kot eno izmed kontekstualnih lastnosti dodamo tudi oceno dvoumnosti primerjanega para grozdov. Le-to definiramo kot:

$$Amb(c_i, c_j) = \frac{IsAmb(c_i) + IsAmb(c_j)}{2}, \tag{3.11}$$



Slika 3.2: Celotna arhitektura sistema za združevanje omrežij.

kjer z $IsAmb(c)$ označimo dvoumnost grozda c. Na podlagi dvoumnosti referenc definiramo dvoumnost grozda kot

$$IsAmb(c) = \begin{cases} 1, & \exists r \in c.R \mid IsAmb(r) = 1 \\ 0, & \text{sicer} \end{cases}, \tag{3.12}$$

kar označuje, da je grozd dvoumen v primeru, da je vsaj ena izmed njegovih referenc označena za dvoumno. Z opisano kontekstualno značilnostjo želimo klasifikatorju omogočiti prepoznavanje takšnih primerov, v katerih je uporaba napovedi atributnih metrik nezanesljiva.

Dolžina geodetke je naslednja kontekstualna značilnost, ki jo uporabimo v sistemu. Z njo želimo klasifikatorju omogočiti prepoznavanje primerov, v katerih uporaba relacijskih metrik, ki za primerjavo izkoriščajo soseščino, ni primerna. Slednje ponazorimo na naslednjem primeru. Če je dolžina geodetke med obravnavanima grozdoma daljša od 2, to pomeni, da skupne soseščine ne bosta imela. Vse omenjene relacijske metrike v tem primeru napovejo, da primerjani par grozdov ne predstavlja iste entitete, kar pa lahko ne velja. Zato želimo, da klasifikator v teh primerih napovedi omenjenih relacijskih metrik ne upošteva.

Če primerjana grozda iz različnih omrežij dejansko predstavljata isto entiteto, velja, da imata podobne strukturne lastnosti oziroma je njun obstoj v omrežju približno enako pomemben. Zato kot dodatni kontekstualni značilnosti uporabimo tudi razmerje vmesne centralnosti in razmerje velikosti soseščin obravnavanih grozdov (glej razdelek 2.1.3), ki sta dve možni meri za opazovanje pomembnosti obstoja gradnikov v omrežjih. Slednji dve kontekstualni značilnosti, sicer klasifikatorju nista v neposredno pomoč pri izbiri oziroma

utežitvi metrik, kakor prvi dve opisani značilnosti. Predstavljata pa neko dodatno informacijo, ki lahko v posameznih primerih služi kot dodaten dokaz za ali proti združitvi nekega para grozdov.

Izgradnja in uporaba klasifikatorja

Za vsak podobnostni par grozdov iz učnega nabora podatkov zgradimo vektor $[d_1, d_2, d_3 \ldots, f_1, f_2, f_3 \ldots, L]$, kjer je d_i napoved i-te metrike o združitvi opazovanega para v enoten gradnik, f_j pa je j-ta kontekstualna lastnost za opazovani par. Z L označimo klasifikacijski razred - torej oznaka, ki pove ali opazovani par resnično opisuje isto entiteto. Nad izdelano množico primerov zgradimo poljuben klasifikator. Tega pa nato uporabimo za izdelavo odločitve o združitvi para grozdov v vsaki iteraciji iterativnega združevanja za vsak par posebej (glej razdelek 3.4.1). Za lažjo predstavo je na sliki 3.2 prikazana celotna arhitektura opisanega sistema za združevanje omrežij.

Opisan način izdelave odločitve o združitvi para grozdov torej zahteva označen začetni nabor podatkov, s pomočjo katerih zgradimo sam klasifikator. Potreba po označenem začetnem naboru podatkov je sicer slabost predlaganega pristopa. Prednost pa je v tem, da zato ne potrebujemo odločitev domenskega eksperta o utežitvi posameznih izbranih metrik in določitvi meje. S tem razbremenimo domenskega eksperta hkrati pa to pomeni tudi, da lahko pričakujemo višjo kakovost delovanja sistema v primerjavi s podobnimi sistemi.

Poglavje 4

Eksperimentalni rezultati

Za merjenje kakovosti predlaganega sistema za združevanje omrežij smo uporabili meri natančnosti (ang. *precision*) in priklica (ang. *recall*), z ozirom na vse možne pare referenc. Natančnost je definirana kot razmerje med pravilno združenimi pari in vsemi pari referenc, ki jih je sistem združil. Priklic pa je razmerje med pravilno združenimi referencami in vsemi pari referenc, ki bi jih bilo potrebno združiti. Rezultate podamo v vrednosti F_α, ki je uteženo harmonično povprečje natančnosti in priklica:

$$F_\alpha = \frac{(1+\alpha) \times precision \times recall}{\alpha \times precision + recall} \tag{4.1}$$

V praksi se pri problemu razločevanja entitet najpogosteje uporablja metrika F_1, ki natančnost in priklic ovrednoti enakovredno.

4.1 Podatki

Delovanje predlaganega sistema testiramo na dveh realnih naborih podatkov. Prav tako pa sistem preizkusimo na sintetičnih podatkih, ki jih pridobimo z uporabo naključnega generatorja omrežij. V nadaljevanju najprej podajamo opis uporabljenih realnih naborov podatkov, nato pa še opis naključnega generatorja omrežij.

4.1.1 Realni nabori podatkov

CiteSeer

CiteSeer je digitalna knjižnica znanstvenih in akademskih člankov. Uporabljen podatkovni nabor vsebuje 1504 reference, ki predstavljajo 862 resničnih znan-

stvenih člankov in 2892 referenc, ki predstavljajo 1164 resničnih avtorjev omenjenih člankov. Edina atributa, ki sta na voljo, sta 'ime' pri avtorjih in 'naslov' pri člankih. Za naš problem združevanja omrežij pa je bilo potrebno spremeniti sintakso podatkovnega nabora. Podatki so bili namreč podani v običajni tabelarični obliki, potrebno pa jih je bilo predstaviti z omrežji. Tako smo določili, da vsaka referenca članka z referencami avtorjev predstavlja posamezno omrežje. Reference člankov smo predstavili s hiper-povezavami, reference avtorjev pa z vozlišči omenjenih hiper-povezav. Naloga je zgraditi omrežje, v katerem je vsak članek predstavljen le z eno hiper-povezavo in vsak avtor le z enim vozliščem.

Facebook in Twitter

Facebook je socialni spletni portal, ki uporabnikom omogoča, da se povežejo na eno ali več omrežij (kot je na primer omrežje univerze, delovnega mesta ali nekega zemljepisnega območja) in tako preko interneta lažje komunicirajo z ostalimi uporabniki istega omrežja. Uporabnika znotraj omrežij pa sta med seboj povezana, če oba potrdita medsebojno 'prijateljstvo'.

Tudi *Twitter* je socialni spletni portal, ki uporabnikom omogoča komunikacijo preko interneta in tudi preko SMS sporočil. Tu je povezanost uporabnikov definirana nekoliko drugače kot pri Facebook-u, saj sta uporabnika lahko tudi enostransko povezana. To pomeni, da nek uporabnik sledi (ang. *follow*) drugemu. Obojestransko povezanost uporabnikov pa lahko enačimo z relacijo prijateljstva pri Facebook-u.

Za namen izvedbe poskusa smo z vsakega portala pridobili po eno omrežje uporabnikov, ki predstavljajo člane neke določene interesne skupine. Za naš primer smo uporabili uporabnike interesne skupine 'Slovenija 2020'. Na tem mestu je vredno omeniti, da se omrežje članov omenjene skupine na obeh portalih verjetno spreminja. Pridobljeno omrežje Facebook tako vsebuje 198 vozlišč in 579 povezav, omrežje Twitter pa 78 vozlišč in 656 povezav. Vozlišča v obeh omrežjih predstavljajo reference uporabnikov, povezave pa relacije prijateljstva med njimi. Naloga sistema je omrežji združiti v enotno omrežje, v katerem je vsak uporabnik predstavljen z enim vozliščem in vsaka relacija prijateljstva z eno povezavo.

Opisan podatkovni nabor smo označili in ugotovili, da omrežji dejansko predstavljata 266 resničnih uporabnikov med katerimi je 1260 relacij prijateljstva. Edini atribut, ki je pri tem podatkovnem naboru na voljo za izvedbo primerjave je 'ime' uporabnikov. Povezave pa pri tem naboru nimajo nobenih atributov.

4.1.2 Naključni generator omrežij

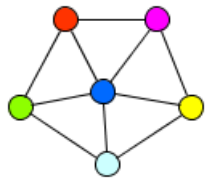
Proces naključnega generiranja omrežij referenc je sestavljen iz dveh delov. V prvem delu generiramo Poissonovo naključno omrežje entitet N_{ent}. Le-to vsebuje V vozlišč z verjetnostjo pojavitve povezave med vsakim parom vozlišč enako p. Zaradi enostavnosti se pri generiranju omrežja entitet omejimo na zgolj dva entitetna tipa, pri čemer so entitete prvega tipa predstavljene z vozlišči, entitete drugega tipa pa s povezavami. Vsako entiteto predstavljata dva naključno zgrajena tekstovna atributa s stopnjo dvoumnosti p_{amb}. To pomeni, da bosta naključno izbrani entiteti imeli z verjetnostjo p_{amb} enako vrednost nekega atributa. S tem parametrom torej nadziramo stopnjo dvoumnosti v omrežju entitet, pri čemer večja vrednost parametra pomeni tudi večjo stopnjo dvoumnosti.

V drugem delu iz zgeneriranega omrežja entitet N_{ent} kreiramo s različnih omrežij referenc N_{ref}. Vsako omrežje referenc kreiramo neodvisno od drugega in vedno izhajamo iz začetnega omrežja N_{ent}. Posamezno omrežje N_{ref_i} zgradimo tako, da vsak gradnik[1] omrežja N_{ent} z verjetnostjo p_c uporabimo pri kreiranju[2] njegove reference. Kreirano referenco nato vstavimo v omrežje N_{ref_i}. S parametrom p_c torej uravnavamo delež entitet, ki se dejansko uporabijo pri gradnji omrežja referenc. To pomeni, da s tem parametrom uravnavamo stopnjo podobnosti strukture med izdelanimi omrežji referenc. Večja kot je vrednost parametra p_c, več referenc v omrežjih N_{ref} predstavlja neko isto entiteto in posledično je podobnost strukture večja. Prikaz parametra p_c je prikazan na sliki 4.1.

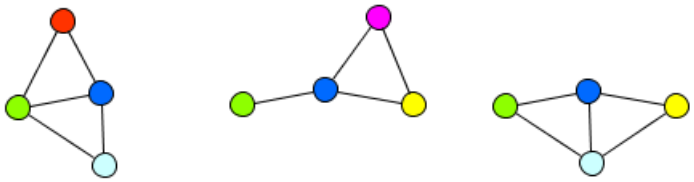
Pojasnimo še, kako sploh izdelamo referenco neke entitete. Le-to kreiramo tako, da deformiramo oziroma popačimo vrednosti njenih atributov. V ta namen smo generatorju dodali parameter p_{df}, s katerim uravnavamo stopnjo šuma pri kreiranju referenc. To pomeni, da vsak znak vrednosti atributa z verjetnostjo p_{df} nadomestimo z naključnim znakom. Večja kot je vrednost omenjenega parametra, več šuma vnašamo v vrednosti atributov referenc. Celoten postopek grajenja naključnih omrežij referenc je prikazan v algoritmu 2 v dodatku A.

[1] V primeru, da je obravnavan gradnik povezava, kreiramo tudi reference njenih vozlišč, saj sicer njen obstoj v omrežju N_{ref} ni smiseln.

[2] V primeru, da je referenca za določeno entiteto že kreirana za trenutno omrežje referenc, uporabimo kar to. V nasprotnem primeru bi vnašali redundanco tudi v posamezna omrežja referenc, kar pa ni naš namen.



(a)



(b)

Slika 4.1: Prikaz vpliva parametra p_c: (a) Omrežje entitet N_{ent}; (b) Generirana omrežja referenc N_{ref} iz omrežja N_{ent} pri vrednosti parametra $p_c = 0.5$.

4.2 Implementacija

Predlagan sistem je implementiran v programskem jeziku Java. Za potrebe obvladovanja podatkov v obliki omrežij smo uporabili knjižnico Jung[3]. V procesu iterativnega združevanja pa smo kot klasifikator uporabili odločitveno drevo, implementirano v knjižnici Rapid-i[4].

Sistem je zasnovan tako, da vse potrebne parametre modela združevanja nastavi sam. Eden od teh je tudi parameter $meja_{grup}$, ki ga potrebuje komponenta Grupiranje, opisana v razdelku 3.2. Tudi ta parameter bi lahko nastavili z učenjem kot nastavimo tudi meje posameznim metrikam (glej razdelek 3.4.1). Vendar za potrebe testiranja sistema ta parameter nastavimo ročno na razmeroma nizko vrednost 0.5 (možne vrednosti so sicer med 0 in 1). S tako nastavitvijo želimo doseči, da bo rezultat omenjene komponente relativno majhno število skupin z velikim številom referenc v vsaki skupini. Časovna zahtevnost celotnega sistema tako sicer ni najmanjša, vendar pa na ta način dosežemo, da glavno breme združevanja nosita komponenti Vzpostavitev začetnega stanja (glej razdelek 3.3) in pa predvsem Iterativno združevanje (glej razdelek 3.4). Omenjeni komponenti namreč predstavljata jedro opisanega

[3] `http://jung.sourceforge.net/`
[4] `http://rapid-i.com/`

sistema in zato želimo pri testiranju opazovati predvsem kakovost delovanja slednjih dveh.

Za namen primerjave gradnikov na osnovi atributov je v sistemu implementirana atributna metrika Soft TF-IDF s sekundarno metriko Jaro-Winkler in parametrom $\theta = 0$ (glej razdelek 2.2.1). To je tudi edina atributna metrika v sistemu. Vzrok za izbiro le-te je, da je primerna za primerjanje tako eno- kot tudi več-besednih tekstovnih atributov. Poleg tega pa tudi različni raziskovalci [10, 26] navajajo, da se dobro obnese pri problemu razločevanja entitet.

Poleg Soft TF-IDF pa so v sistemu implementirane še naslednje relacijske metrike (glej razdelek 2.2.2):

- Adamic-Adar z oceno dvoumnosti,
- Jaccardov koeficient in
- Primerjava soseščin drugega reda z uporabo metrike Jaccardov koeficient.

Avtorja v [5] navajata, da so se omenjene metrike dobro obnesle pri problemu razločevanja entitet nad bibliografskimi nabori podatkov, zato jih uporabimo tudi pri našem problemu združevanja omrežij. V sistemu pa smo, poleg omenjenih, implementirali tudi Metriko naključni sprehodi s parametroma $M = 15$ in $K = 7$. Vzrok za izbiro tudi slednje pa je, da le-ta predstavlja dobro dopolnitev metrikam, ki za primerjavo izkoriščajo sosednost. Njene značilnosti in prednosti smo sicer opisali že ob njenem opisu v razdelku 2.2.2. Za uporabo slednje pa je potrebno povezave v omrežjih tudi utežiti. V opisanih podatkovnih naborih, nad katerimi sistem testiramo, povsod obstaja le en tip povezav, zato v implementaciji omenjene metrike vse povezave utežimo enakovredno in sicer z maksimalno možno vrednostjo $\frac{1}{2}$. V primeru kompleksnejših podatkovnih naborov pa bi se verjetno morali poslužiti katerega izmed drugih pristopov. Možen pristop je denimo, da tipe povezav utežimo na podlagi izračunane povprečne vmesne centralnosti posameznega tipa povezave. Tipom povezav, ki imajo večjo (povprečno) vmesnost, torej dodelimo večjo utež, saj je njihov obstoj v omrežju bolj pomemben. Seveda pa obstajo tudi drugi modeli nastavljanja uteži.

Vsi podatkovni nabori, ki jih uporabimo pri testiranju našega sistema, so označeni. Zato je potrebno opisati še, kako iz podanega nabora podatkov izberemo podmnožico primerov, ki jih uporabimo za učenje. Popolnoma naključno vzorčenje ni priporočljivo [21], saj se tako lahko zgodi, da se strukturne lastnosti učne in testne množice medseboj močno razlikujejo. V našem sistemu zato najprej poiščemo povezane komponente vsakega obdelovanega omrežja. V kolikor so omrežja sestavljena iz velikih komponent, bi bila učna množica

velika. Zato velike komponente rekurzivno delimo na manjše tako, da odstranjujemo povezave z največjo vmesno centralnostjo, kot smo le-to definirali že v razdelku 2.1.3. Vsaka komponenta tako običajno razpade na dve manjši. Postopek prekinemo, ko so dobljene komponente dovolj majhne, da lahko z naključno izbiro celotnih komponent sestavimo učno množico željene velikosti. Učna množica v našem primeru predstavlja približno 30% celotnega podatkovnega nabora, preostanek pa predstavlja množico za izvedbo testiranja sistema.

Pri uporabi predlaganega sistema v realnosti pa bi bil izbor učne množice veliko kompleksnejši problem. Učno množico bi bilo verjetno potrebno izbrati popolnoma ročno in jo nato seveda tudi označiti, kar pa je vse prej kot trivialna naloga.

4.3 Rezultati

V nadaljevanju predstavimo rezultate delovanja sistema nad opisanimi realnimi in sintetičnimi nabori podatkov.

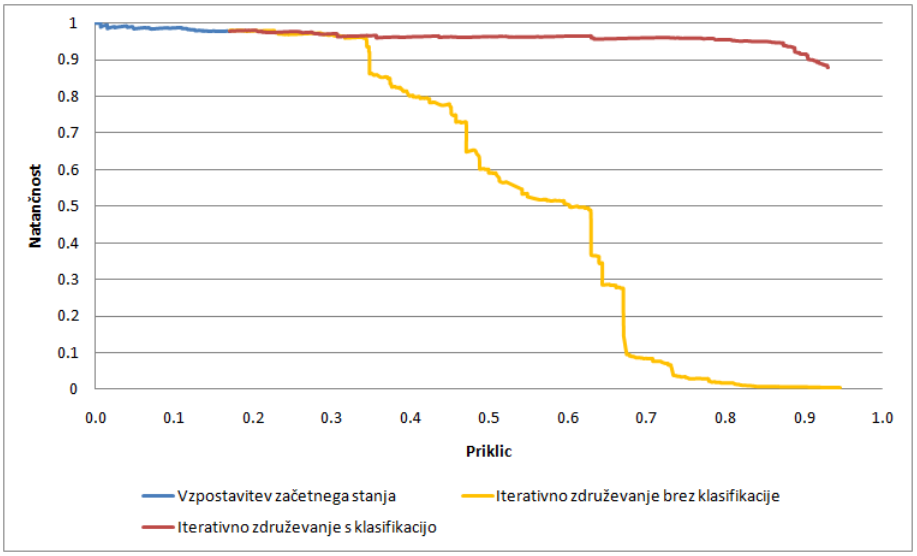
4.3.1 Združevanje realnih omrežij

CiteSeer

Nad podatkovno zbirko CiteSeer smo izvedli deset ponovitev združevanja omrežij. Povprečen rezultat je znašal $F_1 = 0.896$ s standardnim odklonom $s = 0.011$. Kot smo omenili že v razdelku 4.2 podatkovni nabor razdelimo na učno in testno množico. Pri tem podatkovnem naboru naključno izbiramo kar celotna omrežja, saj vsaka referenca članka s pripadajočimi referencami avtorjev predstavlja neko ločeno omrežje.

Na sliki 4.2 je prikazano razmerje med natančnostjo in priklicom med samim izvajanjem[5] sistema za omenjen nabor podatkov. Modri del krivulje prikazuje razmerje med natančnostjo in priklicom za komponento Vzpostavitev začetnega stanja, rdeči del pa za komponento Iterativno združevanje. Iz grafa razberemo, da se določen upad natančnosti pojavi že pri nizkem priklicu (med 0 in 0.15); torej med samim izvajanjem komponente Vzpostavitev začetnega stanja. Naj spomnimo, da je naloga omenjene komponente, z združitvijo določenih referenc v skupne grozde, povezati podana omrežja v enotno omrežje. Omenjen upad natančnosti pokaže, da ta komponenta sistema v skupne grozde

[5]Priklic med samim izvajanjem sistema vedno raste, saj gradnike v našem algoritmu vedno le združujemo.



Slika 4.2: Razmerje med natančnostjo in priklicom za primer združevanja podatkovnega nabora *CiteSeer*.

združi tudi reference, ki ne predstavljajo istih entitet. To so najverjetne dvoumne reference, saj za združitev dveh referenc v skupen grozd v tej komponenti zahtevamo, da se natančno ujemajo v vseh istopomenskih atributih. Možno izboljšavo predstavlja že drugačna nastavitev parametra k, s katerim določimo potrebno število enakih sosedov za združitev dvoumnih parov referenc. Strožji kriterij bi denimo dosegli s pravilom, da dvoumne pare referenc združimo takrat, kadar imajo k enakih sosedov, pri čemer pa bi kot enake sosede obravnavali le tiste, ki se ujemajo v vseh atributih in imajo tudi ti določeno število enakih sosedov. Možna slabost tega pristopa pa je lahko ta, da je opisan pristop prestrog in tako ne bi zadostili nalogi komponente, ki je, povezati omrežja v enotno omrežje.

Konstantno visoka natančnost (nad 0.95) tudi pri visokem priklicu pokaže relativno dobro delovanje komponente Iterativno združevanje. Neznaten upad natančnosti se pojavi le pri priklicu nad 0.88. Upad natančnosti na tem delu pokaže, da sistem proti koncu Iterativnega združevanja združuje tudi nekatere gradnike, ki ne predstavljajo istih entitet. V tem delu namreč na vrsto prihajajo pari z najnižjimi vrednostmi podobnosti, zato je določen upad natančnosti razumljiv.

Oranžna krivulja prikazuje razmerje med natančnostjo in priklicom za komponento Iterativno združevanje, vendar brez uporabe klasifikacije (opisano v

razdelku 3.4.1). Tak model združevanja je tako podoben modelom, ki smo jih opisali v uvodnem delu razdelka 3 in tudi v razdelku 3.4.1. Opazimo, da natančnost prvič znatno upade že pri razmeroma nizkem priklicu 0.35, drugič pa pri priklicu 0.65. Zato je potrebno pri uporabi teh modelov iskati kompromis med željeno natančnostjo in priklicom. To dosežemo z ustrezno nastavitvijo meje podobnosti, do katere pare še združujemo v skupne grozde. Z uporabo klasifikacije pa to pomanjkljivost odpravimo, saj sistem obravnava vse pare in za vsakega posebej izdela odločitev o združitvi. Razlika v natančnosti pri doseženem priklicu potrjuje, da uporaba predlaganega modela (torej s klasifikacijo) daje znatno boljše rezultate kot uporaba obstoječih modelov.

Facebook in Twitter

Tudi nad podatki pridobljenimi s portalov Facebook in Twitter smo izvedli deset ponovitev združevanja. Povprečen rezultat je znašal $F_1 = 0.835$ s standardnim odklonom $s = 0.139$. Natančni rezultati so podani v tabeli 4.1.

	F_1
avg	0.835
s	0.139
max	0.957
min	0.629

Tabela 4.1: Rezultati združevanja omrežij Facebook in Twitter.

Tudi rezultati združevanja tega nabora podatkov so torej dobri, vendar pa o rezultatih ne moremo biti popolnoma prepričani, da so pravilni. Podatke smo namreč označili ročno in ni garancije, da so pravilno označeni. Prav tako pa iz dobljenih rezultatov nad omenjenim podatkovnim naborom ne moremo sklepati o uspešnosti delovanja sistema v splošnem. Izbran podatkovni nabor je namreč premajhen in nereprezentativen, saj je v obeh obdelovanih omrežjih le malo takih referenc, ki dejansko predstavljajo isto entiteto. Za omrežja podatkov, ki predstavljajo isto domeno, je presek običajno veliko večji.

Opozorimo še, da smo pri izdelavi rezultatov za omenjen podatkovni nabor upoštevali le rezultate tistih zagonov, pri katerih je bil, v naključno izbrano učno množico dejansko zajet željen odstotek parov referenc, ki predstavljajo isto entiteto. V nasprotnem primeru rezultati ne bi bili reprezentativni, saj bi se lahko zgodilo, da v učni množici ne bi bil niti en primer para referenc iste entitete. Na splošno je kakovost učne množice pri tem podatkovnem naboru

zelo nihala, kar potrjuje tudi relativno velik standardni odklonu v primerjavi z rezultatom podatkovnega nabora CiteSeer. Vzrok za težavno izbiro učne množice pa je ravno majhen presek med obema obdelovanima omrežjema. Pri drugih podatkovnih naborih takšnih težav ni bilo, saj je bil presek obdelovanih omrežij znatno večji.

4.3.2 Združevanje sintetičnih omrežij

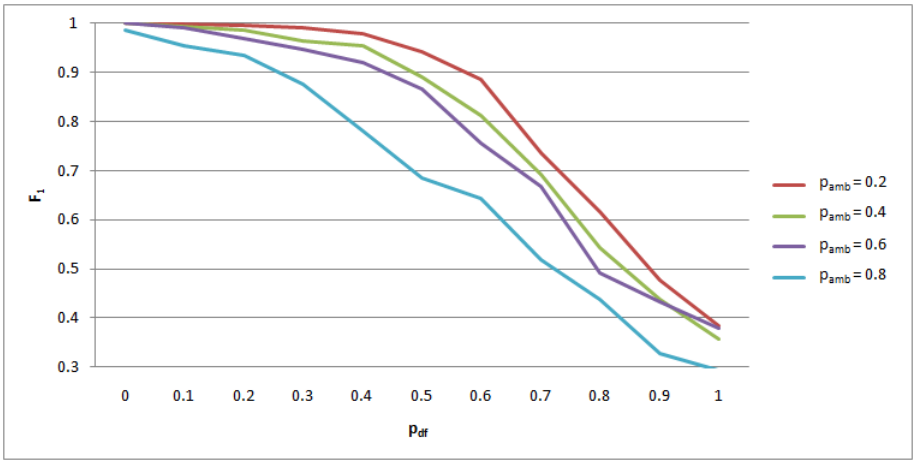
V nadaljevanju prikažemo rezultate dveh eksperimentov, ki smo jih izvedli nad sintetičnimi podatki. Namen izvedbe teh je ugotoviti, kako različne lastnosti podatkovnih naborov vplivajo na samo kakovost združevanja. Rezultati združevanja so predstavljeni kot povprečna vrednost dosežene mere F_1 glede na deset izvedenih ponovitev vsakega poskusa.

Eksperiment 1: Uspešnost združevanja omrežij v odvisnosti od stopnje šuma

Za vsako ponovitev tega eksperimenta smo generirali naključno omrežje entitet velikosti 500 vozlišč in z verjetnostjo $p = \frac{1}{500}$ pojavitve povezave med vsakim parom vozlišč. Iz dobljenega omrežja entitet smo nato generirali po tri omrežja referenc z enako strukturo ($p_c = 1$). Namen eksperimenta je opazovati kakovost združevanja omrežij v odvisnosti od stopnje deformiranja referenc pri različnih stopnjah dvoumnosti. Stopnja deformiranja (p_{df}) pomeni šum, s katerim generiramo referenco neke entitete, dvoumnost (p_{amb}) pa pomeni verjetnost, da naključno izbrani referenci z enakimi atributi dejansko predstavljata različni entiteti. Podroben opis parametrov je podan pri samem opisu uporabljenega generatorja v razdelku 4.1.2.

Rezultati eksperimenta so prikazani na sliki 4.3, od koder lahko razberemo, da pojav šuma močno vpliva na kakovost združevanja omrežij. Kakovost združevanja omrežij do stopnje deformiranja $p_{df} = 0.5$ je relativno visoka, saj je mera F_1 v večini primerov nad 0.85, nato pa začne zelo strmo padati. Iz tega lahko sklepamo, da je sistem na pojav šuma v podatkih do neke mere relativno obstojen, nad njo pa kakovost strmo upade. Vrednost parametra $p_{df} = 0.5$ sicer pomeni, da z verjetnostjo 0.5 vsak znak atributa nadomestimo z naključno izbiro novega znaka.

Rezultati tega ekperimenta pokažejo tudi, da na kakovost združevanja omrežij vpliva tudi stopnja dvoumnosti. Večja vrednost p_{amb} (s katero uravnavamo dvoumnost) v splošnem pomeni slabši rezultat. Najbolj izmed vseh izstopa rezultat združevanja visoko ($p_{amb} = 0.8$) dvoumnih omrežij, saj je



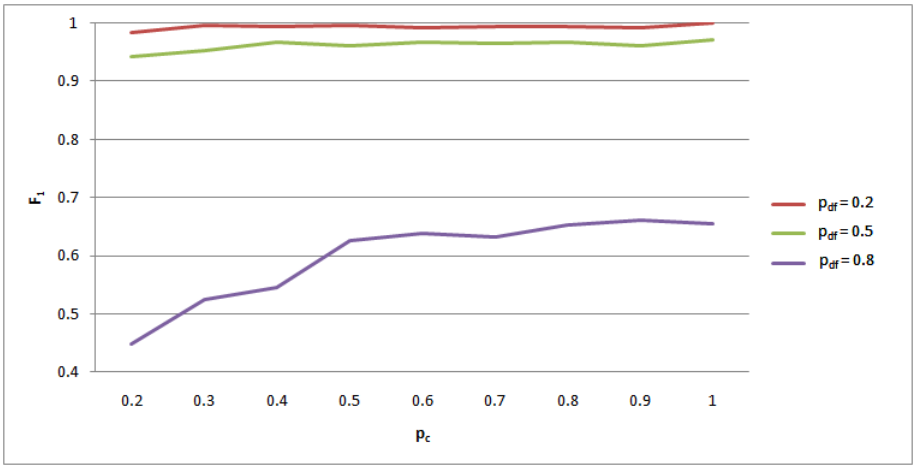
Slika 4.3: Kakovost združevanja omrežij v odvisnosti od stopnje deformiranja.

kakovost delovanja sistema nad temi najnižja. Pri teh namreč kakovost združevanja strmo upade že pri majhni pojavitvi šuma ($p_{df} > 0.2$), medtem ko je sistem pri nižjih stopnjah dvoumnosti na pojav šuma bolj obstojen.

Eksperiment 2: Uspešnost združevanja omrežij v odvisnosti od stopnje podobnosti strukture omrežij

Namen naslednjega eksperimenta je ugotoviti, ali različnost strukture omrežij vpliva na samo kakovost združevanja le-teh. Za namen tega eksperimenta smo v vsaki ponovitvi generirali omrežje entitet velikosti 500 vozlišč in z verjetnostjo $p = \frac{1}{500}$ pojavitve povezave med vsakim parom vozlišč. Stopnja dvoumnosti je bila pri generiranju omrežij za omenjen eksperiment minimalna ($p_{amb} = 0$). Iz dobljenega omrežja smo nato generirali po tri omrežja referenc. Podobnost strukture pri generiranju omrežij referenc uravnavamo s parametrom p_c, ki dejansko pomeni verjetnost, da naključno izbrano entiteto uporabimo pri samem kreiranju reference za omrežje referenc. Podrobnosti tega parametra so predstavljene pri samem opisu generatorja v razdelku 4.1.2.

Rezultati opisanega eksperimenta so prikazani na sliki 4.4. Iz grafa lahko razberemo, da stopnja podobnosti strukture omrežij (p_c) v določenih primerih lahko vpliva na samo kakovost delovanja sistema. Predvsem ima ta velik vpliv pri podatkovnih naborih, kjer je stopnja šuma oziroma deformiranja visoka ($p_{df} = 0.8$). Pri podatkovnih naborih, v katerih šum ni tako znatno prisoten ($p_{df} = 0.2$ in $p_{df} = 0.5$), pa je kakovost združevanja omrežij konstantno



Slika 4.4: Kakovost združevanja omrežij v odvisnosti od stopnje podobnosti strukture omrežij.

visoka, ne glede na stopnjo podobnosti (p_c) med njimi. To lahko povežemo z dognanjem iz prejšnega eksperimenta, kjer smo ugotovili, da je sistem do neke mere relativno obstojen na pojav šuma. Pri podatkovnih naborih, kjer je stopnja šuma večja, pa domnevamo, da pridejo v poštev predvsem odločitve relacijskih metrik, saj so rezultati atributnih metrik nad omenjenimi nabori verjetno neuporabni. V primeru raznolikih struktur obdelovanih omrežij pa so tudi odločitve relacijskih metrik lahko napačne, kar je razvidno tudi iz grafa. Kakovost združevanja omrežij z visoko stopnjo šuma namreč narašča z naraščanjem stopnje podobnosti med obdelovanimi omrežji.

4.4 Diskusija

Rezultati obeh opisanih eksperimentov so bili sicer pričakovani. Vendar pa smo z opravljenimi eksperimenti prišli tudi do nekaterih pomembnih zaključkov. S prvim eksperimentom potrdimo domnevo, da pojav šuma in dvoumnosti vpliva na samo kakovost delovanja sistema. Hkrati ugotovimo tudi, da je sistem na pojav šuma do neke mere relativno obstojen. Na primeru našega eksperimenta to kritično točko predstavlja stopnja deformiranja $p_{df} = 0.5$, saj je kakovost združevanja pod omenjeno stopnjo v večini primerov nad 0.8, nato pa strmo upade. Rezultati tega eksperimenta razkrivajo tudi, da je sistem do določene mere odporen tudi na pojav dvoumnosti, saj so si rezultati pri nizkih stopnjah

dvoumnosti ($p_{amb} \leq 0.6$) relativno podobni.

Z drugim eksperimentom potrdimo domnevo, da stopnja podobnosti med obravnavanimi omrežji vpliva na samo kakovost združevanja le-teh. To se izrazito pokaže predvsem pri naboru podatkov, kjer je stopnja šuma visoka.

Doseženi rezultati delovanja sistema nad obema realnima naboroma so dobri in smo z delovanjem sistema nad njimi zadovoljni. Vendar pa lastnosti izbranih podatkovnih naborov niso zadostne, da bi na podlagi dobljenih rezultatov lahko sklepali o uspešnosti sistema v splošnem. Predvsem bi bilo potrebno izvesti testiranje sistema tudi nad drugimi domenami podatkov z različnimi lastnostmi. Uporabljeni podatkovni nabori v našem primeru so predstavljeni z relativno enostavnimi omrežji, saj ta vsebujejo le en tip vozlišč in en tip povezav. Zato bi bilo potrebno izvesti testiranje nad kompleksnejšimi omrežji, kjer obstaja več različnih tipov vozlišč in povezav. Prav tako pa bi bilo potrebno sistem preizkusiti tudi nad usmerjenimi omrežji, saj so bili vsi uporabljeni podatkovni nabori predstavljeni le z neusmerjenimi omrežji.

Poglavje 5

Zaključek

V diplomski nalogi predstavimo sistem za združevanje poljubnega števila omrežij. Sistem je moč uporabiti pri vseh problemih integracije podatkov, ki so predstavljeni z omrežji. Pri tem bi posebej izpostavili problem predprocesiranja podatkov pri izvajanju različnih analiz, saj analize nad združenimi omrežji dajejo boljše rezultate kot pa analiza vsakega omrežja posebej. Prav tako pa je tudi izdelava analize nad enim omrežjem časovno in prostorsko cenejša kot nad več ločenimi omrežji.

V sistemu uporabimo obstoječ pristop skupinskega razločevanja, pri katerem odločitve o združitvi posameznih gradnikov omrežij niso neodvisne. To pomeni, da so odločitve o združitvah odvisne od prejšnjih odločitev v sistemu. Za samo izvedbo primerjave gradnikov uporabimo tako navadne atributne kot tudi relacijske metrike. Potrebno pa je tudi poudariti, da se je predlagan sistem sposoben samodejno prilagajati podanemu naboru podatkov in nastavitev parametrov tako ni potrebna. Sistem pa zato za delovanje potrebuje označen začetni nabor podatkov.

Sistem bi bilo moč izboljšati na številnih področjih. Eno izmed možnih izboljšav predstavlja področje izbire kontekstualnih lastnosti, s katerimi dopolnimo vektor odločitev primerjalnih metrik. Pri določitvi kontekstualnih lastnosti se je izkazalo, da je izbira le-teh problematična za vozlišča in povezave skupaj. Težavno je bilo namreč izbrati lastnosti, ki enakovredno opišejo kontekst, v katerem se nahaja bodisi povezava, bodisi vozlišče. Tako bi lahko kontekstualne lastnosti določili za vsak tip gradnikov omrežij posebej. Na podlagi tega bi izgradili tudi več različnih klasifikatorjev, s katerimi bi verjetno dobili še boljše napovedi o združitvi posameznih gradnikov. Možno izboljšavo predstavlja tudi drugačen način nastavljanja mej posameznim metrikam. Kakovost združevanja bi se verjetno izboljšala v primeru, ko bi tudi meje metrikam

določali za vsak entitetni tip posebej.

Naslednjo stopnjo razvoja predlaganega sistema pa bi predstavljala dodatna komponenta, s katero bi vsa obdelovana omrežja prevedli na skupno sintakso. V praksi namreč lahko naletimo na primere podatkovnih naborov, kjer različna omrežja na različne načine predstavljajo isto podatkovno domeno. Tega postopka seveda ni mogoče v celoti avtomatizirati, saj je način prevedbe omrežij odvisen od primera do primera. Preslikavo bi tako določil domenski ekspert, naloga omenjene komponente pa bi bila prevesti omrežja na podlagi definirane preslikave. Na ta način bi lahko predlagan sistem uporabljali tudi za združevanje sintaktično različnih omrežij. Dodatno razširitev sistema pa bi predstavljala tudi komponenta, v kateri bi iz množice referenc, za katere ugotovimo, da predstavljajo neko entiteto, izbrali tisto, ki najbolje opisuje entiteto. S tem bi bile entitete v končnem omrežju predstavljene na najprimernejši način in tako bi sistem dejansko predstavljal celovito orodje za združevanje podatkovnih omrežij za namen nadaljnjih analiz.

Sistem je bil preizkušen nad dvema realnima naboroma podatkov. Rezultati združevanja so nad obema naboroma dobri, vendar pa lastnosti izbranih naborov niso zadostne, da bi lahko sklepali o uspešnosti delovanja sistema v splošnem. Prav tako je bil sistem preizkušen nad sintetičnimi podatki, pridobljenimi z uporabo naključnega generatorja omrežij, kjer smo opazovali delovanje sistema v odvisnosti od različnih lastnosti podatkovnih naborov. Na podlagi dobljenih rezultatov testiranja sistema lahko zaključimo, da smo z delovanjem sistema zadovoljni, saj so rezultati nad različnimi podatkovnimi nabori dobri. Prav tako je sistem sposoben tudi samodejnega prilagajanja glede na podan nabor podatkov in tako lahko zaključimo, da so cilji naloge v veliki meri doseženi.

Dodatek A

Algoritmi

Algoritem 2 Naključni generator omrežij

```
1: init N_ent
2: for i < V do
3:    N_ent.vertices ← create vertex(p_amb)
4: end for
5: for all v_1 in N_ent.vertices do
6:    for all v_2 in N_ent.vertices do
7:       if random() < p then
8:          e ← create edge(p_amb)
9:          N_ent.add edge (v_1, v_2, e)
10:      end if
11:   end for
12: end for

13: for i < s do
14:   init N_ref
15:   for all t in N_ent.elements do
16:      if random() < p_c then
17:         N_ref ← create/get reference(t, p_df)
18:      end if
19:   end for
20: end for
```

Slike

2.1 Prikaz problema *Seven Bridges of Königsberg* 8
2.2 Prikaz različnih tipov omrežij 9
2.3 Primer socialnega omrežja spolnih odnosov med ljudmi z virusom HIV 10
2.4 Primer slabega delovanja metrike *Skupni sosedi* 19

3.1 Komponente sistema za združevanje omrežij 25
3.2 Celotna arhitektura sistema za združevanje omrežij 36

4.1 Prikaz vpliva parametra p_c 41
4.2 Razmerje med natančnostjo in priklicom za primer združevanja podatkovnega nabora *CiteSeer* 44
4.3 Kakovost združevanja omrežij v odvisnosti od stopnje deformiranja 47
4.4 Kakovost združevanja omrežij v odvisnosti od stopnje podobnosti strukture omrežij 48

Tabele

4.1 Rezultati združevanja omrežij Facebook in Twitter 45

Seznam algoritmov

1 Algoritem iterativnega združevanja 33
2 Naključni generator omrežij 52

Literatura

[1] Adamic A., Adar E., "Friends and neighbours on the Web", v *Social Network vol. 25*, str. 211-230, 2001.

[2] Ananthakrishna R., Chaudhuri S. in Ganti V., "Eliminating fuzzy duplicates in data warehouses", v *The International Conference on Very Large Databases (VLDB)*, Hong Kong, China, 2002.

[3] Bilenko M. in Mooney R., "Adaptive duplicate detection using learnable string similarity measure", v *KDD*, 2003.

[4] Erdos P. in Rényi A., "On random graphs", v *Publicationes Mathematicae Debrecen 6*, str. 290-297, 1959.

[5] Bhattacharya I. in Getoor L., "Collective Entity resolution in relational data", 2007.

[6] Bhattacharya I. in Getoor L., "A Leatent Dirichlet Model for Unsupervised Entity Resolution", v *SDM* 2006.

[7] Bhattacharya I. in Getoor L., "Generator for Noisy Reference Data with Co-occurrence Relationships", 2007.

[8] Bhattacharya I. in Getoor L., "Entity resolution in graphs", *Mining Graph Data*, str. 311 2006.

[9] Bhattacharya I. in Getoor L., "Iterative Record Linkage for Cleaning and Integration", v *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, str. 11-18, 2004.

[10] Cohen W., Ravikumar P. in Fienberg S., "A comparison of string distance metrics for name-matching tasks", v *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (HWeb-03)*, 2003.

[11] Chen Z., Kalashnikov D. in Mehrotra S., "Exploiting Context Analysis for Combining Multiple Entity Resolution Systems", v *Proceedings of the 35th SIGMOD international conference on Management of data*, Providence, RI, USA, str. 207-218, 2009.

[12] Chen Z., Kalashnikov D. in Mehrotra S., "Adaptive Graphical Approach to Entity Resolution", v *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, str. 204-213, ACM New York, NY, USA, 2007.

[13] Eppstein D. in Wang J., "A steady state model for graph power laws", v *2nd Int. Worksh. Web Dynamics*, Honolulu, 2002.

[14] Fellegi I. in Sunter A., "A theory for record linkage", v *Journal of the American Statistical Association*, str. 1183-1210, 1969.

[15] Freeman L. C., "A set of measures of centrality based on betweenness", v *Sociometry*, str. 35-41, 1977.

[16] Hernandez M. in Stolfo S., "The merge/purge problem for large databases", v *SIGMOND*, 1995.

[17] Hölzer R., Malin B. in Sweeney L., "Email Alias Detection Using Social Network Analysis", v *Proceedings of the 3rd international workshop on Link discovery*, Chicago, IL, USA, str. 52-57, 2005.

[18] Herzog T, Scheure F. in Winkler W., "Data quality and record linkage techniques", 2007.

[19] Iria J., Xia L. in Zhang Z., "Web People Search Disambiguation using Random Walks", *Proceedings of the 4th International Workshop on Semantic Evaluations (Semeval 2007)*, Praga, Češka, 2007.

[20] Jaro M. A., "Advances in record linking methodolg as applied to the 1985 census of Tampa Florida", v *Journal of the American Statistical Society 84 (406)*, str. 414-420, 1989.

[21] Jensen D., "Statistical Challenges to Inductive Inference in Linked Data", v *Seventh International Workshop on Artificial Intelligence and Statistics*, 1999.

[22] Kalashnikov D. in Mehrotra S., "A probabilistic model for entity disambiguation using relationships", v *SIAM Internation Conference on Data Mining (SDM)*, Newport Beach, CA, USA, str. 21-23, 2005.

[23] Kalashnikov D. in Mehrotra S., "Domain-independent data cleaning via anlysis of entity-relationship graph", v *ACM Trans. Database Syst.*, vol. 31, no 2, str. 716-767, 2006.

[24] Levenshtein V. I., "Binary codes capable of correcting deletions, insertions, and reversals", v *Soviet Physics Doklady 10*, str. 707–710, Rusia, 1966.

[25] Malin B., "Unsupervised Name Disambiguation via Social Network Similarity", *Proceedings of the Workshop on Link Analysis, Counterterrorism and Security* v sodelovanju z *SIAM International Conference on Data Mining*, Newport Beach, CA, USA, str. 93-102, 2005.

[26] Moreau E., Yvon F. in Cappé O., "Robust Similarity Measures for Named Entites Matching", v *Proceedings of the 22nd International Conference on Computational Linguistics*, str. 593-600, 2008.

[27] McCallum A., Nigam K. in Ungar L., "Efficient clustering of high-dimensional data sets with application to reference matching", v *SIGKDD*, 2000.

[28] Milgram S., "The small world problem", v *Today*, str. 60-67, 1967.

[29] Newcombe H., Kennedy J., Axford S. in James A., "Automatic linkage of vital records", v *Science*, str. 954-959, 1959.

[30] Newman M. E. J., "Mathematics of networks", v *The New Palgrave Encyclopedia of Economics* 2nd edition, 2008.

[31] Newman M. E. J., "The structure and function of complex networks", v *SIAM Review 45*, str. 167-256, 2003.

[32] Nuray-Turan R., Kalashnikov D. in Mehrotra S., "Self-tuning in Graph-based Reference Disambiguation", v *Advances in Databases: Concepts, Systems and Applications*, str. 325-336, 2008.

[33] Potterat J. J., Phillips-Plummer L., Muth S. Q., Rothenberg R. B., Woodhouse T. S., Maldonado-Long T. S., Zimmerman H. P. in Muth J. B. "'Risk network structure in early epidemic phase of HIV transmission in Colorado Springs', v *Sexually Transmitted Infections*, str. i159-i163, 2002.

[34] Shu L., Long B. in Meng W., "A Latent Topic Model for Complete Entity Resolution", v *Proceedings of the 2009 IEEE International Conference on Data Engineering*, str. 880-891, 2009.

[35] Singla P. in Domingos P., "Entity resolution with markov logic", v *Proceedings of the Sixth IEEE International Conference on Data Mining*, str. 572-582, 2006.

[36] Sabidussi G., "The centrality index of a graph", v *Psychometrika 31 (4)* str. 1966.

[37] Salton G., Wong A. in Yang C., "A vector space model for automatic indexing", v *Communications of the ACM*, volume 18, no. 11, str. 613-620, 1975.

[38] Solomonoff R. in Rapoport A., "Connectivity of random nets", v *The Bulletin of mathematical biophysics 13*, str. 107-117, 1951.

[39] Watts D. J. in Strogatz S. H., "Collective dynamics of small-world networks", v *Nature*, str. 440-442, 1998.

[40] Winkler W. E., "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage", v *Proceedings of the Section on Survey Research Methods, American Statistical Assn.*, str. 354-359, 1990.

[41] Winkler W. E., "The state of record linkage and current research problems", *Statistical Research Division, US Bureau of the Census, Washington, DC*, 1999.

[42] Whang S., Menestrina D., Koutrika G. in Theobald M, "Entity Resolution with Iterative Blocking", *Proceedings of the 35th SIGMOD international conference on Management of data*, str. 212-232, 2009.

[43] "Seven Bridges of Königsberg", dostopno na `http://en.wikipedia.org/wiki/Graph_theory`

[44] "Citeseer dataset", dostopno na `http://www.cs.umd.edu/projects/linqs/projects/er/DATA/citeseer.dat`

[45] "Facebook", dostopno na `http://www.facebook.com/`

[46] "Twitter", dostopno na `http://www.twitter.com/`