

*Iterativno pridobivanje semantičnih podatkov iz
nestrukturiranih besedilnih virov*

Slavko Žitnik

DOKTORSKA DISERTACIJA

PREDANA

FAKULTETI ZA RAČUNALNIŠTVO IN INFORMATIKO

KOT DEL IZPOLNJEVANJA POGOJEV ZA PRIDOBITEV NAZIVA

DOKTOR ZNANOSTI

S PODROČJA

RAČUNALNIŠTVA IN INFORMATIKE



Ljubljana, 2014

*Iterativno pridobivanje semantičnih podatkov iz
nestrukturiranih besedilnih virov*

Slavko Žitnik

DOKTORSKA DISERTACIJA

PREDANA

FAKULTETI ZA RAČUNALNIŠTVO IN INFORMATIKO

KOT DEL IZPOLNJEVANJA POGOJEV ZA PRIDOBITEV NAZIVA

DOKTOR ZNANOSTI

S PODROČJA

RAČUNALNIŠTVA IN INFORMATIKE



Ljubljana, 2014

IZJAVA

Izjavljam, da sem avtor dela in da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali na drugem visokošolskem zavodu, razen v primerih, kjer so navedeni viri.

— Slavko Žitnik —

december 2014

ODDAJO SO ODOBRILI

dr. Marko Bajec

redni profesor za računalništvo in informatiko

MENTOR IN ČLAN OCENJEVALNE KOMISIJE

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

dr. Zoran Bosnić

izredni profesor za računalništvo in informatiko

PREDSEDNIK OCENJEVALNE KOMISIJE

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

dr. Dunja Mladenić

izredna profesorica za računalništvo in informatiko

ZUNANJA ČLANICA OCENJEVALNE KOMISIJE

Institut Jožef Stefan, Mednarodna podiplomska šola Jožefa Štefana in Univerza na Primorskem

PREDHODNA OBJAVA

Izjavljam, da so bili rezultati obravnavane raziskave predhodno objavljeni/sprejeti za objavo v recenzirani reviji ali javno predstavljeni v naslednjih primerih:

- [1] S. Žitnik, L. Šubelj, in M. Bajec. SkipCor: Skip-Mention Coreference Resolution Using Linear-Chain Conditional Random Fields. *PLoS ONE*, 9(6):e100101, 2014.
doi: [10.1371/journal.pone.0100101](https://doi.org/10.1371/journal.pone.0100101).
- [2] S. Žitnik, L. Šubelj, D. Lavbič, O. Vasilecas, in M. Bajec. General context-aware data matching and merging framework. *Informatica*, 9(1):119–152, 2013.
- [3] S. Žitnik, M. Žitnik, B. Zupan, in M. Bajec. Sieve-based relation extraction of gene regulatory networks from biological literature. *BMC Bioinformatics*, str. 1–29, 2015.
- [4] S. Žitnik, M. Žitnik, B. Zupan, in M. Bajec. Extracting Gene Regulation Networks Using Linear-Chain Conditional Random Fields and Rules. *Proceedings of BioNLP Shared Task 2013 Workshop*, str. 178–187, Sofija, Bolgarija, 2013.
- [5] S. Žitnik, in M. Bajec. Token- and Constituent-based Linear-chain CRF with SVM for Named Entity Recognition. *BioCreative Challenge Evaluation Workshop*, str. 144–151, Washington, ZDA, 2013.

Potrjujem, da sem pridobil pisna dovoljenja vseh lastnikov avtorskih pravic, ki mi dovoljujejo vključitev zgoraj navedenega materiala v pričujočo disertacijo. Potrjujem, da zgoraj navedeni material opisuje rezultate raziskav, izvedenih v času mojega podiplomskega študija na Univerzi v Ljubljani.

IN MEMORIAM

Marinka Brezovar

1957–1960

"Data is the new oil."

— Clive Humby, *Kellogg School*, 2006.



— qingqing, *Heavy steel machines pumping oil.*

POVZETEK

Živimo v času, ko ustvarjamo ogromne količine podatkov, od katerih je večina nestrukturiranih. Na internetu uporabniki vsako minuto objavijo več kot 200.000 besedilnih dokumentov in skupaj napišejo več kot 200 milijonov e-poštnih sporočil. Do teh podatkov bi želeli dostopati v strukturirani obliki, zato se v okviru te disertacije ukvarjamo z ekstrakcijo informacij iz besedil. Ekstrakcija informacij je tip informacijskega poizvedovanja, pri čemer so glavne naloge prepoznavanje imenskih entitet, ekstrakcija povezav in odkrivanje koreferenčnosti. Disertacija sestoji iz štirih jedrnih poglavij, kjer v vsakem predstavimo svojo nalogo ekstrakcije in jih na koncu združimo z iterativno metodo v sistem za celostno ekstrakcijo informacij. Najprej predstavimo nalogo odkrivanja koreferenčnosti, katere cilj je poiskati vse omenitve za določeno entiteto in jih združiti. Kot lastno rešitev predlagamo sistem SkipCor, ki problem pretvori v označevanje zaporedij, nad katerimi uporabimo verjetnostne modele prvega reda. Za uspešno odkrivanje koreferenčnih omenitev na daljših razdaljah predlagamo inovativno transformacijo v zaporedja z izpuščenimi omenitvami in dosežemo primerljive ali boljše rezultate kot ostali znani pristopi. Podoben način uporabimo tudi pri odkrivanju povezav. Tu z različnimi tipi oznak in s pomočjo pravil omogočimo razpoznavanje hierarhičnih povezav. S predlaganim pristopom dosežemo najboljši rezultat na tekmovanju za ekstrakcijo povezav za namene odkrivanja genskega regulatornega omrežja. Nazadnje predstavimo še najstarejšo in najbolj raziskano nalogo za prepoznavanje imenskih entitet. Naloga se ukvarja z označevanjem ene ali več besed, ki predstavljajo določen tip entiete – na primer osebe. V disertaciji prilagodimo uporabo standardnih postopkov za označevanje zaporedij ter na tekmovanju v odkrivanju kemijskih spojin in zdravil dosežemo sedmo mesto. Vse naloge uspemo rešiti z linearno-verižnimi modeli pogojnih naključnih polj in jih združiti v iterativni metodi, ki kot vhod sprejme nestrukturirano besedilo ter vrne ekstrahirane entitete s povezavami med njimi. Le-te so

označene na podlagi sistemske ontologije, kar v prihodnje zagotavlja boljšo interoperabilnost. Področje ekstrakcije informacij je v slovenskem jeziku še zelo neraziskano, zato v dodatku vključimo še seznam prevodov izbranih terminov iz angleškega v slovenski jezik.

Ključne besede: ekstrakcija informacij, odkrivanje koreferenčnosti, ekstrakcija povezav, prepoznavanje imenskih entitet

ABSTRACT

Nowadays we generate an enormous amount of data and most of it is unstructured. The users of Internet post more than 200,000 text documents and together write more than 200 million e-mails online every single minute. We would like to access this data in a structured form and that is why we throughout this dissertation deal with information extraction from text sources. Information extraction is a type of information retrieval, where the main tasks are named entity recognition, relationship extraction, and coreference resolution. The dissertation consists of the four main chapters, where each of them represents a separate information extraction task and the last chapter which introduces a combination all of the three tasks into an iterative method within an end-to-end information extraction system. First we introduce the task of coreference resolution with its goal of merging all of the mentions that refer to a specific entity. We propose SkipCor system that casts the task into a sequence tagging problem for which first order probabilistic models can be used. To enable the detection of distant coreferent mentions we propose an innovative transformation into skip-mention sequences and achieve comparable or better results than other known approaches. We also use a similar transformation for relationship extraction. There we use different tags and rules that enable the extraction of hierarchical relationships. The proposed solution achieves the best result at the relationship extraction challenge between genes that form a gene regulations network. Lastly we present the oldest and most thoroughly researched task of named entity recognition. The task deals with a tagging of one or more words that represent a specific entity type - for example, persons. In the dissertation we adapt the use of standard procedures for the sequence tagging tasks and achieve the seventh rank at the chemical compound and drug name recognition challenge. We successfully manage to solve all of the three problems using linear-chain conditional random fields models. We combine the tasks in an iterative method that

accepts an unstructured text as input and returns extracted entities along with relationships between them. The output is represented according to a system ontology which provides better data interoperability. The information extraction field for the Slovene language is not yet well researched which is why we also include a list of translations of the selected terms from English to Slovene.

Key words: information extraction, coreference resolution, relationship extraction, named entity recognition

ZAHVALA

Iskreno se zahvaljujem mentorju prof. dr. Marku Bajcu, ki je z nasveti glede raziskovanja in tudi širše najbolj pripomogel k uspešni izdelavi disertacije. Prav tako se še posebej zahvaljujem kolegu doc. dr. Lovru Šublju in sestri Marinki Žitnik, saj sem od njiju vedno pridobil uporabne nasvete in usmeritve. Navsezadnje bi se želel zahvaliti še mami in očetu ter vsem ostalim, ki so kakorkoli prispevali k uspešnemu zaključku mojega študija.

— Slavko Žitnik, Ljubljana, december 2014.

KAZALO

<i>Povzetek</i>	<i>i</i>
<i>Abstract</i>	<i>iii</i>
<i>Zahvala</i>	<i>v</i>
1 <i>Uvod</i>	1
1.1 Motivacija	2
1.2 Pregled vsebine	3
1.3 Prispevki k znanosti	4
2 <i>Opis področja in razlaga pojmov</i>	7
2.1 Ekstrakcija informacij	8
2.1.1 Prepoznavanje imenskih entitet	10
2.1.2 Ekstrakcija povezav	12
2.1.3 Odkrivanje koreferenčnosti	13
2.2 Ekstrakcija informacij s pomočjo ontologije	14
2.3 Ekstrakcija informacij v slovenskem jeziku	14
2.4 Pristopi za ekstrakcijo informacij	15
2.4.1 Pogojna naključna polja	16
3 <i>Odkrivanje koreferenčnosti</i>	23
3.1 O koreferenčnosti	24
3.1.1 Umestitev v sorodno delo	24
3.2 Metoda za odkrivanje koreferenčnosti	26
3.2.1 Predstavitev podatkov	26

3.2.2	Zaporedja izpuščenih omenitev	28
3.2.3	Primer odkrivanja koreferenčnosti	31
3.2.4	Sistem SkipCor	35
3.3	Analiza rezultatov	38
3.3.1	Ocene uspešnosti	38
3.3.2	Opis podatkovnih množic	40
3.3.3	Preizkus na označenih podatkovnih množicah	41
3.3.4	Pričakovana točnost na splošnih podatkih	46
3.3.5	Ročni pregled odkritih koreferenčnosti	47
3.4	Sklep	51
4	<i>Ekstrakcija povezav</i>	55
4.1	O ekstrakciji povezav	56
4.1.1	Umestitev v sorodno delo	56
4.2	Predstavitve podatkov	59
4.2.1	Atributske funkcije	61
4.2.2	Primer ekstrakcije povezav	61
4.3	Metoda za ekstrakcijo povezav	63
4.3.1	Predprocesiranje	66
4.3.2	Razpoznavanje omenitev	66
4.3.3	Ekstrakcija dogodkov	66
4.3.4	Ekstrakcija povezav	66
4.3.5	Uporaba ročno definiranih pravil	69
4.3.6	Čiščenje rezultatov	69
4.4	Analiza rezultatov	71
4.4.1	Ocenjevalni kriterij	71
4.4.2	Podatkovna množica BioNLP 2013	71
4.4.3	Rekonstrukcija genskega regulatornega omrežja	72
4.4.4	Pregled ekstrakcij glede na procesiranje	74
4.4.5	Uspešnost ekstrakcije glede na atributske funkcije	76
4.5	Sklep	76

5	<i>Prepoznavanje imenskih entitet</i>	79
5.1	O prepoznavanju imenskih entitet	80
5.1.1	Umestitev v sorodno delo	80
5.1.2	Tekmovanje CHEMDNER 2013	81
5.2	Metoda za razpoznavanje imenskih entitet	82
5.2.1	Predstavitev podatkov	82
5.2.2	Opis predlaganega sistema	84
5.3	Analiza rezultatov	89
5.3.1	Ocenjevalni kriterij	89
5.3.2	Podatkovna množica CHEMDNER 2013	90
5.3.3	Uspešnost razpoznavanja zdravil in kemijskih spojin	90
5.3.4	Analiza rezultatov glede na model učenja	92
5.3.5	Uspešnost razpoznavanja glede na atributske funkcije	93
5.4	Sklep	93
6	<i>Iterativna ekstrakcija informacij s pomočjo ontologije</i>	99
6.1	O ekstrakciji informacij	100
6.1.1	Umestitev v sorodno delo	100
6.2	Predstavitev podatkov	103
6.2.1	Pogojna naključna polja	103
6.2.2	Odkrivanje koreferenčnosti	105
6.2.3	Ekstrakcija povezav	109
6.2.4	Prepoznavanje imenskih entitet	110
6.3	Sistem za iterativno ekstrakcijo informacij	111
6.4	Analiza rezultatov	113
7	<i>Zaključki</i>	117
7.1	Zaključki	118
A	<i>Slovarček izrazov</i>	121
A.1	Angleško-slovenski slovar izrazov	122
	<i>Literatura</i>	125

Uvod

1.1 Motivacija

Podatki in informacije so bili od nekdaj pomembni in cenjeni. V starih časih brez računalniških tehnologij in povezav so z njimi rokovali predvsem učenjaki v knjižnicah, cerkvah ali pod vplivom bogatih mecenov. Prav tako je bilo pomembno obveščanje o nasprotnikih pred morebitnim vojskovanjem, kar je še vedno zelo aktualno. V dobi združenih tehnologij želijo tajne agencije preko interneta nadzorovati naše vsakodnevno življenje in odkrivati nevarnosti, ki pretijo človeštvu. Semantični obdelavi ogromnih količin zajetih podatkov se zato namenja čedalje več denarja in nekatera izmed najbogatejših podjetij na svetu se prav tako ukvarjajo z obdelavo ali hranjenjem naših podatkov, pa naj bodo to sporočila, dokumenti, slike, zvočni, video posnetki ali drugi senzorski podatki. V času internetne revolucije lahko torej govorimo, da so podatki novo črno zlato – “Data is the new oil” (Clive Humby, 2006). To so sicer vedno bili, a nikoli tako očitno kot v zadnjih desetletjih.

Vizionar Vannevar Bush si je že sredi prejšnjega stoletja zamislil stroj z imenom *memex*, ki bi na majhnem mestu – v obliki mize – hranil ogromne količine podatkov in omogočal hitro poizvedovanje po zelenih podatkih [1]. Menim, da so njegovi problemi že dobro rešeni, vendar informacijsko poizvedovanje potrebuje preporod [2] na način, da uporabniku ne bo treba samemu “rudariti” po zadetkih za njegovo informacijsko zahtevo. To bi namesto njega storil računalnik in mu ekstrahirane podatke pokazal v obliki strukturiranega miselnega vzorca.

Trenuno dostopni iskalniki (tj. sistemi za informacijsko poizvedovanje) za podano zahtevo vrnejo le seznam zadetkov s krajšimi izvlečki iz dokumentov (slika 1.1 levo). Nekateri iskalniki uporabljajo tudi semantične podatkovne zbirke in poleg rezultatov prikažejo še delček strukturiranih podatkov, ki pa so ročno vzdrževani. Razvoj dobrih tehnologij za ekstrakcijo informacij bi omogočil, da sistem sam prebere dokumente in nam jih pokaže v strukturirani obliki, od koder lahko navigiramo dalje (slika 1.1 desno). Ekstrakcija informacij – informacij zato, ker mora sistem podatke najprej pridobiti in jih nato semantično obdelati, da jih predstavi v pravi obliki – mora zato poleg sintaktičnih in leksikalnih analiz besedil omogočati višjenivojske funkcije prepoznavanja imenskih entitet, ekstrakcije povezav in odkrivanja koreferenčnosti. Čeprav termin ekstrakcija informacij pomeni pridobivanje strukturiranih podatkov iz poljubnih virov, ki so lahko zvok, slika, video ali kaj drugega, se v okviru te disertacije osredotočamo na besedilne vire.

Zooglo - Traditional
http://tradicionalni-iskalnik.si

Ivan Cankar

666 najdenih zadetkov:

[Ivan Cankar - Wikipedia, prosta enciklopedija](#) ★★★★★
Ivan Cankar se je rodil v hiši Na Klancu 141, kot eden od dvanajstih otrok obrtniško-proletarske družine. Leta 1882 se je vpisal v osnovno ...

[Ivan Cankar – največji mojster slovenske besede](#) ★★★★★★
Ivan Cankar se je rodil na Vrhniki (na Klancu) 10. maja 1876 kot osmi otrok v propadajoči obrtniško – proletarski družini irškega krojača ...

[Cankarjeva smrt je bila političen umor](#) ★★★★★★
Cankarjeva smrt je bila političen umor. Pisatelj, politik in ljudski tribun Ivan Cankar Ivan Cankar velja za največjega slovenskega pisatelja in ...

[\[PDF\] Ivan Cankar \(1876 - 1918\)](#) ★★★★★★
Ivan Cankar (1876 - 1918). Največji mojster slovenske besede in osrednja postava v moderni književnosti izvira iz revne družine ...

[ŽIVLJENJEIS Ivan Cankar](#) ★★★★★★
Ivan Cankar se je rodil 10. maja leta 1876 v kmečko družino na Vrhniki. V družini je bilo osem otrok. Zapustil je družino. Ker je bil zelo ...

[Ivan Cankar memorial house](#) ★★★★★★
Ivan Cankar memorial house. Ivan Cankar (1876 - 1918) is considered to be Slovenia's most important writer. The original house ...

1 | 2 | 3 | 4 | 5 | ... | Zadnja

Zooglo - Semantic
http://semantični-iskalnik.si

Ivan Cankar

Informacije ekstrahirane iz 25 zadetkov, najdenih 666:

[Ivan Cankar - Wikipedia, prosta enciklopedija](#) ★★★★★
Ivan Cankar se je rodil v hiši Na Klancu 141, kot eden od dvanajstih otrok obrtniško-proletarske družine. Leta 1882 se je vpisal v osnovno ...

[Ivan Cankar – največji mojster slovenske besede](#) ★★★★★★
Ivan Cankar se je rodil na Vrhniki (na Klancu) 10. maja 1876 kot osmi otrok v propadajoči obrtniško – proletarski družini irškega krojača ...

Slika 1.1

Informacijsko poizvedovanje. Tradicionalno iskanje (levo), semantično obogateno iskanje z avtomatsko ekstrakcijo informacij iz zadetkov (desno).

Raziskovalci so že analizirali tipe podatkov na spletu, se ukvarjali z različnimi problemi ekstrakcije [3, 4], vendar trenutno znane metode še niso dovolj zrele za izvedbo zastavljenega cilja [5–8]. Zaradi tega bomo v pričujoči disertaciji poskusili izboljšati glavne naloge ekstrakcije informacij (tj. prepoznavanje imenskih entitet, ekstrakcija povezav, odkrivanje koreferenčnosti) z uporabo enotnega in enostavnega algoritma ter skušali vse tri naloge povezati v soodvisno iterativno metodo, ki bi še dodatno izboljšala rezultate. Poleg tega bomo zaradi večje interoperabilnosti ekstrahiranih podatkov v prihodnje uporabili ontologijo, ki omogoča semantično obogateno poizvedovanje.

1.2 Pregled vsebine

Uvodnemu poglavju pričujoče disertacije sledi krajši opis področja ekstrakcije informacij in dosedanjega raziskovalnega dela na izbranem področju, ki hkrati definira potrebno podlago za boljše razumevanje nadaljnje vsebine. Nato sledijo štiri jedrna poglavja, od katerih prva tri opišejo po eno izmed glavnih nalog ekstrakcije informacij in zadnje združitev vseh nalog v sistem za celostno ekstrakcijo informacij. Najprej opišemo predlagano metodo za odkrivanje koreferenčnosti (poglavje 3), kjer razložimo transformacijo zaporedij v zaporedja izpuščenih omenitev ter primerjamo predlagan sistem z ostalimi znanimi pristopi na različnih podatkovnih množicah. Nadaljujemo z opi-

som predlagane metode za ekstrakcijo povezav (poglavje 4), s katero smo sodelovali na tekmovanju v odkrivanju genskih regulatornih mrež in dosegli najboljši rezultat. Tu ponovno uporabimo predhodno predlagana zaporedja izpuščenih omenitev, izgradimo več tipov modelov za ekstrakcijo hierarhičnih povezav in sistemu dodamo še modul za ekstrakcijo povezav z ročno definiranimi pravili. Kot zadnjo glavno nalogo ekstrakcije informacij predstavimo prepoznavanje imenskih entitet (poglavje 5). Tudi tu poskušamo izpuščati člene, vendar ne dosežemo izboljšav v primerjavi z znanimi pristopi, zato jih malenkost prilagodimo in kljub temu dosežemo zelo dobro sedmo mesto na tekmovanju v odkrivanju kemijskih spojin in zdravil. Nazadnje predstavimo še sistem za celostno in skupno ekstrakcijo informacij (poglavje 6) s pomočjo ontologije, kjer povežemo vse tri predhodno opisane postopke v iterativno metodo. Sistem na vходу sprejme nestrukturirano besedilo in vrne semantični graf označenih entitet na podlagi sistemske ontologije. V fazi evalvacije pokažemo, da se uspešnost ekstrakcije posamezne naloge tekom iteracij izboljšuje. V dodatku A priložimo še slovarček prevodov angleški izrazov, ki so povezani z ekstrakcijo informacij, saj je v slovenskem jeziku področje še zelo novo in zato smo nekatere izraze predlagali sami.

1.3 Prispevki k znanosti

V disertaciji so predstavljeni naslednji prispevki k znanosti:

Metoda za odkrivanje koreferenčnosti in metoda za ekstrakcijo povezav z uporabo verjetnostnih modelov prvega reda. Predstavimo inovativno transformacijo zaporedij omenitev v zaporedja z izpuščenimi omenitvami. S tem omogočimo uporabo verjetnostnih modelov prvega reda za izvedbo celotne naloge, kar dokažemo z uporabo linearno-verižnih modelov pogojnih naključnih polj. Poleg evalvacije, ki pokaže, da naša metoda dosega enake ali boljše rezultate kot ostale znane metode, preverimo, kakšne so pričakovane razdalje med koreferenčnimi omenitvami v dokumentih in z evalvacijo modelov nad drugačnim tipom podatkovne množice pokažemo, kakšno je pričakovano delovanje metode na realnih podatkih. Podobno kot pri odkrivanju koreferenčnosti tudi pri ekstrakciji povezav preverimo pričakovane razdalje med atributi povezav in predlagamo podobne transformacije omenitev. Z drugačnim označevanjem zaporedij in uporabo linearno-verižnih modelov pogojnih naključnih polj predstavimo metodo, s katero lahko iz besedila ekstrahiramo povezave. Predlagano metodo z več raz-

ličnimi tipi označevanj in s pomočjo ročno definiranih pravil uspešno testiramo na biomedicinskih besedilih, kjer dosežemo najboljše rezultate. Obe metodi sta podrobneje predstavljeni v poglavjih 3 in 4.

Celostna in iterativna ekstrakcija informacij. Predlagamo inteligentno ekstrakcijo informacij, ki na vходу sprejme besedilo ter kot rezultat vrne ekstrahirane entitete s povezavami med njimi. Z iterativno metodo združujemo vse ti glavne naloge ekstrakcije informacij – prepoznavanje imenskih entitet, ekstrakcijo povezav in odkrivanje koreferenčnosti, pri čemer za vsako od nalog uporabimo linearno-verižne modele pogojnih naključnih polj. Rezultati kažejo, da se upoštevanje vmesnih rezultatov iz predhodnih iteracij splača, saj se natančnost modelov skozi tri iteracije izboljša. Ontologijo v sistemu uporabljamo kot shemo podatkov za ekstrakcijo, množico znanih primerkov in za definicijo pravil, preko katerih lahko uporabnik vpliva na sistem. Lahko bi uporabili tudi drugačen tip predstavitve podatkov, vendar ta omogoča večjo interoperabilnost ekstrahiranih podatkov. Sistem podrobneje opišemo v poglavju 6.

V prijavi teme doktorske disertacije smo zapisali, da bomo izdelali tudi podatkovno množico z označenimi imenskimi entitetami v slovenskem jeziku. V času našega raziskovanja so slovenski raziskovalci že izdelali podatkovno množico imenskih entitet *ssj500k* [9], ki vsebuje označena osebna, zemljepisna in stvarna imena, ter nad njo izgradili model za prepoznavanje lastnih imen [10]. Na podlagi njihovih ugotovitev, analiz za ostale slovanske jezike in lastnih preskusov zato v razdelku 2.3 opišemo še nekaj razlik in problemov pri ekstrakciji informacij med slovenskim in angleškim jezikom.

Deli disertacije so bili predstavljeni na mednarodnih znanstvenih konferencah [11–14], revijah [15–18] ali v poglavju v knjigi [19]. Vsa koda, s katero smo pokazali delovanje metod, je dostopna v javnem programskem repozitoriju [20].

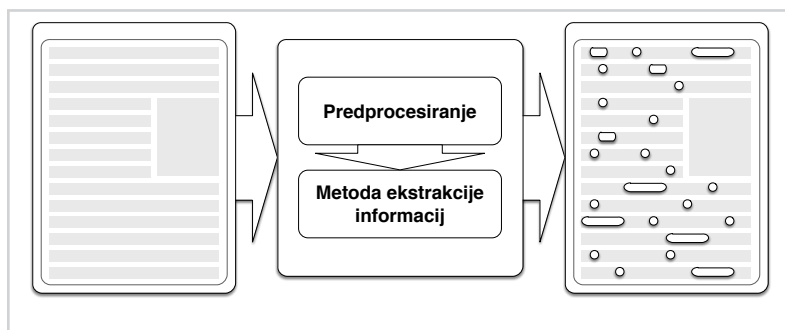


Opis področja in razlaga pojmov

2.1 Ekstrakcija informacij

Področje ekstrakcije informacij (angl. information extraction) [3, 21–24] se je naglo začelo razvijati v 1970-ih letih, ko so se začeli pojavljati prvotni sistemi za razpoznavanje imenskih entitet [25]. Od tedaj je bilo predlaganih mnogo sistemov, ki so se poleg imenskih entitet osredotočali na ekstrakcijo povezav, odkrivanje koreferenčnosti in pri tem uporabljali tudi semantične tehnologije [26, 27]. Glavni cilj ekstrakcije informacij je pridobiti strukturirane podatke iz nestrukturiranih ali pol-strukturiranih podatkovnih virov (tj. viri, ki vsebujejo del strukturiranih in del nestrukturiranih podatkov). Med viri se najpogosteje pojavljajo besedila, čeprav je vir v splošnem lahko tudi zvok, video ali slika. Ekstrahirani podatki nato omogočajo nove načine poizvedovanja, organizacije, analize in predstavitve podatkov. Pred kratkim so se nagleje pričeli razvijati pristopi za ekstrakcijo informacij na biomedicinski domeni [19, 28], kjer je glavni problem odkriti že objavljene poskuse, zakonitosti in znanje v znanstvenih člankih z namenom hitrejšega izvajanja novih poskusov ali sklepanja o novih rezultatih.

Na sliki 2.1 prikazujemo generičen pristop k ekstrakciji informacij. Vhodni dokument moramo vedno predstaviti v primerni obliki in ga obogatiti, kar naredimo v fazi predprocesiranja. S tem poleg osnovnih lastnosti besedila pridobimo še dodatne značilke, kot so na primer leme ali oblikoslovne oznake, ki zelo pripomorejo k boljšemu rezultatu v fazi ekstrakcije. Nad obogatenimi podatki nato izvedemo želeno metodo ekstrakcije, ki ustrezno označi dele besedila, ki nas zanimajo. Ti deli so lahko imenske entitete, povezave med omenitvami, koreferenčnosti ali kaj drugega.



Slika 2.1

Splošen pristop k ekstrakciji informacij. Vhodni dokument najprej v fazi predprocesiranja pretvorimo v ustrezno obliko in obogatimo z značilkami, nato izvedemo metodo ekstrakcije ter vrnemo rezultate.

Predprocesiranje pred izvedbo ekstrakcije je zelo pomembno, saj nam omogoči vpo-

gled v dodatne lastnosti besedila, med katerimi lahko odkrivamo zakonitosti za boljše natančnost ekstrakcije [21]. Večina metod za predprocesiranje je zelo odvisnih od izbranega jezika, saj za vsakega veljajo na primer drugačna sintaktična pravila, tipična zaporedja skladenjskih tipov, sklanjanje, upoštevanje spola ali števila. Prosta besedila navadno vedno najprej razbijemo v povedi in člene. Čeprav se zdi naloga enostavna, velja veliko pravil za ustrezno razbitje. Na primer vhodni dokument "Mojca je srečala g. Janeza v Merkurju." lahko algoritem razbije v dve povedi (poved se konča s končnim ločilom in nato se naslednja prične z veliko začetnico), s čimer bi izgubili možnost za pravilno razpoznavanje povezav. V posamezni povedi nato razpoznamo člene, ki navadno predstavljajo besede, ločila in posebne simbole. Takšna predstavitev je navadno že dovolj za izvajanje nadaljnjih analiz, vendar boljše rezultate dosežemo z dodatnimi obogatitvami. Zaradi tega uporabimo še oblikoslovne označevalnike [29, 30], razčlenjevalnike odvisnosti [31], lematizatorje ali plitke razčlenjevalnike odvisnosti [32]. Nekatere izmed teh metod so lahko časovno zahtevne in tudi enako težke za izvedbo kot izbrana ekstrakcija, zato zanje uporabljamo že pripravljene knjižnice [31, 33, 34]. V nekaterih primerih lahko namesto prostih besedil kot vhod dobimo spletno stran, dokument PDF ali kakšen drug tip dokumenta, ki ga moramo pred analizo še dodatno obdelati, da ugotovimo, kje se pojavlja besedilo, nad katerim želimo izvesti ekstrakcijo informacij [35–37].

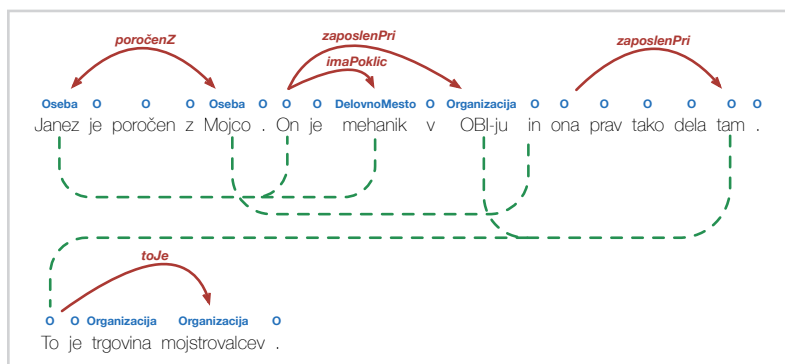
Evalvacija metod za ekstrakcijo informacij je bila pogosto otežena, saj je ročno označevanje reprezentativne množice besedil počasno in drago. Največ podatkovnih množic in sistemov za ekstrakcijo je bilo zato pripravljenih v okviru tekmovanj oziroma konferenc, ki so naslavljale določen problem ekstrakcije. Prva je bila serija sedmih konferenc Message Understanding Conference (MUC), ki je v letih od 1987 (MUC-1) do 1997 (MUC-7) skrbela za pripravo podatkovnih množic in javno evalvacijo za različne naloge ekstrakcije. Trenutno še vedno najbolj aktualna podatkovna množica ACE 2004 [38] vsebuje različne tipe besedil v več jezikih in označene podatke za vse tri glavne naloge ekstrakcije informacij. Kasneje so bile organizirane tudi konference s tekmovanji, kot sta na primer Semantic Evaluation (SemEval 2010) [39] ali Conference on Natural Language Learning (CoNLL) [40], ki so se osredotočale le na eno izmed nalog ekstrakcije. Podobne dogodke so kasneje organizirali tudi na biomedicinskem področju. Tam so namesto splošnih entitet, kot so osebe in organizacije, odkrivali gene, proteine ali interakcije med njimi. Najbolj znana tekmovanja na biomedicinskem področju so BioCreative I in II [41], tekmovanje LLL [42] in serija tekmovanj

BioNLP [28].

Na sliki 2.2 prikazujemo vse tri glavne naloge ekstrakcije informacij, ki so razpoznavanje imenskih entitet (modro), ekstrakcija povezav (rdeče) in odkrivanje koreferenčnosti (zeleno). Vsako od teh nalog podrobneje opišemo v nadaljnjih poglavjih (poglavja 3, 4 in 5). Pri razpoznavanju imenskih entitet kot vhod vzamemo poved, ki je sestavljena iz posameznih členov (tj. besed ali simbolov) in za vsak člen posebej napovemo njegovo oznako. Oznake so lahko tipi entitet (npr. oseba, lokacija ali organizacija) ali druge pomenške oznake, ki jih želimo napovedovati. Ekstrakcija povezav želi razpoznati tipe povezav (npr. je poročen z, živi v), ki veljajo med delčki besedila – omenitvami. Omenitev je ena ali več besed, ki označujejo neko entiteto ali stvar. Na primer omenitvi *Janez* in *on* se lahko v besedilu obe sklicujeta na isto entiteto, ki je oseba Janez. Povezave pa predstavljajo lastnosti, ki veljajo med entitetami in jih želimo razbrati iz besedila. Pri odkrivanju koreferenčnosti pa želimo ugotoviti, katere omenitve v dokumentu se sklicujejo na iste entitete.

Slika 2.2

Prikaz glavnih nalog ekstrakcije informacij. Razpoznavanje imenskih entitet vsak člen označi z ustrežno oznako (modra), ekstrakcija povezav s tipom povezave med seboj poveže omenitve (rdeče) in odkrivanje koreferenčnosti identificira omenitve na isto entiteto (zeleno).



2.1.1 Prepoznavanje imenskih entitet

Eden prvih člankov na področju prepoznavanja imenskih entitet je bil objavljen v letu 1991, ko je raziskovalka Lisa F. Rau [43] predstavila sistem za avtomatsko ekstrakcijo in prepoznavanje imen poslovnih subjektov. Termin imenska entiteta (angl. named entity) se na področju ekstrakcije informacij pogosto uporablja za označevanje členov z njihovimi tipi. Izraz se je prvič pojavil v letu 1996 v okviru konference MUC-6 [44], ko je bila glavna naloga razpoznavati imena podjetij in vojaške aktivnosti iz nestruk-

turiranih besedil, kot so časopisni članki [45]. Naloga je bila tedaj identificirana kot ena glavnih nalog ekstrakcije informacij in torej poimenovana prepoznavanje imenskih entitet (angl. named entity recognition and classification). V literaturi zasledimo tudi izraz ekstrakcija entitet (angl. entity extraction), ki ga nekateri napačno uporabljajo za poimenovanje te naloge, saj entiteta predstavlja gručo vseh sklicev na določen objekt ali predmet.

Kot primer prepoznavanja imenskih entitet vzemimo poved “Janez Novak nakupuje v Trgovini za male živali.”. Rezultat, ki ga želimo doseči, je za vsak člen v povedi določiti njen tip, torej “Janez/OSEBA Novak/OSEBA nakupuje/O v/O Trgovini/ORG za/ORG male/ORG živali/ORG ./O” (V primeru ORG pomeni organizacijo in O ostalo.). Nekateri raziskovalci so predlagali tudi bolj natančne sheme za označevanje členov, kjer posebej označijo prve in zaporedne člene imenskih entitet – BIO (angl. beginning-inside-outside) notacija: “Janez/B-OSEBA Novak/I-OSEBA nakupuje/O v/O Trgovini/B-ORG za/I-ORG male/I-ORG živali/I-ORG ./O”. Še bolj natančna notacija posebej označi imenske entitete z enim členom ali zadnje člene v entiteti – BILOU (angl. beginning-inside-last-outside-unit) notacija: “Janez/B-OSEBA Novak/L-OSEBA nakupuje/O v/O Trgovini/B-ORG za/I-ORG male/I-ORG živali/L-ORG in/O OBI-ju/U-ORG ./O”. Čeprav sta bili shemi v zgodnjih letih razvoja metod za prepoznavanje imenskih entitet zelo popularni in sta dosegali boljše rezultate v primerjavi z osnovnim označevanjem [46], se je takšno označevanje kasneje pri uporabi bolj naprednih atributskih funkcij izkazalo za nepotrebno.

Podobno kot vse raziskovalno delo na področju ekstrakcije informacij so tudi pri prepoznavanju imenskih entitet prvi sistemi delovali na osnovi ročno definiranih pravil in bili kasneje dopolnjeni ali zamenjani z metodami iz strojnega učenja [47]. Raziskovanje so najbolj pohitrili dogodki, kot so tekmovanja ali konference, v okviru katerih so pripravili ustrezne podatkovne množice in iskali najboljše postopke prepoznavanja. Takšni dogodki so na primer HUB-4 [48], MUC-7 in MET-2 [49], IREX [50], CONLL [51, 52], ACE [38] in HAREM [53]. Od leta 2000 dalje pa se večina raziskav predstavi v okviru konference LREC – The Language Resources and Evaluation Conference. Velika večina raziskav se nanaša na procesiranje angleškega jezika, ki mu sledijo kitajski jezik, francoščina, italijanščina ter ostali jeziki.

Izraz imenska entiteta pomeni določen del besedila, ki predstavlja lastno ime objekta oziroma ga eksplicitno določa [54]. Med raziskovalci velja konsenz, da k tem lastnim imenom prištevajo tudi časovne oznake (npr. imena mesecev) in nekatere numerične

vrednosti (npr. količina denarja). Sicer so glavna lastna imena, ki so tudi označena v podatkovnih množicah imena oseb, geografskih lokacij in organizacij, ki so lahko klasificirana tudi v različne podtipе. V primerjavi z angleščino se slovenska in nemška lastna imena delijo na osebna lastna imena in zemljepisna lastna imena [10]. Med lastna imena po slovenskem pravopisu uvrščamo tudi imena stvaritev, organizacij, delovnih skupnosti ipd. Prav tako posebno vrsto lastnih imen predstavljajo imena produktov, ki pa jih lahko uvrščamo tudi med obča imena. Ker v angleščini med lastna imena sodijo tudi imena dni, mesecev, praznikov, poimenovanja valut in odstotkov, pri teh metodah ne moremo govoriti le o razpoznavanju lastnih imen.

2.1.2 Ekstrakcija povezav

Ekstrakcija povezav (angl. relationship extraction, relation extraction) se ukvarja s pridobivanjem povezav med omenitvami v besedilih. V primerjavi s prepoznavanjem imenskih entitet je slabše rešen problem, saj moramo za uspešno izvedbo celotne naloge pravilno razpoznati omenitve in njihove tipe ter poleg tega še tipe povezav med njimi. Večinoma so povezave usmerjene in predstavljene z dvema atributoma, ki jima pravimo osebek in predmet (slika 4.1). Pri mnogo-teričnih povezavah (tj. povezujejo več členov) lahko upoštevamo še dodatne attribute, vendar je vsako takšno povezavo možno predstaviti z več binarnimi povezavami. Osnovni vhodni parameter pri ekstrakciji povezav predstavlja poved. Na primer iz povedi "Janez je poročen z Mojco." želimo ekstrahirati povezavo $JANEZ \rightarrow jePoročenZ \rightarrow MOJCO$. Nekateri raziskovalci problem ekstrakcije povezav imenujejo tudi semantično označevanje vlog (angl. semantic role labeling) [55].

Raziskovalci so pričeli z raziskovanjem ekstrakcije povezav proti koncu 1990-ih let. Prvi znan sistem za ekstrakcijo povezav – DIPRE [56] je vseboval začetno množico povezav, iz katerih je s pomočjo iskanj po internetu pridobil nova pravila in nato razširil povezave. Postopek temelji na širjenju semena, kjer je potrebno še vedno veliko ročnega nadzorovanja, da sistem primerno deluje. Poleg ročno definiranih pravil so se kasneje na področju ekstrakcije povezav najbolj uveljavile metode iz strojnega učenja na osnovi jeder [57, 58] (predvsem metoda podpornih vektorjev). Predlaganih je bilo mnogo postopkov, ki so definirali in skupno uporabljali različne tipe jeder. Zaradi najbolj zanesljivega in enostavnega napovedovanja večina pristopov pretvori problem v nalogo binarne klasifikacije, kjer posamezen klasifikator napove, ali med izbranimi omenitvama obstaja določen tip povezave ali ne [59].

Poleg standardnih postopkov ekstrakcije povezav se je vzporedno pojavilo področje odprte ekstrakcije informacij (angl. open information extraction), kjer je cilj iz besedila ekstrahirati čimveč trojčkov (osebek, povezava, predmet), pri čemer se ne osredotoča na tip povezave ali tip atributa. Takšni pristopi so navadno predlagani z metodami nadzorovanega učenja in pogosto povezani z avtomatsko gradnjo ontologij [2, 60–62]. Nekateri raziskovalci so se ukvarjali tudi z ekstrakcijo opisov povezav (angl. relationship descriptors) [63]. V teh primerih pa je pomembno razpoznati le zaporedje besed, ki predstavlja povezavo. V primerjavi z definicijo ekstrakcije povezav je to lažje, saj ni treba prepoznavati tipa povezave. Povezave so mnogokrat predstavljene v obliki povedka in problem lahko zaradi definicije predstavimo kot enostavno označevanje zaporedij. Našo zgornjo poved bi tako predstavili v obliki “Janez/O je/POVEZAVA poročen/POVEZAVA z/POVEZAVA Mojco/O . /O”

2.1.3 Odkrivanje koreferenčnosti

Odkrivanje koreferenčnosti (angl. coreference resolution, entity linking [64]) [65, 66] se ukvarja z gručenjem omenitev oziroma z združevanjem omenitev, ki se sklicujejo na isto entiteto [67]. Nekateri uporabljajo tudi izraz ekstrakcija entitet (angl. entity extraction), pri čemer razširijo nalogo in za gručo koreferenčnih omenitev določijo še njen tip. Od vseh treh glavnih nalog ekstrakcije je odkrivanje koreferenčnosti raziskovano najmanj časa in tudi najslabše rešeno. Eden izmed razlogov za to je tudi, da diskurz v besedilu ni vedno pravilno določen. Na primer iz dokumenta “Janez si je ogledoval novega Golfa. On ga je predstavil Marku. On ga je kupil.” ne moremo razbrati ali je Golfa kupil Marko ali Janez.

Pomembno je ločiti termin odkrivanje koreferenčnosti od odkrivanja anafor (angl. anaphora resolution). Anafore so tip ponavljanj (grš. anafora = obnova) in so predvsem znane iz poezije, ko se eden ali več verzov začneja z istimi besedami. Medtem, ko pri odkrivanju koreferenčnosti stremimo za združevanjem omenitev v gručo, se pri odkrivanju anafor ukvarjamo z odkrivanjem predhodne omenitve z enakim pomenom [68]. Na primer v povedih “Janez je brnil žogo. Ta je odletela daleč stran.” beseda *Ta* predstavlja anaforo, ki se navezuje na žogo. Poleg odkrivanja anafor je pri odkrivanju koreferenčnosti tudi zelo uporabno upoštevati antonomazijo ali zamenjavo imen (angl. apposition). Na primer v stavku “Mojca, Janezova žena, dela v OBI-ju.” Mojco še dodatno definiramo s terminom *Janezova žena*, kar lahko zaradi lepšega sklicevanja na Mojco uporabljamo tudi v nadaljevanju.

Naloga odkrivanja koreferenčnosti je v grobem razdeljena na (a) identifikacijo omenitev in (b) gručenje omenitev. Omenitve so lahko imenske (npr. Janez), nominalne (npr. moški s klobukom) ali so predstavljene v obliki zaimkov (npr. on) [69]. Raziskovalci se identifikaciji omenitev niso zelo posvečali in so jih identificirali s pravili s pomočjo razpoznanih samostalnikov v drevesnicah odvisnosti [70]. Pri gručenju omenitev pa največ pristopov predstavi problem odkrivanja koreferenčnosti kot binarno klasifikacijo med dvema omenitvama, na podlagi katere se odločijo ali sta omenitvi koreferenčni ali ne. Poleg tega z različnimi tehnikami filtrirajo podatke, da ni treba pregledovati vseh n^2 parov. Kljub temu, da je bilo predstavljenih mnogo različnih tehnik z uporabo metod strojnega učenja [71], predlaganih veliko atributskih funkcij [72–74], se nekateri nenadzorovani sistemi ali sistemi, ki temeljijo na pravilih [70], na vnaprej definirani domeni odrežejo enako dobro. Najbolj znani dogodki, ki so vzpodbudili večje zanimanje za raziskave na tem področju, so MUC-7 [49], ACE 2004 [38] ter seriji evalvacij CoNLL [40] in SemEval [39].

2.2 Ekstrakcija informacij s pomočjo ontologije

V začetku 21. stoletja je Tim Berners Lee predstavljal idejo semantičnega spleta [75]. To je splet, v katerem so vsi dokumenti med seboj poleg navadnih hiperpovezav povezani s pomenskimi povezavami. Torej, da je besedilo označeno na podoben način, kot je to storjeno pri ekstrakciji informacij. Popolnoma jasno je, da nihče ne bo pisati besedila in ga nato še ročno označevati, zato se je ekstrakcija informacij hitro razpoznala kot avtomatična rešitev [76] problema.

Raziskovalci so predlagali kar nekaj platform in sistemov, ki pri ekstrakciji uporabljajo ontologijo in jo sproti dopolnjujejo [26, 77]. Ontologija s primerki lahko tako služi kot shema, s katero sistemi ekstrahirajo podatke, kot leksikon znanih primerkov ali kot množica pravil [78]. Nekateri (predvsem nenadzorovani) pristopi so bili izdelani za avtomatsko gradnjo ontologij [76, 79], vendar so zaradi slabih natančnosti in napak bolj uporabni kot pomoč pri raziskovanju množice besedil [80].

2.3 Ekstrakcija informacij v slovenskem jeziku

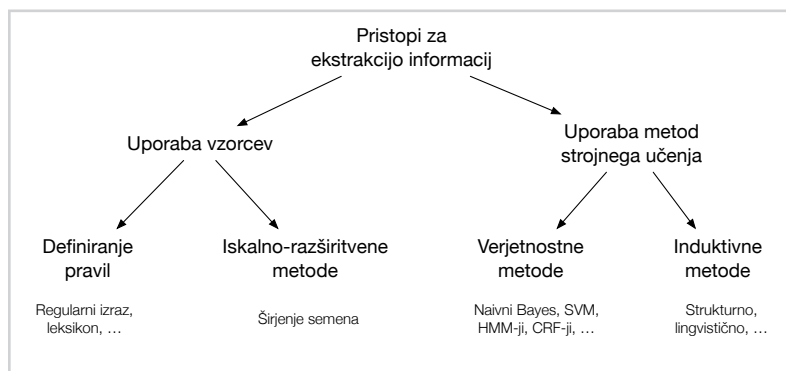
Precej manj ljudi za sporazumevanje uporablja slovenski jezik kot angleškega. Kljub temu je razvitih nekaj orodij, ki bodo omogočila boljše ekstrakcijo informacij. Slovenščina je v primerjavi z angleščino precej bolj kompleksna, predvsem za specifične

probleme, ki jih moramo rešiti v predprocesiranju. Pravilno moramo modelirati število, spol, sklon, implicitne zaimke ipd. Z enakimi problemi se ukvarjajo tudi ostali slovanski raziskovalci, pri čemer pa upajo, da bodo glavne tri naloge ekstrakcije informacij lahko bolje rešene, saj so vsi slovanski jeziki v primerjavi z angleškim veliko bolj izrazni [81].

Za slovenščino je bilo razvitih že nekaj uspešnih metod za predprocesiranje besedil, kot so na primer lematizator [82], oblikoslovni označevalnik [83] ali skladenjski razčlenjevalnik [84]. Razvit je bil tudi semantični leksikon sloWNet [85], ki je zgrajen podobno kot angleški WordNet [86]. Pred kratkim je bila izdelana podatkovna množica z označenimi lastnimi osebnimi in zemljepisnimi imeni [9]. Nad njo je bil zgrajen tudi standarden model za prepoznavanje imenskih entitet, ki je dosegel primerljive rezultate z angleškim jezikom [10]. Največ novih raziskav na področju ekstrakcije informacij za slovenski jezik se na dve leti predstavi na Konferenci za jezikovne tehnologije [87].

2.4 Pristopi za ekstrakcijo informacij

Za namene ekstrakcije informacij ali reševanje specifičnih problemov ekstrakcije je bilo uporabljenih mnogo pristopov [21, 22, 24], ki jih v osnovi delimo na pristope z uporabo vzorcev in pristope z uporabo metod strojnega učenja (slika 2.3).



Slika 2.3

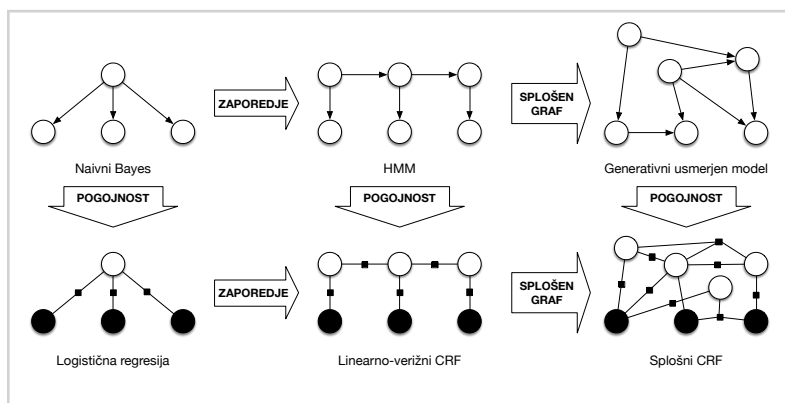
Razdelitev pristopov za ekstrakcijo informacij glede na tip uporabljenih metod.

Sistemi za ekstrakcijo informacij, ki uporabljajo pristop z uporabo vzorcev zahtevajo ročno delo domenskih ekspertov, ki definirajo pravila ali regularne izraze za ekstrakcijo. Takšni pristopi so zaradi definicije natančnih pravil enostavni za interpretacijo

in razvoj. Z namenom razbremenitve ročnega dela lahko s poenostavljenimi polavtomatskimi pristopi najprej iz izbranih besedil ročno izberemo množico znanih entitet (tj. seme) in nato avtomatsko pregledamo, kako se v besedilu pojavljajo, da izgradimo pravila za nadaljnje ekstrakcije.

Pristopi z uporabo metod strojnega učenja nasprotno od ročne definicije pravil zahtevajo pripravo besedil, ki vsebujejo označene dele z entitetami. Nato nad temi besedili avtomatsko izdelamo model ekstrakcije in ga uporabljamo za ekstrakcijo novih informacij. Takšni sistemi so bolj primerni, ko moramo razpoznavati entitete in povezave iz splošnih dokumentov, kjer ne moremo enostavno ročno definirati pravil.

V tej disertaciji vsakega od treh problemov ekstrakcije informacij predstavimo kot problem označevanja zaporedij in ustrezno uporabimo algoritem linearno-verižnih pogojnih naključnih polj (angl. linear-chain conditional random fields – linear-chain CRFs). Na sliki 2.4 [88, 89] prikazujemo izbran algoritem v primerjavi z ostalimi znanimi algoritmi v strojnem učenju. Njegova ključna lastnost je, da napoveduje oznake za celotno podano zaporedje simbolov (npr. besed), lahko sprejme in upošteva ogro- men nabor značilk ter je v primerjavi s splošnejšimi modeli bolj enostaven in hiter za učenje ter napovedovanje.



Slika 2.4

Podobnost med modeli strojnega učenja: naivni Bayes, logistična regresija, HMM, linearno-verižni CRF, generativni usmerjen model in splošni CRF.

2.4.1 Pogojna naključna polja

Pri mnogih problemih procesiranja zvoka, govora ali besedil želimo iz vhodnih signalov ekstrahirati semantično predstavitev vhodnih signalov. Natančna predstavitev vhodnih

in izhodnih podatkov je odvisna od izbranega problema, kjer vrstni red signalov v zaporedju močno vpliva na izhodne podatke. Na primer, v besedilu lahko beseda KLOP predstavlja več pomenov in je odvisna od konteksta oziroma zaporedja ostalih besed pred njo in za njo. Če začnemo poved z zaporedjem besed “*Na kožo se mi je prisesal KLOP*”. ali “*Sedel sem na KLOP*”, lahko glede na uvodno zaporedje besed sklepamo o vlogi nadaljnjih besed. Primeri vlog, ki jih želimo napovedovati glede na znano zaporedje, so torej oblikoslovne značilnosti (npr. spol, sklon ali skladenjski tip) ali semantični pomen (npr. oseba, žival ali stvar).

Verjetnostno modeliranje zaporedij s pogojnimi naključnimi polji (angl. conditional random fields – CRF) [90] je bilo prvič uporabljeno pri napovedovanju oblikoslovnih oznak, pri čemer so raziskovalci izboljšali dosedanje rezultate, pridobljene s pomočjo prikritih markovskih modelov (angl. hidden Markov models – HMM) in modelov največje entropije (angl. maximum entropy models – MEMM) [91]. Zatem so jih uspešno uporabili tudi na drugih področjih procesiranja jezika, kot je razpoznavanje entitet [92, 93], razčlenjevanje [94, 95], procesiranje v različnih jezikih [96–98] ter predlagali nove strukture in načine učenja ter napovedovanja z njimi [99–101]. V primerjavi z ostalimi pristopi, so CRF–ji primernejši zaradi sledečih lastnosti [102]:

- Omogočajo izgradnjo skupnega modela za označevanje kompleksnih zaporedij, kjer so ciljni členi med seboj odvisni. Na primer, pri izgradnji drevesnice odvisnosti je izračun posameznih komponent drevesa medsebojno odvisen, zato skupen model vrača boljše rezultate v primerjavi z uporabo več neodvisnih modelov.
- Omogočajo uporabo poljubnih značilk, pri čemer ni treba paziti na medsebojne odvisnosti. Poleg tega učinkovito delujejo z več milijoni podanih značilk, kar omogoča veliko prilagodljivost modela. Zaradi tako velike množice značilk lahko pride do prevelike prilagoditve modela učni podatkovni množici (angl. overfitting). To se rešuje z uporabo Gauss-ove (ali katere druge) porazdelitve vrednosti uteži pred učenjem ali omejitve števila korakov pri učenju modela, kar zmanjša verjetnost, da postanejo nekatere značilke preveč pomembne.
- So verjetnostni modeli, ki izračunavajo verjetnostne porazdelitve nad zaporedji oznak. Torej lahko za vsak označeno zaporedje vrnejo verjetnost pravilnosti za celotno zaporedje.

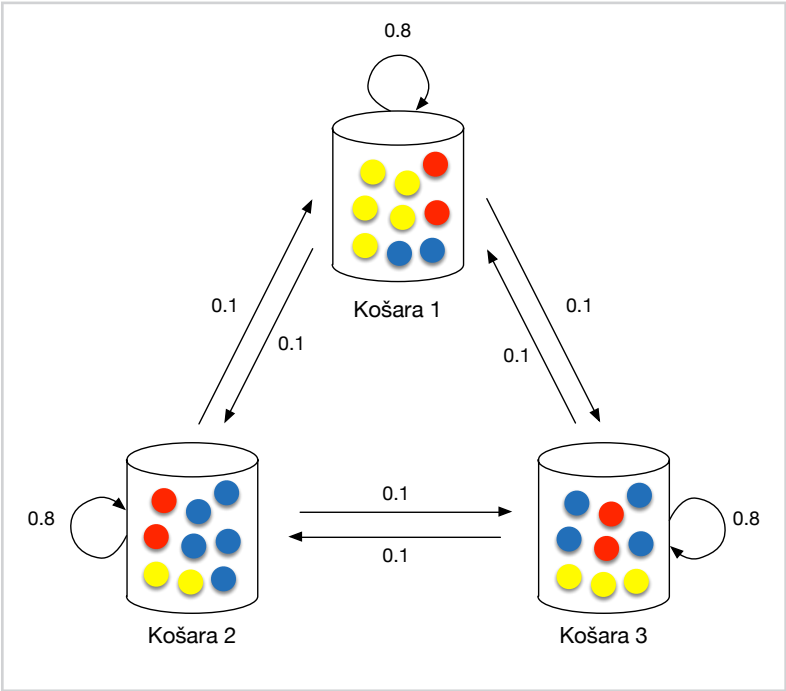
- Uvrščajo se med diskriminativne modele, to je, napovedujejo oznake glede na vhodno zaporedje členov. Hkrati z atributskimi funkcijami omogočajo zajem poljubnih lastnosti v podatkovni množici, ki bi jih bilo zelo zahtevno zajeti v podobnih modelih (npr. HMM).
- Operacije za izračun zaporedij oznak učinkovito maksimizirajo porazdelitev zaporedij oznak. Pri tem se uporablja dinamično programiranje s polinomsko časovno zahtevnostjo glede na dolžino zaporedja. Podoben postopek kot za označevanje se uporablja tudi pri učenju modelov.

CRF-ji so se razvili iz predhodno obstoječih HMM-jev in MEMM-jev. Rabiner je predstavil prikriti Markovske modele s primerom košar in žogic [103]: Naj se v n košarah nahaja nekaj žogic različnih barv, pri čemer se porazdelitve barv žogic po košarah razlikujejo. Na vsakem koraku izberemo žogico iz določene košare, preverimo njeno barvo in jo vrnemo v košaro. Nato z neko verjetnostjo izberemo naslednjo košaro in postopek ponavljamo. Med izvajanjem opazovalec vidi žogice – opazovane (angl. observable) simbole, medtem ko mu ni znano iz katere košare – prikrita (angl. hidden) stanja, je bila določena žogica izbrana (slika 2.5). Končen rezultat izvajanja algoritma je torej zaporedje prikritih stanj.

V pričujoči disertaciji bomo z $\bar{y} = [y_1, y_2, \dots, y_n]$ označili zaporedje prikritih stanj (tj. zaporedje oznak, ki jih želimo napovedati) in z $\bar{x} = [x_1, x_2, \dots, x_n]$ zaporedje opazovanih oznak (npr. zaporedje besed). HMM-ji modelirajo skupno verjetnost med prikritimi (y_i) in opazovanimi (x_i) oznakami, ki je predstavljena kot produkt verjetnosti prehodov med stanji – $P(y_i|y_{i-1})$ in verjetnosti opazovanih oznak pri izbranem skritem stanju – $P(x_i|y_i)$. Grafovska predstavitev modela HMM je predstavljena na sliki 2.6 (a).

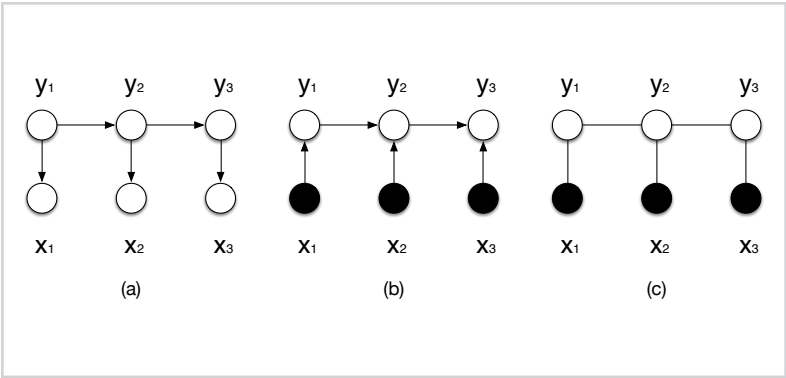
$$P(\bar{y}, \bar{x}) = \prod_{i=1}^n P(y_i|y_{i-1})P(x_i|y_i) \quad (2.1)$$

Pri modelu HMM predpostavimo, da so porazdelitve opazovanih vrednosti znane vnaprej – ko preverimo barvo izbrane žogice iz košare, jo vrnemo nazaj – zato je model HMM generativen. Z uporabo enačbe 2.1 na primeru žogic lahko iz slike 2.5 računamo verjetnostno porazdelitev za podana skupna zaporedja $\bar{y}$ in $\bar{x}$. Na primer $P(111222333, \text{yellow, yellow, blue, blue, red, yellow, red}) = 1,4 \cdot 10^{-7}$.



Slika 2.5

Problem košar in žogic. Tri košare so povezane z verjetnostmi prehodi med njimi in porazdelitvami žogic različnih barv. Na vsakem koraku iz košare izvlečemo naključno žogico, preverimo njeno barvo, jo vrnemo nazaj in nato glede na verjetnost prehodov izberemo naslednjo košaro. Začetno košaro izberemo z enakomerno porazdeljeno verjetnostjo. Rezultat predstavlja zaporedje prikritih stanj.



Slika 2.6

Grafovska predstavitev modelov (a) HMM – $P(\bar{y}, \bar{x})$, (b) MEMM – $P(\bar{y}|\bar{x})$ in (c) linerano-verižnih CRF – $P(\bar{y}|\bar{x})$. HMM-ji uporabljajo opazovane vrednosti (x_i) kot naključne spremenljivke z verjetnostnimi porazdelitvami, medtem ko jih MEMM-ji in CRF-ji uporabljajo kot fiksne vrednosti.

Pri problemu označevanj želimo najti prikrito zaporedje $\bar{y}$ glede na znano zaporedje $\bar{x}$, ki maksimizira pogojno verjetnost $P(\bar{y}|\bar{x}) = P(\bar{y}, \bar{x})/P(\bar{x})$. McCallum in sod. so glede na to dekompozicijo skupne verjetnosti opazili, da želimo napovedati posteriorno verjetnost prikrite oznake glede na prejšnjo prikrito oznako in trenutno opazovano vrednost [104]. Kot rezultat so predlagali model največje entropije – MEMM, ki je predstavljen z enačbo 2.2 in grafovsko predstavitevjo na sliki 2.6 (b). V primerjavi s HMM-ji so prikrita stanja odvisna od opazovanih vrednosti (spremenjene smeri pušče) in opazovane vrednosti so naprej znane – fiksirane (označene črno).

$$P(\bar{y}, \bar{x}) = \prod_{i=1}^n P(y_i | y_{i-1}, x_i) \quad (2.2)$$

V primerjavi z MEMM, model HMM predvideva, da so opazovane vrednosti neodvisno in enakomerno razporejene, kar ne drži v vseh primerih. Na primer, da želimo odkriti vzorec obnašanja, da jemljemo kroglice iz iste košare, dokler ne izberemo rdeče kroglice. Naslednjo kroglice potem izberemo iz ene od poljubno izbranih sosednih košar. V tem primeru, bo naučen model MEMM izračunal verjetnost $P(11, \text{red}, \text{rumen}) = 0$, medtem ko bi model HMM izračunal verjetnost $P(11, \text{red}, \text{rumen}) = 0,013$, kar je manj pravilno.

Glede na prejšnjo oznako in trenutno opazovano vrednost je težko izračunati napoved oznake. Zaradi tega so McCallum in sod. predlagali eksponentni model, ki uporablja utežene atributske funkcije [104]. Model je predstavljen v enačbi (2.3), kjer m predstavlja število atributskih funkcij in $Z(x_i, y_{i-1}) = \sum_{y_i'} \exp\left(\sum_{j=1}^m \lambda_j f_j(x_i, y_{i-1}, y_i')\right)$ normalizacijsko konstanto vseh možnih kombinacij, ki zagotavlja, da je vsota vseh verjetnosti enaka ena.

$$P(y_i | y_{i-1}, x_i) = \frac{\exp\left(\sum_{j=1}^m \lambda_j f_j(x_i, y_{i-1}, y_i)\right)}{Z(x_i, y_{i-1})} \quad (2.3)$$

Atributske funkcije so lahko zelo enostavne, kot na primer indikatorji prehodov med stanji, ali zelo kompleksne, ki lahko upoštevajo raznovrstne značilnosti opazovanih vrednosti ali uporabljajo dodatno znanje. Načrtovalec sistema za ekstrakcijo informacij mora poznati domeno in problem ekstrakcije, da lahko predlaga ustrezne atributske funkcije, saj morajo le-te odražati znanje izbranega problema. V času učenja modela nato učni algoritem poskrbi, da glede na učno podatkovno množico posameznim

atributskim funkcijam priredi ustrezno utež pomembnosti λ_j .

Na primer, za zgornji primer žogic in košar lahko za modeliranje verjetnosti $P(y_i = 1|y_{i-1} = 1 \wedge x_i = \bullet)$ definiramo ustrezno atributsko funkcijo (enačba (2.4)). Podobno lahko definiramo atributske funkcije za verjetnosti $P(y_i = 2|y_{i-1} = 1 \wedge x_i = \bullet)$, $P(y_i = 3|y_{i-1} = 1 \wedge x_i = \bullet)$ ipd. Sedaj lahko učni algoritem izračuna uteži za izbrane atributske funkcije $\lambda_1 = \log(0.89)$, $\lambda_2 = \log(0.04)$, $\lambda_3 = \log(0.07)$, ..., ki so zapisane v tabeli 2.1.

$$f_1(x_i, y_i, y_{i-1}) = \begin{cases} 1, & \text{if } x_i = \bullet, y_i = 1, y_{i-1} = 1 \\ 0, & \text{sicer.} \end{cases} \quad (2.4)$$

Tabela 2.1

Verjetnostne porazdelitve modela največje entropije za problem žogic in košar iz slike 2.5 in izbranih atributskih funkcij, ki modelirajo verjetnosti $P(y_i|y_{i-1} \wedge x_i)$.

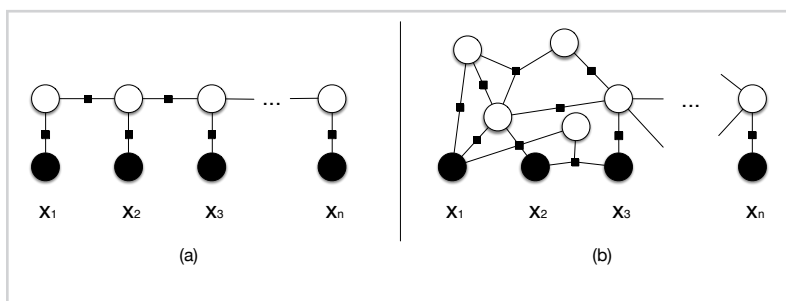
y_{i-1}	x_i	$P(y_i = 1 y_{i-1}, x_i)$	$P(y_i = 2 y_{i-1}, x_i)$	$P(y_i = 3 y_{i-1}, x_i)$
$\emptyset$	$\bullet$	0.5	0.2	0.3
$\emptyset$	$\circ$	0.18	0.45	0.36
$\emptyset$	$\circ$	0.33	0.33	0.33
1	$\bullet$	0.89	0.04	0.07
1	$\circ$	0.64	0.2	0.16
1	$\circ$	0.8	0.1	0.1
2	$\bullet$	0.21	0.67	0.12
2	$\circ$	0.04	0.87	0.09
2	$\circ$	0.1	0.8	0.1
3	$\bullet$	0.16	0.06	0.77
3	$\circ$	0.05	0.13	0.82
3	$\circ$	0.1	0.1	0.8

Pri učenju modelov MEMM lahko zaradi predstavitve verjetnosti $P(\bar{y}|\bar{x})$ s produktom posameznih verjetnosti $P(y_i|y_{i-1}, x_i)$ prihaja do problema pristranosti oznak [105]. To pomeni, da če obstaja zelo malo stanj y_i , ki lahko sledijo stanju y_{i-1} , potem vloga opazovane vrednosti x_i ne more bistveno vplivati na rezultat. V ekstremnem primeru, ko lahko samo eno stanje $y_i = a$ sledi stanju $y_{i-1} = b$, je $P(y_i = a|y_{i-1} = b, x_i) = 1 \forall x_i$ in posledično $P(y_i \neq a|y_{i-1} = b, x_i) = 0$, kar pomeni, da x_i sploh nima vpliva na

rezultat. Problem se pojavi zaradi tega, ker je vsako stanje posebej normalizirano, kar model pogojnih naključnih polj reši z globalno normalizacijo preko celotnega zaporedja. Pogojna naključna polja – CRF-je – so prvi predlagali Lafferty in sod.[90] in so definirana z enačbo 2.5 (linearno-verižni) ter predstavljena na sliki 2.6 (c). Vrednosti m_1 in m_2 predstavljata število atributskih funkcij in $Z(\bar{x})$ normalizacijsko konstanto, ki poskrbi, da model predstavlja pravilno verjetnostno porazdelitev. Model je hkrati tudi neusmerjen, saj so povezave med vozlišči definirane s tipom atributskih funkcij in ne z odvisnostmi med vozlišči.

$$P(\bar{y}|\bar{x}) = \frac{1}{Z(\bar{x})} \prod_{i=1}^n \left(\exp \left(\sum_{j=1}^{m_1} \lambda_j f_j(y_i, x_i) \right) \exp \left(\sum_{j=1}^{m_2} \lambda_j f_j(y_i, y_{i-1}, x_i) \right) \right) \quad (2.5)$$

Predstavljen model CRF (slika 2.6 (c)) je po strukturi enak modelu HMM (slika 2.6 (a)). Glede na definirane tipe atributskih funkcij je lahko grafovska predstavitev modela CRF poljubna (slika 2.7 (b)). Pri drevesnih predstavitvah modelov lahko izvedemo učenje in napovedovanje z enostavnimi postopki naprej-nazaj in vsota-produkt ali z algoritmi izmenjave oznak [106, 107]. Pri splošnejših nedrevesnih strukturah sicer lahko modeliramo poljubne odvisnosti, vendar se računska kompleksnost učenja in napovedovanja močno poveča ter zahteva uporabo aproksimacijskih algoritmov [89, 108–111]. Zaradi tega v nadaljevanju pričujočega dela predstavimo načine, kako lahko učinkovito rešujemo probleme ekstrakcije informacij le z uporabo linearno-verižnih modelov CRF.



Slika 2.7

Grafovska predstavitev (a) linearno-verižnih in (b) splošnih modelov pogojnih naključnih polj.

Odkrivanje koreferenčnosti

3.1 O koreferenčnosti

Odkrivanje koreferenčnosti želi identificirati vse dele besedila (tj. omenitve), ki naslavljajo isto entiteto. Poleg prepoznavanja imenskih entitet in ekstrakcije povezav predstavlja eno izmed glavnih nalog ekstrakcije informacij.

V tem poglavju predstavimo sistem za odkrivanje koreferenčnosti – *SkipCor*, ki pretvori osnovno nalogo v problem označevanja zaporedij. V primerjavi z obstoječimi nadzorovanimi ali nenadzorovanimi pristopi je *SkipCor* edini, ki uporablja le modele prvega reda in s tem omogoči hitro učenje, napovedovanje, visoko skalabilnost in možnost paralelizacije. Da omogočimo odkrivanje koreferenčnosti z izbranimi linearno-verižnimi modeli CRF, predlagamo transformacije osnovnega zaporedja omenitev v zaporedja z n izpuščenimi omenitvami, ki vsebujejo vsako $(n + 1)$ -to omenitev. *SkipCor* primerjamo z njegovima osnovnima izvedenkama, ki delujeta le nad osnovnim zaporedjem (tj. *SkipCorZero*) in nad pari omenitev (tj. *SkipCorPair*) ter z ostalimi predlaganimi sistemi. Evalvacijo izvedemo na sedmih podatkovnih množicah, kjer pokažemo, da predlagan pristop doseže boljše ali primerljive rezultate z ostalimi znanimi sistemi. Poleg tega raziščemo tudi pričakovan upad natančnosti, če modele naučimo nad enim tipom podatkovne množice in jo testiramo na drugem, kar je primerljivo z obnašanjem sistema na realnih problemih.

3.1.1 Umestitev v sorodno delo

Večina tehnik za odkrivanje koreferenčnosti predstavi problem kot binarno klasifikacijo, kar omogoča uporabo že znanih modelov strojnega učenja [73, 112]. Pri tem načinu algoritem za vsak par omenitev preveri, če sta koreferenčni (tj. se sklicujeta na isto entiteto). Nasprotno, nenadzorovane tehnike za odkrivanje koreferenčnosti temeljijo na upoštevanju zaporedij omenitev [70, 113] in so zaradi tega težje za vzdrževanje in slabše prilagodljive za sorodne probleme ali nove domene.

Eden izmed začetnih pristopov za odkrivanje koreferenčnosti s pomočjo metod nadzorovanega učenja je temeljil na odločitvenih drevesih in dvanajstih lokalnih atributskih funkcijah [71]. S tem je hkrati izboljšal rezultate prejšnjih metod, ki so temeljile le na ročno definiranih pravilih. Čeprav je bil pristop zelo enostaven, raziskovalci niso mogli izboljšati rezultatov le z uporabo bolj naprednih klasifikatorjev, kot so SVM [114], modeli največje entropije [115] ali markovska omrežja [116], ki so jih predlagali kasneje. Zaradi tega so nekateri pokazali, da je jedro uspeha v definiciji ino-

vativnih, lingvistično bogatih in globalnih atributskih funkcijah [72, 73]. Bengtson in Roth [72] sta jih celo sistematično razdelila v kategorije in demonstrirala njihovo pomembnost, s čimer sta pokazala, da je z uporabo primerno zasnovanih atributskih funkcij možno značilno izboljšati odkrivanje koreferenčnosti. Vsi predlagani sistemi so si med seboj zelo podobni, zato je bila zasnovana tudi platforma Reconcile [117], ki omogoča enotno predstavitev podatkov in evalvacijo novih pristopov.

Nenadzorovani pristopi ne potrebujejo učnih podatkov, vendar kljub temu dosegajo odlične rezultate. Haghighi in Klein [113] sta predlagala modularen in nenadzorovan sistem z bogatimi značilkami. Sistem je sestavljen iz treh delov, kjer prvi identificira sintaktične poti med omenitvami, nato oceni semantično ujemanje med njimi in izbere referenčne omenitve, ki služijo kot seme za nadaljnje razvrščanje omenitev. Lee in sod. [70] so izboljšali Raghunathan-ov sistem [118], ki temelji na večkratnih procesiranjih podatkovne množice. Predlagali so trinajst zaporednih procesiranj, ki so jih razvrstili padajoče po njihovi natančnosti. Vsako procesiranje nad podatkovno množico uporabi nekaj ročno definiranih pravil, ki temeljijo na sintaktičnih drevesnicah odvisnosti, imenskih entitetah, različnih heuristikah in lastnostih podatkovne množice. V primerjavi z odkrivanjem koreferenčnosti le med pari omenitev je bilo predlaganih tudi nekaj nenadzorovanih sistemov, ki odkrivajo koreferenčnost nad njihovimi seznami [119–121].

Na področju faktorskih grafov so McCallum in sod. [100] predlagali tri modele CRF za odkrivanje koreferenčnosti. Prvi je splošen model, zato je učenje in napovedovanje z njim kompleksno ter počasno. Drugi model deluje med pari omenitev in tretji model predstavi pare omenitev kot opazovana stanja v modelu. Wellner in sod. [101] so uporabili McCallum-ov prvi model in namesto odkrivanja koreferenčnosti združevali citate, kar je podoben problem. Najbolj podobno našemu pristopu je uporaba posebne vrste linearno-verižnih modelov CRF z dodanimi faktorji na nezaporednimi stanji (angl. skip-chain CRF) [93], vendar je njihovo učenje in napovedovanje še vedno počasnejše v primerjavi z osnovnim modelom. Culotta in sod. [112] so predlagali uporabo grafovskih modelov prvega reda za odkrivanje koreferenčnosti na podmnožicah omenitev v stilu hierarhičnega razvrščanja, torej njihov sistem deluje nad potencialnimi entitetami (tj. množicami omenitev) in ne ločeno nad omenitvami. Nazadnje so Sundar in sod. [122] predlagali sistem za odkrivanje koreferenc, kjer uporabijo dva tipa modelov CRF. Najprej s splošnim modelom CRF identificirajo omenitve s pomočjo drevesnice odvisnosti in nato uporabijo linearno-verižne modele CRF za odkrivanje

koreferenčnosti. Čeprav uporabijo le linearne modele, kot vhod jemljejo le pare omenitev in ne celega zaporedja omenitev, kot predlagamo v okviru te disertacije.

V tabeli 3.1 prikazujemo klasifikacijo pristopov za odkrivanje koreferenčnosti, ki so bili predlagani v literaturi. Predlagane sisteme delimo po tipu vhoda v algoritem in po tipu učenja modela. Glede na izbrano klasifikacijo je naš predlog sistema SkipCor drugačen od predhodno predlaganih. Nenadzorovani sistemi so bili razviti za razpoznavanje koreferenčnosti nad pari omenitev in prav tako nad njihovimi zaporedji. Nasprotno so nadzorovani pristopi večinoma razločevali le pare omenitev. Sistem [122] je podoben našemu osnovnemu pristopu SkipCorPair, ki kot vhod prejme le dve omenitvi in se odloči o njuni koreferenčnosti. Prav tako nekaj primernih modelov nad zaporedji omenitev predstavita McCallum in Wellner [100], vendar v okviru evalvacije uporabita le pristop, ki deluje na parih omenitev.

Tabela 3.1

Klasifikacija pristopov za odkrivanje koreferenčnosti. Glede na dimenziji vhoda v sistem in tip učenja modela, predlagan sistem SkipCor predstavlja novost.

	NENADZOROVANO	NADZOROVANO
ZAPOREDJE OMENITEV	[119–121]	SkipCor
PARI OMENITEV	[70, 113, 118] ipd.	[71–73, 100, 112, 114–116, 122]

3.2 Metoda za odkrivanje koreferenčnosti

V tem razdelku predstavimo algoritem SkipCor za odkrivanje koreferenčnosti. Najprej definiramo predstavitev podatkov, ki je primerna za uporabo linearno-verižnih modelov CRF. Nato na primeru besedila ugotovimo pomankljivosti v osnovni predstavitvi podatkov in jo nadgradimo z dodatnimi zaporedji izpuščenih omenitev. Potrebo po dodatnih zaporedjih utemeljimo s pregledom razdalj med zaporednimi koreferenčnimi omenitvami in nazadnje podrobno razložimo implementacijo predlaganega sistema [20] ter podamo njegovo časovno kompleksnost.

3.2.1 Predstavitev podatkov

Pri odkrivanju koreferenčnosti je pomembno, da ugotovimo, katere omenitve se sklicujejo na isto entiteto, torej sklepamo, da morajo med takšnimi omenitvami veljati nekakšne zakonitosti.

Zaradi uporabe linearno-verižnih modelov CRF smo omejeni le na predstavitev podatkov v obliki zaporedij (slika 2.6 (c)). Izbrani modeli poskušajo glede na verjetnostno porazdelitev čimbolj pravilno napovedati verjetnost celotnega zaporedja $P(\bar{y}|\bar{x}, \lambda)$, kjer je $\bar{y}$ želeno zaporedje glede na vhodno podano zaporedje $\bar{x}$ in predhodno naučene uteži atributskih funkcij λ . Za odkrivanje koreferenčnosti z zaporedjem $\bar{x}$ predstavimo zaporedje vseh omenitev v dokumentu in z zaporedjem $\bar{y}$ definiramo ali je omenitev x_i koreferenčna z omenitvijo x_{i-1} . Ciljne oznake $y_i \in \{C, O\}$ definiramo tako, da oznaka C določa koreferenčnost s predhodno omenitvijo in oznaka O določa nasprotno.

Atributske funkcije

Izbira ustreznih in informativnih značilk zelo vpliva na uspešnost učnih algoritmov v strojnem učenju. Atributske funkcije na področju procesiranja naravnih besedil navadno definiramo kot predloge in nato končne atributske funkcije (enačba (2.4)) pridobimo s prehodom čez podatkovno množico. V literaturi je bilo predlaganih kar nekaj atributskih funkcij [67, 71, 72, 123–125], ki jih razdelimo v sledeče kategorije in predlagamo dodatne:

Predprocesiranje. Atributske funkcije uporabljajo oznake iz predprocesiranja, kot so leme, oblikoslovne oznake ali odvisnostna drevesa. Končne funkcije so *porazdelitev ciljnih oznak, ali se oblikoslovne oznake ujemajo na razdalji do dveh omenitev, porazdelitev oblikoslovnih oznak, ujemanje tipa omenitve, ali je omenitev samostalniški zaimsek, ali je omenitev zaimsek, razdalja med omenitvama v drevesu odvisnosti, oznake na poti do korena v drevesu odvisnosti, oznake na poti med omenitvama v drevesu odvisnosti, globina omenitve v drevesu odvisnosti in ujemanje starševe oznake v drevesu odvisnosti.*

Lokacija. Včasih je pomembno vedeti, kje se posamezna omenitev nahaja glede na dokument, poved ali ostale omenitve. Naš pristop razdaljo med omenitvami že implicitno upošteva zaradi predstavitve podatkov. Ostale atributske funkcije, ki jih uporabljamo, so *razdalja stavkov/besed med omenitvama, je prva / zadnja omenitev in sta omenitvi v isti povedi.*

Oblika omenitve. Omenitve so predstavljene kot posamezne besede ali fraze. Glede na podobne lastnosti pojavne oblike lahko sklepamo o njihovi koreferenčnosti, zato uporabljamo atributske funkcije *ali se omenitev začne z veliko začetnico, ali*

se zaporedni omenitvi začneta z veliko začetnico, ali se predpona / pripona / celoten niz / podniz level/desne omenitve na razdalji do pet ujemajo, ali se celoten niz/glava omenitev ujemata, Hearst-ova pravila ujemanja [126], ali je omenitev v dobesednem navedku, ali omenitev vsebuje glavo/celoten niz druge in razlika v dolžini nizov obeh omenitev.

Semantični pomen. Kategorija združuje atributske funkcije, ki zajemajo povezave med omenitvami z uporabo semantičnih podatkovnih virov, kot je WordNet [86], specializirani leksikoni, semantični sezname in ontologije. Atributske funkcije so *ali se oznake imenskih entitet omenitev ujemajo, ali omenitve določajo isti spol/število [127], ali sta si omenitvi apozitivni, ali je omenitev okrajšava za drugo (hevrstično), podobnost nizov omenitev, tip povezave v WordNet-u – nadpomenka / sopomenka / podpomenka, ali sta omenitvi v isti WordNet-ovi semantični množici, ali obe omenitvi naslavljata živo bitje ali ne [74], ali obe omenitvi govorita (upoštevanje konteksta).*

Kratek opis vsake od atributskih funkcij predstavimo v tabelah 3.2 in 3.3. Njihova natančna implementacija je dostopna v javnem repozitoriju programske kode [20] (v razredu *FeatureFunctionPackages*).

3.2.2 Zaporedja izpuščenih omenitev

Za odkrivanje koreferenčnosti predlagamo uporabo le izključno linearno-verižnih modelov CRF. Zaradi njihovega načina delovanja lahko odkrivajo le koreferenčnost med trenutno omenitvijo in prejšnjo omenitvijo. Precej zaporednih koreferenčnih omenitev se pojavlja na daljših razdaljah, kjer je lahko med njimi ena ali več ostalih omenitev. Zaradi tega v tem razdelku utemeljimo uporabnost našega predloga transformacije zaporedij.

Na sliki 3.1 prikazujemo porazdelitev razdalj med dvema zaporednima koreferenčnima omenitvama na podatkovni množici SemEval2010. Čeprav prikazujemo podatke le za eno množico, so reprezentativni in enakovredni rezultatom na ostalih množicah. Glede na porazdelitev ugotovimo, da je približno 10% neposredno zaporednih omenitev koreferenčnih, medtem ko je največ zaporednih koreferenčnih parov omenitev (tj. 12,5%) na razdalji ena – tj. se med njima nahaja neka druga omenitev. V primeru, da bi odkrili vse koreferenčnosti do razdalje 20, bi lahko identificirali 81% vseh parov.

Tabela 3.2

Opis atributskih funkcij - tabela 1/2. Vse funkcije so uporabljene v vseh modelih CRF v obeh faktorjih. Natančen opis (npr. katere vrednosti ali dodatne oznake omenitev so uporabljene) in implementacija je dostopna v javnem repozitoriju programske kode [20] (v razredu *FeatureFunctionPackages*).

Naziv	Opis
Porazdelitev ciljnih oznak	Verjetnost ciljne oznake.
Velika začetnica	Ali se omenitev začne z veliko začetnico.
Dvojna velika začetnica	Ali se obe zaporedni omenitvi začneta z veliko začetnico
Predpona	Predpona dolžine 2 ali 3 za omenitve na razdalji od -5 do 5 glede na trenutno omenitev.
Pripona	Pripona dolžine 2 ali 3 za omenitve na razdalji od -5 do 5 glede na trenutno omenitev.
Zaporedna vrednost	Vrednosti dveh zaporednih omenitev na razdalji -4 do 4 glede na trenutno omenitev.
Vrednost	Ali se predhodna omenitev ujema s trenutno.
Ujemanje spola	Ujemanje v spolu s predhodno omenitvijo.
Spol	Vrednost spola trenutne omenitve.
Apozitiv	Ali je omenitev apozitiv druge.
Drugo ime	Ali predhodna omenitev pomeni isto kot trenutna, je krajšava.
Je predpona	Ali je omenitev predpona druge.
Je pripona	Ali je omenitev pripona druge.
Vrednost podobnosti	Podobnost omenitev glede na metriko Jaro Winkler [128].
Je zaimsek	Ali je omenitev zaimsek.
Ista poved	Ali se omenitvi nahajata v isti povedi.
Hearst-ova pravila	Ali med omenitvami veljajo specifična pravila [126].

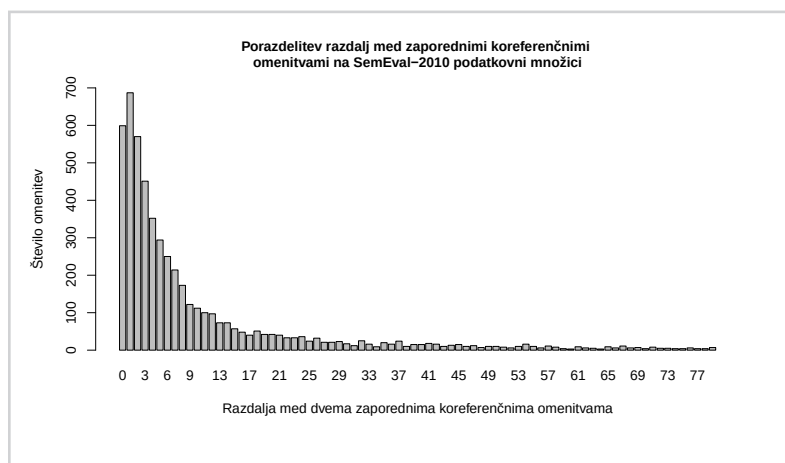
Podobno bi do razdalje 50 lahko odkrili 92% vseh parov omenitev. Z uporabo daljših ali vseh možnih razdalj pričakujemo, da bi se uspešnost odkrivanja koreferenčnosti zmanjšala, zato za izbiro največje vrednosti izberemo optimalno razdaljo (slika 3.6).

Glede na pregled besedil za vsak dokument ustvarimo zaporedje z nič izpuščenimi

Tabela 3.3

Opis atributskih funkcij - tabela 2/2. Vse funkcije so uporabljene v vseh modelih CRF v obeh faktorjih. Natančen opis (npr. katere vrednosti ali dodatne oznake omenitev so uporabljene) in implementacija je dostopna v javnem repozitoriju programske kode [20] (v razredu *FeatureFunctionPackages*).

Naziv	Opis
Razdalja povedi	Število povedi med omenitvama.
Ja v dobesednem navedku	Ali je omenitev med enojnimi / dvojnimi narekovanji.
Delno ujemanje	Ali je omenitev podniz druge.
Začetek	Ali je omenitev predpona druge.
Konec	Ali je omenitev pripona druge.
Število	Ali se omenitvi ujemata v številu (ednina, množina).
Tip	Tip omenitve (npr. ime, zaimek, nominalna vrednost).
Oziralni zaimek	Hevristična odločitev, če sta omenitvi povezani z oziralnim zaimkom.
WordNet	Semantična povezava med omenitvama (nadpomenka, podpomenka, sopomenka) [86].
Skupina WordNet	Ali sta omenitvi v isti semantični skupini.
Imenska entiteta	Tip imenske entitete omenitve.
Dolžina	Razlika v številu znakov omenitev.
Kazalni zaimek	Ali je omenitev eden izmed kazalnih zaimkov.
Oblikoslovna oznaka	Zaporedni oblikoslovni oznaki na razdalji -2 do 2 glede na trenutno oznako.
Pot odvisnosti	Oznake v drevesu odvisnosti na poti med omenitvama.
Globina odvisnosti	Globina omenitve v drevesu odvisnosti.
Starševska odvisnost	Oznaka starša v drevesu odvisnosti.
Povezava	Ali obstaja znana povezava med zaporednima omenitvama.
Govorec	Kdo izgovarja besedilo (za transkripte pogovorov).



Slika 3.1

Porazdelitev razdalj med zaporednimi koreferenčnimi omenitvami na podatkovni množici SemEval2010 [39]. Razdalja x med dvema zaporednima omenitvama pomeni, da med njima obstaja x ostalih omenitev.

omenitvami iz celotnega dokumenta. Nato iz tega za vsako razdaljo s tvorimo nova zaporedja z s izpuščenimi omenitvami. Zaporedje z s izpuščenimi omenitvami vsebuje vsako $(s + 1)$ -to omenitev iz dokumenta in za vsako razdaljo s naučimo ločen linearno-verižni model CRF.

3.2.3 Primer odkrivanja koreferenčnosti

V tem razdelku na vzorčnem primeru dokumenta “JANEZ JE POROČEN Z MOJCO. ON JE MEHANIČ V OBI-JU IN ONA PRAV TAKO DELA TAM. TO JE TRGOVINA MOJSTROVALCEV.” predstavimo predlagan postopek odkrivanja koreferenčnosti.

Naj zaporedje $\vec{x} = [x_1, x_2, x_3, \dots, x_n]$ označuje zaporedje vseh omenitev v dokumentu, ki so urejene glede na njihovo pojavitev. Torej za izbrani dokument pridobimo sledeče zaporedje $\vec{x}$:

$$\vec{x} = [\text{JANEZ}, \text{MOJCO}, \text{ON}, \text{OBI-JU}, \text{ONA}, \text{TAM}, \text{TO}, \text{TRGOVINA MOJSTROVALCEV}].$$

Omenitve so v splošnem samostalniške besede, torej bi lahko v fazi predprocesiranja kot omenitev razpoznali tudi besedo MEHANIČ, ki pa je zaradi lažjega prikaza delovanja predlaganega postopka nismo upoštevali kot omenitev. Glede na vhodni dokument želimo torej identificirati sklice na isto entiteto oziroma pravilno združiti omenitve v

gruče x_{JANEZ} , x_{MOJCA} in x_{OBI} :

$$x_{\text{JANEZ}} = \{\text{JANEZ, ON}\},$$

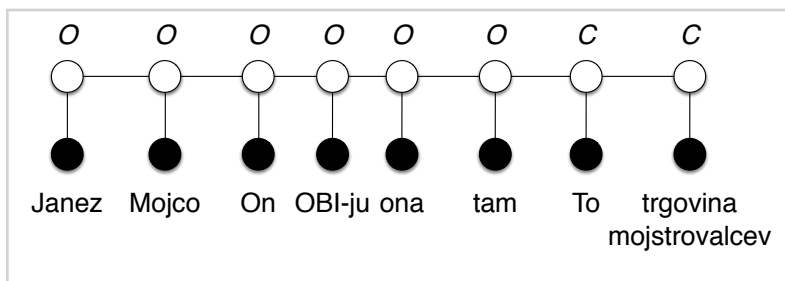
$$x_{\text{MOJCA}} = \{\text{MOJCO, ONA}\},$$

$$x_{\text{OBI}} = \{\text{OBI-JU, TAM, To, TRGOVINA MOJSTROVALCEV}\}.$$

V nekaterih primerih se lahko zgodi, da se ena omenitev prekriva z drugo omenitvijo. Tedaj jih obravnavamo kot ločeni omenitvi in jih v zaporedju omenitev uredimo leksikografsko glede na prvo besedo omenitve.

Slika 3.2

Zaporedje z nič izpuščenimi omenitvami. Osnovno zaporedje omenitev, ki vsebuje vse omenitve iz vhodnega dokumenta *Janez JE POROČEN z Mojco. On JE MEHANIČ V OBI-ju in ona PRAV TAKO DELA tam. To JE trgovina mojstrovalcev.. Če* je omenitev koreferenčna s predhodno omenitvijo, je označena s črko C in nasprotno z O.



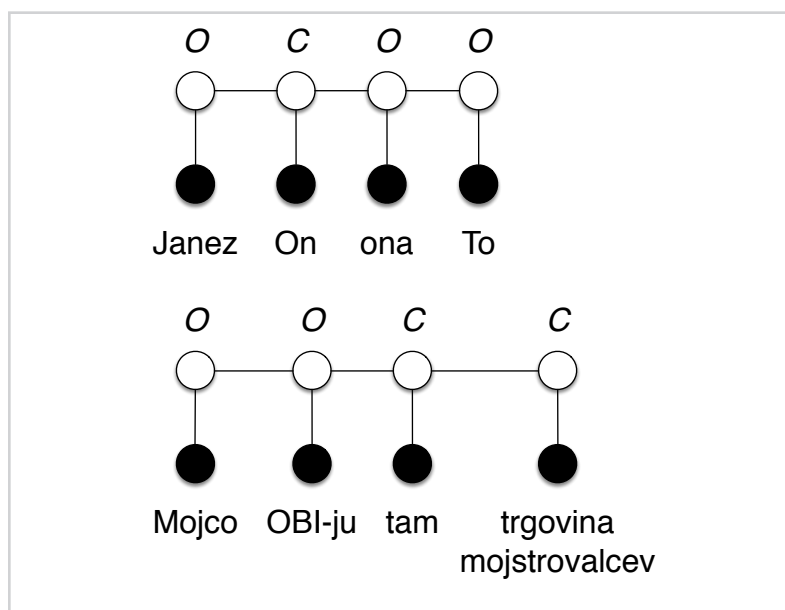
Pred začetkom se najprej odločimo, da bomo uporabili zaporedja z nič, eno in dvema izpuščenima omenitvama, kar je tudi obvezen parameter sistema. Na sliki 3.2 prikazujemo osnovno zaporedje $\bar{x}$, nad katerim lahko uporabimo katerikoli model prvega reda. To zaporedje imenujemo “zaporedje z nič izpuščenimi omenitvami”, ker vsebuje vse omenitve iz dokumenta in med dvema neposredno zaporednima omenitvama v besedilu ne nastopa nobena druga omenitev. Za pripravo učne podatkovne množice omenitve označimo z oznakami $\{O, C\}$, kjer C pomeni, da je trenutna omenitev koreferenčna s prejšnjo in O nasprotno. Nad takšnimi zaporedji naučimo linearno-verižni model CRF in nato lahko napovedujemo koreferenčnost med neposredno sosednimi omenitvami. Za izbran primer bi lahko na takšen način odkrili kvečjemu tri koreferenčne omenitve $\{\text{TAM, To, TRGOVINA MOJSTROVALCEV}\}$.

Na opisan način ne moremo razpoznati koreferenčnih omenitev na daljših razdaljah (tj. OBI-JU in TAM), zato predlagamo transformacije osnovnega zaporedja, ki so ustrezno označene z oznakami C in O. Za vsak tip transformacije naučimo ločen linearno-verižni model CRF in ga uporabimo za označevanje koreferenčnosti omenitev na izbrani razdalji. To nam omogoči, da vsak tip modela razpozna koreferenčnosti med

omenitvami na določeni razdalji. Rezultati vseh različnih tipov modelov nato tvorijo pare razpoznanih koreferenčnosti, ki jih nazadnje sestavimo v entitete.

Torej, iz osnovnega zaporedja – zaporedja z nič izpuščenimi omenitvami (slika 3.3) – ustvarimo še ostala zaporedja z izpuščenimi simboli. Zaporedja z eno izpuščeno omenitvijo (slika 3.3) vsebujejo vsako drugo omenitev iz osnovnega zaporedja. Glede na vhodne povedi nam omogočijo, da razširimo množico rezultatov z dvema novima paroma {JANEZ, ON} in {OBI-JU, TAM}. Analogno postopek ponovimo za zaporedja z dvema izpuščenima omenitvama (slika 3.4) in pridobimo še zadnja manjkajoča para {OBI-JU, To} in {MOJCO, ONA}.

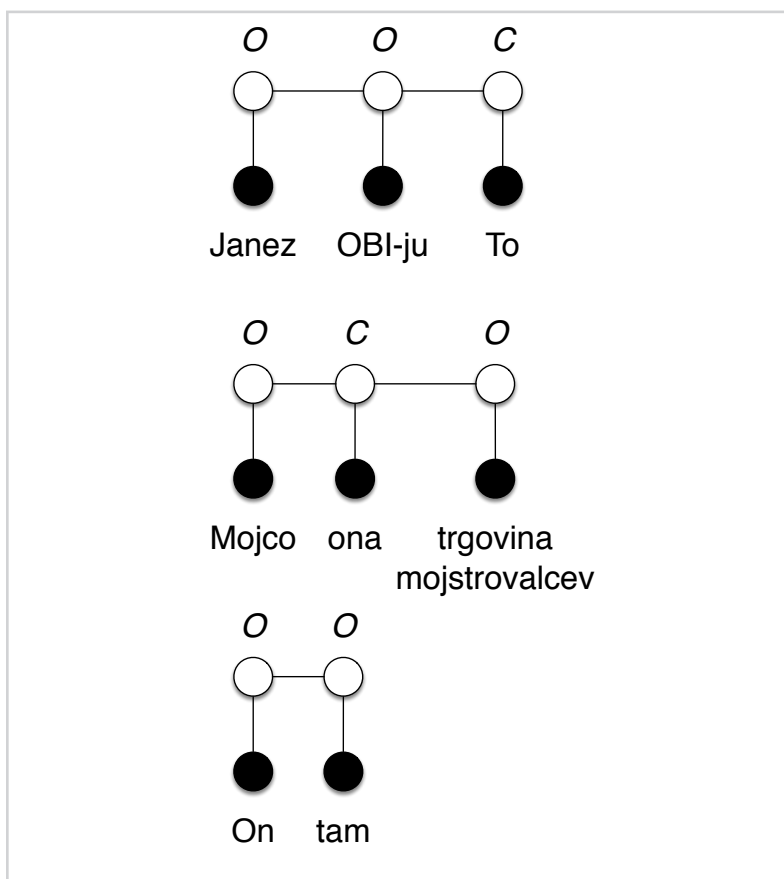
V zadnjem koraku izvedemo gručenje omenitev glede na razpoznanne koreferenčne omenitve iz vseh zaporedij in kot rezultat vrnemo ciljne skupine omenitev x_{JANEZ} , x_{MOJCA} in x_{OBI} , kjer vsaka skupina predstavlja eno entiteto.



Slika 3.3

Zaporedji z eno izpuščeno omenitvijo. Zaporedje vsebuje vsako drugo omenitev iz vhodnega dokumenta *Janez JE POROČEN z Mojco. On JE MEHANIČ v OBI-ju in ona PRAV TAKO DELA tam. To JE trgovina mojestrovalcev.. Če* je omenitev koreferenčna s predhodno omenitvijo, je označena s črko C in nasprotno z O.

Glede na zgornje primere lahko opazimo, da transformacija iz osnovnega zaporedja omenitev v zaporedja z izpuščenimi omenitvami vrne več kot eno zaporedje na dokument. Na primer, za nič izpuščenih omenitev nam vrne le eno zaporedje (le to vsebuje



Slika 3.4

Zaporedja z dvema izpuščenima omenitvama. Zaporedje vsebuje vsako tretjo omenitev iz vhodnega dokumenta *Janez JE POROČEN z Mojco. On JE MEHANIK v OBI-ju in ona PRAV TAKO DELA tam. To JE trgovina mojstrovalcev.* Če je omenitev koreferenčna s predhodno omenitvijo, je označena s črko C in nasprotno z O.

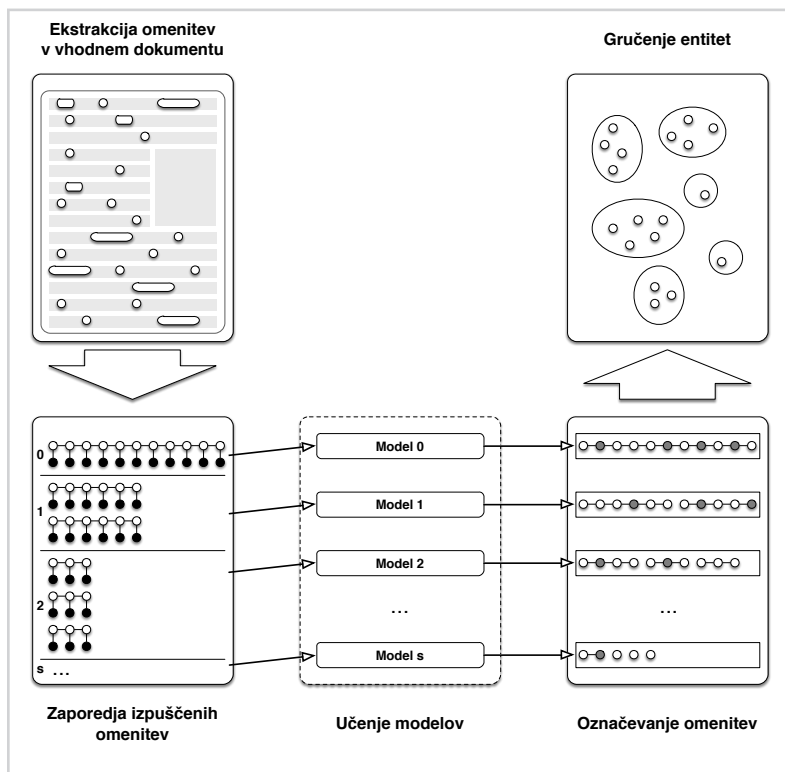
vse omenitve iz dokumenta), za eno izpuščeno omenitev nam vrne dve zaporedji (vsebujeta vsako drugo omenitev iz dokumenta), za dve izpuščeni omenitvi nam vrne tri zaporedja, ipd. Torej vsaka transformacija v zaporedja z d izpuščenimi omenitvami vrne $d + 1$ zaporedij dolžine $\lceil \frac{n}{d} \rceil$ omenitev, kjer n predstavlja število vseh omenitev v dokumentu.

3.2.4 Sistem SkipCor

Sistem SkipCor na vходу prejme dokumente z besedili in kot izhod vrne gruče koreferenčnih omenitev, pri čemer vsaka od njih predstavlja entiteto. Pri napovedovanju koreferenčnosti omenitev algoritem najprej identificira omenitve v vhodnem besedilu in jih pretvori v zaporedja izpuščenih omenitev. Nato za vsak tip zaporedja uporabi ustrezen linearno-verižni model CRF in vrne označena zaporedja, ki se uporabijo v postopku gručenja. Kot končni rezultat vrne seznam entitet, ki so predstavljene kot množice omenitev. Postopek učenja je podoben postopku napovedovanja, le da modele CRF pred njihovo prvo uporabo najprej naučimo (črtkan kvadrat na sliki 3.5). Vsakega od naučenih linearno-verižnih modelov CRF nato uporabimo za napovedovanje koreferenčnosti za določeno število izpuščenih omenitev. Na sliki 3.5 prikazujemo visokonivojsko predstavitev toka podatkov v sistemu SkipCor, čemur sledita algoritma za učenje (algoritem (3.4)) in napovedovanje (algoritem (3.5)) koreferenčnosti.

V fazi učenja, ki je predstavljena z algoritmom 3.4, izgradimo model za odkrivanje koreferenčnosti z izpuščenimi omenitvami. Parametri poleg učne podatkovne množice sestojijo še iz seznama atributskih funkcij in seznama razdalj za izpuščanje omenitev. Predlagan algoritem najprej prebere učno podatkovno množico in jo obogati z dodatnimi oznakami, kot so oblikoslovne oznake, leme ali drevesa odvisnosti. Nato iz vsakega podanega dokumenta ustvari zaporedje z nič izpuščenimi omenitvami, ki ohranijo povezave do prvotnega besedila, da lahko atributske funkcije uporabijo tudi njihov kontekst v izvirnem dokumentu. Temu sledi vzporedno učenje modelov CRF za vsako podano razdaljo izpuščenih omenitev. Pred učenjem vsakega izmed modelov pretvorimo osnovna zaporedja v zaporedja z s_i izpuščenimi omenitvami in nad novimi zaporedji s pomočjo atributskih funkcij pridobimo značilke za vsako omenitev. Nato neodvisno za vsak tip zaporedja naučimo linearno-verižni model CRF. Kot rezultat vrnemo model SkipCor, ki sestoji iz seznama naučenih modelov CRF vključno z njihovimi razdaljami izpuščenih omenitev in seznama uporabljenih atributskih funkcij.

Odkrivanje koreferenčnosti na novih besedilih izvajamo z algoritmom 3.5. Algoritem kot vhod sprejme dokument z besedilom in model SkipCor (rezultat algoritma 3.4). Podobno kot v fazi učenja tudi tu predprocesiramo besedilo in pridobimo osnovno zaporedje z nič izpuščenimi omenitvami. V primeru, da omenitve niso znane že vnaprej, jih pridobimo z uporabo pravil, ki temeljijo na samostalniških besedah [70]. Glede na to, da na vходу sprejmemo samo en dokument, imamo samo eno osnovno



Slika 3.5

Visokonivojska predstavitev toka podatkov za odkrivanje koreferenčnosti z izpuščenimi omenitvami. V vhodnem dokumentu najprej identificiramo omenitve in jih pretvorimo v zaporedje (brez izpuščenih) omenitev. Nato glede na vhodne parametre sistema to zaporedje pretvorimo v ustrezna zaporedja z izpuščenimi omenitvami in za vsak tip zaporedja naučimo ustrezne linearno-verižne modele CRF (črtkano), ki jih kasneje uporabimo za napovedovanje. Po napovedovanju izvedemo gručenje omenitev in kot rezultat vrnemo seznam gruč, ki predstavljajo entitete.

zaporedje omenitev. Nato algoritem v vzporedni zanki ustvari primerna zaporedja za vsak model CRF, pridobi značilke in z ustreznim modelom CRF označi zaporedja s_i . Vse pare koreferenčnih omenitev shranimo v množico, ki jo obdelamo z razvrščanjem in izdelamo gruče omenitev. Vsaka od gruč predstavlja svojo entiteto, kar je tudi rezultat odkrivanja koreferenčnosti.

Razvrščanje omenitev izvajamo po slečem postopku. Vse omenitve iz osnovnega zaporedja na začetku predstavljajo svojo gručo (angl. singleton). Nato pregledujemo odkrite koreferenčnosti iz napovedanih oznak zaporedij in združujemo gruče med seboj. Rezultat razvrščanja je hkrati tudi rezultat odkrivanja koreferenčnosti, saj vsaka gruča predstavlja svojo entiteto.

Tabela 3.4

Algoritem za učenje modela SkipCor.

Vhod: učni dokumenti D , atributske funkcije $f_l \in F$ in razdalje $s_i \in S$
Izhod: model SkipCor ($skipMentionCRF, S, F$)

```

1:  $sentences \leftarrow \text{uvoziUcnePodatke}(D)$ 
2:  $sentences \leftarrow \text{predprocesiraj}(sentences)$ 
3:  $mentionSequences \leftarrow \text{preberiOmenitve}(sentences)$ 
4:  $skipMentionCRF \leftarrow []$  //empty list
5: parallel for each  $s_i \in S$ :
6:    $skipMentionSequences \leftarrow \text{preoblikuj}(mentionSequences, s_i)$ 
7:    $\text{inicializirajAtributskeFunkcije}(skipMentionSequences, F)$ 
8:    $skipMentionCRF_i \leftarrow \text{LCRFTrain}(skipMentionSequences)$ 
9: return ( $skipMentionCRF, S, F$ )

```

Tabela 3.5

Algoritem za napovedovanje koreferenčnosti z modelom SkipCor.

Vhod: dokument D in model SkipCor ($skipMentionCRF, S, F$)
Izhod: gruče koreferenčnih omenitev $mentionClusters$ – entitete

```

1:  $sentences \leftarrow \text{predprocesiraj}(D)$ 
2:  $mentionSequence \leftarrow \text{razpoznavajOmenitve}(sentences)$ 
3:  $coreferentMentions \leftarrow \emptyset$ 
4: parallel for each  $s_i \in S$ :
5:    $skipMentionSequences \leftarrow \text{preoblikuj}(mentionSequence, s_i)$ 
6:    $\text{inicializirajAtributskeFunkcije}(skipMentionSequences, F)$ 
7:    $coreferentMentions \text{ addAll}$ 
8:    $\text{LCRFLabel}(skipMentionCRF_i, skipMentionSequences)$ 
9:  $mentionClusters \leftarrow \text{izvediGrucenje}(coreferentMentions)$ 
10: return  $mentionClusters$ 

```

Časovna kompleksnost obeh algoritmov je v največji meri odvisna od učenja ali označevanja z modeli CRF (tj. funkciji $LCRFTrain$ in $LCRFLabel$), saj so ostale me-

tode izvedljive v linearnem času. Kljub temu se lahko zgodi, da označevanja v fazi predprocesiranja ali uporaba atributskih funkcij zahtevajo nadlinearno časovno zahtevnost. Zaradi paralelnega izvajanja želimo torej najti najdlje trajajočo zanko. Naj velja, da se učenje ali napovedovanje z modeli CRF izvaja s časovno kompleksnostjo $O(EL^Q)$ [99], kjer je E število povezav, L število oznak in O velikost največjega faktorja (tj. klike v grafovski predstavitvi modela). Število oznak L je v našem primeru dve, saj za vsako omenitev ugotavljamo le, ali je koreferenčna s prejšnjo ali ne. Število povezav E je odvisno od vhodnega zaporedja v algoritmu. Naj ima vhodno zaporedje n omenitev, kar pomeni, da bo osnovno zaporedje vsebovalo $2n - 1 = O(n)$ povezav. Vsako izpeljano zaporedje z d izpuščenimi omenitvami pa bo zaradi razpada v več podzaporedij prav tako vsebovalo $d(\lceil \frac{2n}{d} \rceil - 1) = 2n - d = O(n)$ povezav. Torej z uporabo paralelizacije potrebujejo modeli CRF $O(2^2n) = O(n)$ časa. Poleg ostalih linearnih funkcij je pomembno, da upoštevamo inicializacijo značilk z atributskimi funkcijami, kar potem znaša $O(mn)$, kjer je m število atributskih funkcij.

3.3 Analiza rezultatov

V poglavju analize rezultatov najprej predstavimo metrike za ocenjevanje koreferenčnosti, razložimo predpostavke, ki veljajo za celotno analizo, in razložimo delovanje osnovnih izvedenk predlaganega pristopa SkipCor. Nato opišemo podatkovne množice z osnovnimi statistikami, načine njihovega ročnega označevanja in dodatne lastnosti, ki jih uporabljamo pri učenju modelov. Zatem evalviramo predlagane pristope na vseh podatkovnih množicah in jih primerjamo med seboj ter z drugimi znanimi pristopi. Nazadnje raziščemo še pričakovan upad natančnosti odkrivanja koreferenčnosti na realnih podatkih.

3.3.1 Ocene uspešnosti

V času razvoja metod za odkrivanje koreferenčnosti je bilo predlaganih kar nekaj ocen uspešnosti, vendar se raziskovalci še vedno niso jasno odločili, katero uporabljati. Prve ocene [129, 130] so temeljile na grafovskih značilnostih rezultatov ali na primerjavi parov omenitev. Kljub temu, da je bilo število odkritih koreferenčnosti majhno, so bile njihove ocene visoke in so slabo ločevale dobre sisteme od slabih. V naši analizi uporabljamo torej več novejših metrik, da se lahko preko njih primerjamo z že obstoječimi rezultati. Sledeče metrike so izpeljanke ocene F_1 z različnimi pristopi računanja natančnosti in priklica:

MUC. Glavni namen razvoja ocene MUC [131] je bil doseči boljšo razlago rezultatov pri odkrivanju koreferenčnosti. Ime je dobila po istoimenski konferenci – MUC (angl. Message Understanding Conference). Ocena temelji na povezavah (angl. link-based metric) med omenitvami in je največkrat uporabljena. Natančnost ocene izračunamo tako, da preštejemo, koliko bi bilo najmanjše število povezav, ki bi jih morali dodati k rezultatu, da bi bile entitete pravilno razpoznane. Nasprotno, priklic ocene meri, koliko povezav bi morali odstraniti, da ne bi obstajala nobena povezava, ki povezuje omenitvi dveh različnih entitet. Ocena MUC torej bolje oceni sisteme, ki entitetam priredijo veliko omenitev, medtem ko popolnoma ignorira razpoznavanje entitet, ki sestojijo le iz ene omenitve (angl. singleton entity). Na primer, da sistem v dokumentu razpozna le eno entiteto (tj. vse omenitve so povezane med seboj), bo dosegel 100% priklic in precej visoko natančnost.

BCubed. Zaradi slabosti MUC ocene so raziskovalci predlagali oceno BCubed [132], ki se osredotoča na posamezne omenitve in meri preseke med napovedanimi gručami omenitev in pravimi gručami. Naj bo k prava entiteta in r razpoznana entiteta, ki sestoji iz omenitev m . Priklic omenitve m izračunamo kot $\frac{|k \cap r|}{|k|}$ in natančnost kot $\frac{|k \cap r|}{|r|}$. Prednost te ocene je, da upošteva tudi entitete z le eno omenitvijo in daje višjo težo združevanju ali združevanju entitetam z več omenitvami.

CEAF. Nazadnje je bila predlagana ocena CEAF [133] (angl. Constrained Entity-Alignment F-Measure) s katero so raziskovalci želeli doseči boljšo razlago rezultatov oziroma razlik med sistemi. Ocena pomeni delež pravilno razpoznanih entitet. Za namene naše evalvacije uporabljamo oceno, ki temelji na ocenjevanju entitet (obstaja tudi različica, ki temelji na omenitvah) in želi razpoznani entiteti najti čimboljši par v množici pravih entitet. Pri oceni izračunamo priklic kot $\frac{\text{total similarity}}{|k|}$ in natančnost kot $\frac{\text{total similarity}}{|r|}$, kjer za problem največjega dvojnega ujemanja (angl. maximum bipartite matching) uporabljamo algoritem Kuhn-Munkres.

Za namene evalvacije v tem delu kot pravilno upoštevamo le natančno ujemanje omenitev [40] z dodatnimi spremembami, ki sta jih predlagala Cai in Strube [66].

3.3.2 Opis podatkovnih množic

Večina sistemov za odkrivanje koreferenčnosti je bila predstavljena in evalvirana v okviru konferenc MUC (Message Understanding Conference) [134], ACE (Automatic Content Extraction) [38], SemEval2010 (Semantic Evaluation) [39] in nazadnje na konferencah CoNLL-2011 and CoNLL-2012 (Conference on Computational Language Learning) [40]. V tabeli 3.6 prikazujemo nekaj osnovnih statistik angleških podatkovnih množic na katerih evalviramo predlagan pristop. Podobno kot predhodni raziskovalci se tudi mi osredotočamo na novičarska besedila in uporabljamo naslednje podatkovne množice: (1) ACE 2004 poleg časopisnih in televizijskih novic vsebuje še zapise pogovorov in prepise novic iz kitajščine in arabščine. Vsebuje oznake za vse glavne naloge ekstrakcije informacij, zato na področju predstavlja *de facto* standard med podatkovnimi množicami. (2) SemEval 2010 je bil zasnovan z namenom evalvacije sistemov za odkrivanje koreferenčnosti v šestih jezikih. Angleški del vsebuje časopisne in televizijske novice iz časnika The Wall Street Journal in zbirke TDT-4. (3) Podatkovna množica CoNLL 2012 je najbolj obširna in skuša pokriti večji del entitet, med katere vključi tudi dogodke. Množica vsebuje časopisne in televizijske novice, revijalne članke, spletne podatke, pogovore in angleški prevod Nove Zaveze.

Tabela 3.6

Opis podatkovnih množic ACE2004 [38], SemEval2010 [39] in CoNLL2012 [40]. Kratice ALL (tj. celoten), NW (tj. časopisne novice), BN (tj. televizijske novice) definirajo posamezne podmnožice, ki so prav tako razdeljene na učni in testni del.

Množica	# dokumentov	# povedi	# členov	# omenitev	# entitet
ACE2004-ALL	450	7,518	191,387	29,724	12,439
ACE2004-NW	127	2,865	74,987	11,188	4,701
ACE2004-BN	220	3,782	71,602	11,323	4,918
SemEval2010-Train	229	3,648	78,831	21,550	16,082
SemEval2010-Test	85	1,141	24,121	6,692	4,839
CoNLL2012-ALL-Train	1,914	75,185	1,299,310	154,760	33,113
CoNLL2012-ALL-Test	221	9,479	169,579	19,677	4,217
CoNLL2012-NW-Train	734	15,288	387,082	34,470	9,404
CoNLL2012-NW-Test	88	1,898	49,235	4,361	1,168
CoNLL2012-BN-Train	748	9,723	180,300	22,262	6,433
CoNLL2012-BN-Test	93	1,252	23,209	2,936	790

Predlagan sistem SkipCor uporabimo za odkrivanje koreferenčnosti med vsemi tipi omenitev (tj. imenskimi, nominalnimi in zaimki). Zaradi različnosti podatkov in drugačnih pravil za označevanje omenitev ne moremo neposredno primerjati rezultatov preko različnih podatkovnih množic. Na primer obe množici, ACE in CoNLL, vsebujeta vse tri tipe omenitev, vendar CoNLL označuje več splošnih entitet. CoNLL vsebuje tudi natančne meje posameznih omenitev, ki se ne ujemajo s podnizi iz drevesnice odvisnosti, torej je pričakovano, da bomo na CoNLL množici dosegali slabše rezultate v primerjavi z ACE. SemEval definira le nominalne omenitve. Poleg tega pa so bile entitete z eno samo omenitvijo označene hevristično. V drugem delu analize kljub različnostim izvedemo učenje na eni podatkovni množici in testiranje na drugi, saj s tem pokažemo, kakšna je pričakovana uspešnost metod na realnih podatkih. Tako kot imamo ljudje različne predstave o koreferenčnosti, tako so različne tudi podatkovne množice.

V fazi predprocesiranja uporabimo Apache OpenNLP [33] za razpoznavanje povedi, oblikoslovno označevanje in razčlenjevanje odvisnosti. Za učenje in označevanje z linearno-verižnimi modeli CRF uporabljamo knjižnico CRFSuite [135] z optimizacijsko metodo Broyden-Fletcher-Goldfarb-Shanno, ki ji podamo atributske funkcije, ki so bile prožene najmanj petkrat. Celotna evalvacija je ponovljiva in dostopna v javnem repozitoriju programske kode [20].

3.3.3 Preizkus na označenih podatkovnih množicah

Sistem SkipCor poleg seznama atributskih funkcij kot vhod sprejme tudi seznam razdalj med izpuščenimi omenitvami. Z uporabo preveč razdalj lahko uspešnost sistema ostaja na isti točki ali se celo poslabša. Na sliki 3.6 prikazujemo rezultate v odvisnosti od razdalj med izpuščenimi omenitvami. Opazimo lahko, da z upoštevanjem razdalj, daljših od 40, uspešnost ne narašča ali se značilno spreminja, saj vrednost priklica narašča za približno toliko, kolikor vrednost natančnosti pada. Zaradi tega ocene F_1 ostajajo približno enake. Rezultati, ki jih predstavimo v nadaljevanju, so pridobljeni z upoštevanjem vseh razdalj med nič in 25 (pokončna črta na sliki 3.6). V vseh primerih analize uporabimo že odkrite omenitve in raziščemo le odkrivanje koreferenčnosti med njimi.

Sistem SkipCor primerjamo z dvema osnovnima izvedenkama enakega sistema – SkipCorZero in SkipCorPair – z enakim naborom atributskih funkcij. Edina razlika med njimi je uporaba različnih zaporedij omenitev. SkipCorZero odkriva koreferenč-

nost le nad osnovnimi zaporedij izpuščenih omenitev, medtem ko SkipCorPair preverja le zaporedja dolžine dve, v katerih zaradi velikega števila parov nastopajo le pari omenitev na razdalji največ deset omenitev.

V tabeli 3.7 predstavimo rezultate na podatkovni množici ACE2004. Ko analiziramo množico časopisnih in televizijskih prispevkov, razdelimo množico za učenje in testiranje v razmerju 70:30. Za celotno množico pa uporabimo 336 dokumentov za učenje in ostale za testiranje [112]. SkipCorZero in SkipCorPair sta dosegla dokaj dobro ali najboljšo natančnost, vendar zelo nizek priklic. V splošnem se SkipCorPair izkaže bolje kot SkipCorZero, medtem ko predlagan sistem SkipCor preseže oba. V primerjavi z ostalimi pristopi SkipCor doseže malenkost boljšo oceno BCubed, vendar nižjo oceno MUC. Ker so rezultati tako zelo tesni in nasprotujoči si na dveh merah, je težko zaključiti, kateri sistem je v splošnem boljši. Na časopisnih in televizijskih prispevkih dosežemo boljši oceni MUC in BCubed. Čeprav dosežemo nižje natančnosti, je naša razlika med natančnostjo in priklicem veliko manjša kot pri ostalih sistemih. Glede na rezultate trdimo, da smo odkrili več entitet z nekaj več napakami, vendar skupni rezultati kažejo izboljšanje v primerjavi z drugimi sistemi. Rezultati na celotni množici (ACE2004-ALL) pa so primerljivi z obstoječimi najboljšimi sistemi.

V tabeli 3.8 prikazujemo rezultate na podatkovni množici CoNLL2012. Množica je že razdeljena v dele za učenje, nastavljanje parametrov (pri učenju tega dela nismo uporabili) in testiranje. Iz množice uporabimo meje, kjer se nahajajo omenitve in dodatne značilke, ki so prisotne poleg besedila, da so naši rezultati primerljivi s sistemi v načinu *Gold Mention Boundaries*. Fernandes in sod. [125] so predlagali zmagovalni sistem in hkrati so edini, ki so objavili rezultate še na posameznih delih podatkovne množice. Podobno kot pri rezultatih na množici ACE2004, tudi tu sistem SkipCor preseže osnovna modela SkipCorPair in SkipCorZero, razen na delu televizijskih novic, kjer ga SkipCorPair preseže z najboljšo natančnostjo in zelo dobrim priklicem. SkipCor v primerjavi z ostalimi sistemi malenkostno izboljša rezultate, še posebej z oceno MUC, kar pomeni, da ustvari čistejše gruče omenitev. Za namene tekmovanja v času konference je rezultate objavilo devet ekip, vendar tu prikazujemo le rezultate najboljših šest. Glede na oceno MUC naš sistem značilno izboljša rezultate, kjer precej izboljšamo natančnost in zadržimo enak nivo priklica. Z oceno BCubed je podobno, le pri oceni CEAF se odrežemo malenkost slabše. Sodelujoči sistemi na konferenci so bili ocenjeni z oceno CoNLL2012, ki je povprečje ocen MUC, BCubed in CEAF. Zmagovalni sistem [125] je dosegel vrednost 63.1, medtem ko bi se naš sistem kot drugi uvrstil z

Tabela 3.7

Rezultati predlaganega sistema SkipCor, njegovih osnovnih izvedenk in ostalih pristopov na podatkovni množici ACE2004, kjer ALL pomeni celotno množico, NW časopisne prispevke in BN televizijske prispevke. Uporabljene oceni sta MUC [131] in BCubed [132]. Vrednosti P, R in F predstavljajo natančnost, priklic in oceno F.

Sistem	MUC			BCubed		
	P	R	F	P	R	F
ACE2004-NW						
SkipCor	78.6	68.8	73.4	75.7	78.6	77.1
SkipCorZero	78.5	22.6	35.1	96.3	51.9	67.4
SkipCorPair	78.2	49.0	60.3	85.3	61.7	71.6
Finkel in sod.[136]	78.7	58.5	67.1	86.8	65.2	74.5
Soon in sod.[71] ¹	85.3	37.8	52.4	94.1	56.9	70.9
Haghighi in sod.[113]	77.0	75.9	76.5	79.4	74.5	76.9
Stoyanov in sod.[117]	-	-	62.1	-	-	75.5
ACE2004-BN						
SkipCor	76.3	71.3	73.7	76.2	81.5	78.8
SkipCorZero	79.3	28.3	41.7	95.9	57.3	71.8
SkipCorPair	80.9	59.4	68.5	86.3	70.7	77.7
Finkel in sod.[136]	87.8	46.8	61.1	93.5	59.9	73.1
Soon in sod.[71] ¹	90.0	43.2	58.3	95.6	58.4	72.5
ACE2004-ALL						
SkipCor	79.5	70.9	75.0	76.3	81.1	78.6
SkipCorZero	81.3	28.9	42.6	95.6	55.4	70.2
SkipCorPair	80.5	57.1	66.8	84.8	68.9	76.0
Cullota in sod.[112]	-	-	-	86.7	73.2	79.3
Bengston in sod.[72]	-	-	-	88.3	74.5	80.8
Haghighi in sod.[113] ²	74.8	77.7	76.2	79.6	78.5	79.0

¹ Rezultate sta rekonstruirala Finkel in Manning [133].

² Ocena MUC se ne ujema z zapisano natančnostjo in priklicem, zato smo jo ustrezno preračunali.

oceno 61.3. Naslednji bi mu nato sledil z oceno 60.7 in vsi ostali do najnižje ocene 43.0.

Tabela 3.8

Rezultati predlaganega sistema SkipCor, njegovih osnovnih izvedenk in ostalih pristopov na podatkovni množici CoNLL2012, kjer ALL pomeni celotno množico, NW časopisne prispevke in BN televizijske prispevke. Uporabljene ocene so MUC [131], BCubed [132] in CEAF [133]. Vrednosti P, R in F predstavljajo natančnost, priklic in oceno F.

Sistem	MUC			BCubed			CEAF		
	P	R	F	P	R	F	P	R	F
	CoNLL2012-NW								
SkipCor	76.1	61.5	68.0	69.6	70.9	70.2	39.9	59.8	47.8
SkipCorZero	76.7	16.8	27.5	95.5	36.7	53.0	35.1	35.3	35.2
SkipCorPair	81.3	45.9	58.7	86.4	56.7	68.5	42.0	54.7	47.5
Fernandes in sod.[125]	-	-	61.6	-	-	70.0	-	-	46.6
CoNLL2012-BN									
SkipCor	76.2	62.0	68.4	69.8	71.3	70.5	34.2	59.2	43.3
SkipCorZero	76.8	19.4	31.0	95.9	45.8	62.0	37.4	37.6	37.5
SkipCorPair	81.9	44.7	57.8	88.6	54.6	67.6	39.1	56.3	46.1
Fernandes in sod.[125]	-	-	65.6	-	-	70.0	-	-	49.1
CoNLL2012-ALL									
SkipCor	84.9	63.6	72.7	74.6	65.6	69.8	32.9	57.2	41.7
SkipCorZero	81.8	21.7	34.3	95.7	39.7	56.1	32.1	32.7	32.4
SkipCorPair	85.9	50.7	63.8	86.5	53.4	66.0	30.8	53.8	39.2
Fernandes in sod.[125]	77.5	64.9	70.7	79.0	64.3	70.9	41.7	56.5	48.0
Björkelund in sod.[137]	71.6	63.4	67.3	76.6	64.0	69.7	41.4	50.0	45.3
Chen in sod.[138]	66.8	63.3	65.0	73.6	65.4	69.2	44.9	48.8	46.8
Stamborg in sod.[139]	58.8	66.2	62.3	65.0	71.2	68.0	45.8	38.5	41.8
Zhekova in sod.[140]	54.7	55.0	54.8	55.6	61.9	58.6	34.7	34.4	34.5
Li in sod.[141]	33.7	44.2	38.2	53.9	66.4	59.5	36.5	27.5	31.4

Tabela 3.9 prikazuje rezultate na podatkovni množici SemEval2010, ki je že ustrezno ločena v del za učenje in testiranje. Glede na predpostavke se primerjamo z ostalimi sistemi v rezultatih tipa *Gold-standard Closed*, kjer lahko uporabljamo že identificirane omenitve in le vrednosti, ki nastopajo v podatkovni množici. Sistem SkipCor se izkaže bolje kot SkipCorZero na vseh možnostih, medtem ko v primerjavi s sistemom SkipCorPair doseže boljše rezultate z ocenami CEAF in BCubed. V primerjavi z ostalimi sistemi SkipCor doseže boljše ocene BCubed in CEAF, vendar slabšo oceno MUC.

Kot zanimivost, sistem RelaxCor se je na tekmovanju izkazal kot najboljši, vendar ga SkipCor prehiti v vseh treh ocenah. Če se primerjamo z oceno MUC, pa dosežemo drugo mesto, saj nas sistem SUCRE prehiti.

Tabela 3.9

Rezultati predlaganega sistema SkipCor, njegovih osnovnih izvedenk in ostalih pristopov na podatkovni množici SemEval2010. Uporabljene ocene so MUC [131], BCubed [132] in CEAF [133]. Vrednosti P, R in F predstavljajo natančnost, priklic in oceno F.

Sistem	MUC			BCubed			CEAF		
	P	R	F	P	R	F	P	R	F
	SemEval2010								
SkipCor	68.8	30.1	41.8	94.8	80.8	87.3	74.0	78.5	76.2
SkipCorZero	67.0	3.6	6.8	99.6	75.1	85.7	73.0	73.1	73.1
SkipCorPair	76.7	35.6	48.7	97.1	79.0	87.1	72.7	79.4	75.9
RelaxCor [142]	72.4	21.9	33.7	97.0	74.8	84.5	75.6	75.6	75.6
SUCRE [143]	54.9	68.1	60.8	78.5	86.7	82.4	74.3	74.3	74.3
TANL-1 [124]	24.4	23.7	24.0	72.1	74.6	73.4	61.4	75.0	67.6
UBIU [144]	25.5	17.2	20.5	83.5	67.8	74.8	68.2	63.4	65.7

Sistem UBIU [144], ki se je udeležil tekmovanja SemEval2010, je z nekaj izboljšavami tekmoval tudi v okviru CoNLL2012 [140]. Sistem SkipCor se izkaže značilno boljši kot UBIU v vseh treh ocenah. V primerjavi z delovanjem našega sistema UBIU pregleduje pare omenitev in uporablja tehniko strojnega učenja s pomnjenjem (angl. memory-based machine learning).

Glede na prikazane rezultate dosega sistem SkipCor boljše ocene kot oba osnovna sistema SkipCorZero in SkipCorPair. SkipCorZero večinoma doseže dobro natančnost, vendar zelo nizek priklic, kar je pričakovano, saj lahko odkriva le koreferenčnosti med neposredno zaporednimi omenitvami. SkipCor zaradi tega uporablja dodatna zaporedja z izpuščenimi omenitvami. SkipCorPair se uvršča nekje med njima in deluje zelo podobno kot ostali sistemi, ki preverjajo koreferenčnosti med pari omenitev. Zaradi velike količine parov moramo ustvariti mnogo zaporedij dolžine dve, zato je tudi čas izvajanja s SkipCorPair daljši, še posebej, če bi uporabili pare še na daljših razdaljah.

V splošnem dosega sistem SkipCor izboljšave ali primerljive rezultate na vseh podatkovnih množicah. Statističnih testov nismo izvedli, ker tudi v ostalih člankih poročajo le o doseženih ocenah F_1 , zato nimamo dovolj podatkov in tudi njihovi sistemi niso

javno na voljo. Čeprav obstajajo nekateri sistemi, ki delujejo le s pomočjo ročno definiranih pravil, niso dobro prilagodljivi za različne domene in tudi to je bil eden od razlogov, da predlagamo enostaven postopek z uporabo metode strojnega učenja, ki dosega zelo dobre priklice, a malo slabše natančnosti. Ostali najboljši sistemi uporabljajo hibridne pristope, v katerih kombinirajo pravila in metode strojnega učenja, a vsi od njih kljub temu izkoriščajo bogat nabor atributskih funkcij. Na tekmovanju ACE2004 so Haghighi in sod. [113] uporabili popolnoma determinističen pristop, ki temelji na sintaktičnih in semantičnih omejitvah. Nasprotno sta se na primer Bengston in Roth [72] omejila na iskanje čimboljših atributskih funkcij z uporabo enostavnega modela iz strojnega učenja. Na tekmovanju SemEval2010 sta najboljša sistema uporabljala seznam pravil in množico klasifikatorjev (tj. odločitvena drevesa, naivni Bayes ali modeli največje entropije). Nazadnje so zmagovalci tekmovanja CoNLL2012 Fernandes in sod. [125] iskali najboljše gručenje omenitev z uporabo prilagojenega perceptrona in predstavitev omenitev v obliki koreferenčnih dreves. Edini sistem, ki je uporabljal faktorske grafe prvega reda, so zasnovali Cullota in sod. [112] na podatkovni množici ACE. Njihova zasnova je popolnoma drugačna od naše, saj še vedno primerjajo le posamezne pare omenitev in nato uporabijo klasifikator prvega reda nad gručami omenitev. Ostali pristopi, ki uporabljajo modele CRF in so bili omenjeni v pregledu sorodnega dela, so bili evalvirani le na zelo omejenih množicah koreferenčnosti ali so bili prilagojeni za združevanje entitet, ki je soroden problem odkrivanju koreferenčnosti.

3.3.4 Pričakovana točnost na splošnih podatkih

Poleg standardne evalvacije naučimo model SkipCor na eni podatkovni množici in ga testiramo na drugi (tabela 3.10). Čeprav podatkovne množice niso imele enakih pravil označevanje ali domene, so rezultati zanimivi, saj so pričakovani v realnem svetu, ko uporabniki želijo, da se sistem obnaša čimbolj robustno.

Najprej opazimo malenkostni padec uspešnosti modela, ko ga testiramo znotraj podmnožic iste podatkovne množice. Na primer, rezultati med časopisnimi in televizijskimi novicami se skoraj ohranijo, če ločeno opazujemo rezultate na množicah CoNLL in ACE2004. Poleg tega dosegajo modeli CoNLL malenkostno slabše rezultate na množici ACE2004 kot originalni rezultati. Obratno se modeli ACE2004 izkažejo slabše na množici CoNLL s približno 20% poslabšanjem. Zanimivo je, da so modeli ACE2004 in CoNLL dosegli slabše ocene MUC na množici SemEval, a najboljše ocene BCubed

in CEAF. Razlika je verjetno v tem, da SemEval vsebuje le nominalne omenitve in hevristično določene entitete z eno omenitvijo, ki jih je precej enostavneje odkriti in zato doprinesejo k ocenama. Model, ki je bil naučen na množici SemEval, se je na množicah CoNLL in ACE2004 izkazal najslabše, a je kljub temu na podatkovni množici CoNLL dosegel boljše ocene MUC kot na lastni množici.

Rezultati tipično kažejo znižanje v uspešnosti na drugih domenah ali podatkovnih množicah, kot je bil naučen izvorni model. Podobne analize so bile že izvedene [145] in prav tako potrjujejo domneve, da evalvacija na istem tipu podatkovne množice daje najboljše rezultate.

Tabela 3.10

Primerjava rezultatov med podatkovnimi množicami ACE2004 (tj. A), CoNLL2012 (tj. C) in SemEval2010 ter izvedenkami časopisnih (tj. NW) in televizijskih (tj. BN) novic, ko je sistem SkipCor naučen na eni in testiran na drugi. Uporabljene ocene so MUC [131], BCubed [132] in CEAF [133].

	<i>Model</i>				
<i>Podatki</i>	<i>A-BN</i>	<i>A-NW</i>	<i>C-BN</i>	<i>C-NW</i>	<i>SemEval2010</i>
<i>A-BN</i>	74, 78, 54	72, 77, 39	65, 70, 28	64, 69, 29	42, 71, 49
<i>A-NW</i>	72, 73, 42	73, 75, 58	60, 64, 27	59, 69, 29	42, 67, 50
<i>C-BN</i>	33, 56, 37	40, 58, 39	68, 70, 43	65, 70, 27	57, 64, 31
<i>C-NW</i>	39, 57, 39	41, 59, 41	67, 66, 28	68, 70, 48	56, 64, 32
<i>SemEval2010</i>	19, 82, 70	23, 85, 74	39, 76, 40	39, 77, 33	42, 87, 76

3.3.5 Ročni pregled odkritih koreferenčnosti

V tabeli 3.11 prikazujemo ročno izbrane entitete, ki jih odkrije predlagan algoritem SkipCor, in jih primerjamo z entitetami, ki bi jih moral odkriti glede na podatke v podatkovni množici. V nekaterih primerih SkipCor ne odkrije koreferenčnosti med omenitvami, ki zato ostanejo razpoznane kot ločene entitete – na primer pri Carli Del Ponte. V drugih primerih se zgodi, da več entitet prepoznamo kot eno – na primer osebi Slobodana Miloševića in Vojislava Koštunice smo prepoznali kot eno entiteto. Včasih nekatere omenitve napačno pripišemo neki entiteti – na primer Zoranu Đinđiću smo dodali omenitvi “shot”. Del entitet pa seveda odkrijemo popolno, kot na primer v zadnjem dokumentu v tabeli 3.11. Delež pravilno odkritih celotnih entitet najbolj natančno opisuje ocena CEAF.

Z natančnejšo analizo napak pri odkrivanju koreferenčnosti so se raziskovalci začeli

Tabela 3.11

Primerjava gruč omenitev, ki so označene v podatkovni množici (ciljne entitete) in gruč omenitev, ki jih odkrije SkipCor (identificirane entitete). Rezultati so pridobljeni iz podatkovne množice CoNLL 2012.

Ciljne entitete	Identificirane entitete
Gruča(Milosevic 's successor, Vojislav Kostunica, a critic of the War Crimes Tribunal at The Hague, He, Kostunica),	Gruča(Milosevic 's, Milosevic 's successor, Vojislav Kostunica, a critic of the War Crimes Tribunal at The Hague, Milosevic, He, Milosevic, Kostunica, former President Slobodan Milosevic),
Gruča(Carla Ponte, The chief U.N. war crimes prosecutor),	Gruča(Carla Ponte),
Gruča(Milosevic 's, Milosevic, Milosevic, former President Slobodan Milosevic)	Gruča(The chief U.N. war crimes prosecutor)
Gruča(Israel and the Palestinians, The two sides),	Gruča(Israel and the Palestinians),
Gruča(Israel, Israel, Israel)	Gruča(The two sides), Gruča(Israel, Israel, Israel)
Gruča(shot, shot, an assassination attempt),	Gruča(an assassination attempt),
Gruča(Belgrade, Belgrade),	Gruča(Belgrade, Belgrade),
Gruča(The Serbian Prime Minister, Zoran Djindjic, Zoran Djindjic, the Prime Minister, his, he, his, he)	Gruča(shot, The Serbian Prime Minister, Zoran Djindjic, Zoran Djindjic, the Prime Minister, his, shot, he, his, he)
Gruča(Northern Ireland, Northern Ireland),	Gruča(Northern Ireland, Northern Ireland),
Gruča(President Clinton, he, Mr. Clinton 's, Mr. Clinton, He, he)	Gruča(President Clinton, he, Mr. Clinton 's, Mr. Clinton, He, he)

ukvarjati šele nedavno [146–148]. Raziskovalca Kummerfeld in Klein [147] sta definirala klasifikacijo napak in predlagala sistem za evalvacijo, ki ugotovi število potrebnih transformacij, ki so potrebne, da vrnjen rezultat preoblikujejo v pričakovan rezultat.

Definirala sta naslednjih pet transformacij:

Posodobi razpon (angl. Alter Span): preoblikuje nepravilno razpoznane meje omenitev, da se ujemajo s pravimi omenitvami.

Razbij (angl. Split): razbije razpoznane entitete na pravilne entitete, ki vsebujejo koreferenčne omenitve.

Izbriši (angl. Remove): izbriše vse omenitve, ki niso označene kot omenitve v podatkovni množici.

Predstavi (angl. Introduce): ustvari entiteto za vsako omenitev (angl. singleton), ki je sistem ne razpozna.

Združi (angl. Merge): združi vse entitete tako, da tvorijo pravilen nabor entitet, kot so označene v podatkovni množici.

Iz seznama izvedenih transformacij, ki omogočijo, da iz razpoznanih entitet pridobimo ciljne entitete, lahko razberemo napake, ki so se zgodile pri odkrivanju koreferenčnosti. V tabeli 3.12 prikazujemo preslikavo med transformacijami in napakami ter podamo primer napake, ki jo naredi sistem SkipCor na podatkovni množici CoNLL2012-BN-Test. Nadalje v tabeli 3.13 prikazujemo seštevke posameznih tipov napak, ki se zgodijo pri odkrivanju koreferenčnosti. Ker je sistem zasnovan le za odkrivanje koreferenčnosti in ne tudi za razpoznavanje omenitev, posledično ne napravimo nobenih napak tipa dodatna entiteta, saj omenitve preberemo iz podatkovne množice. Podobno bi moral biti tudi seštevke napak tipa razpon enak nič, vendar v tem primeru pri predstavitvi podatkov nekaj omenitev (tri) napačno preberemo. Posledično imamo zato tudi tri napake tipa dodatna omenitev, saj se je v vseh treh primerih zgodilo, da je bila vsaka omenitev predstavljena v dveh delih.

Predhodne primerjave sistemov na podatkovni množici CoNLL 2011 kažejo, da je večinoma najbolj pogosta napaka pri odkrivanju koreferenčnosti deljena entiteta [147]. Omenjena analiza je prav tako pokazala, da pri izboljševanju sistemov navadno zmanjšamo število napak enega tipa in povečamo število napak drugega tipa. Iz rezultatov sistema SkipCor (tabela 3.13) opazimo, da naredimo največ napak, ko namesto dveh entitet odkrijemo eno (mešana entiteta) ali obratno (deljena entiteta). Temu sledi število napak, kjer pravilno odkrijemo entiteto, vendar ji ne pridružimo vseh omenitev

Tabela 3.12

Klasifikacija napak pri odkrivanju koreferenčnosti. Napake in transformacije so natančno definirane v [147]. Primeri rezultatov razpoznav so rezultati sistema SkipCor na podatkovni množici CoNLL2012-BN-Test.

Transformacija	Napaka	Rezultat razpoznave	Pravilen rezultat
Posodobi razpon	Razpon	the world at its	the world at its mits
Predstavi & Združi*	Manjkajoča entiteta	– –	Michael Sullivan Mike O’Sullivan
Razbij & Izbriši*	Dodatna entiteta	–	–
Predstavi & Združi	Manjkajoča omenitev	their President The Iraqi leader –	their President Saddam The Iraqi leader
Razbij & Izbriši	Dodatna omenitev	they their time	they their –
Združi	Deljena entiteta	UN ₁ United Nations ₂	UN ₁ United Nations ₁
Razbij	Mešana entiteta	The USS Cole ₁ It ₁ the attack ₁	The USS Cole ₁ It ₁ the attack ₂

* Operacije izvedene večkrat.
V primerih, kjer je navedenih več entitet, so označene s podpisano številko. Sistem SkipCor ni zgeneriral nobene dodatne entitete, zato za ta primer napake ni podanega primera.

(manjkajoča omenitev). Opazimo tudi, da se pojavi nekaj napak tipa manjkajoča entiteta, čeprav SkipCor v rezultatu vrne vse omenitve. Te napake štejemo tudi v primeru, ko entiteta sestoji samo iz omenitev, ki jih predstavljajo zaimki. Takšna odločitev je smiselna, ker je večina ujemajočih se zaimkov koreferenčnih in torej, če vrnemo entiteto, ki sestoji le iz zaimkov, je doprinos podoben, kot če entitete sploh ne bi vrnili.

Glede na evalvacijo v prejšnjem razdelku in ročni pregled rezultatov bi našo metodo

Tabela 3.13

Seštevk posameznih tipov napak pri odkrivanju koreferenčnosti, ki jih sistem SkipCor naredi na podatkovni množici CoNLL2012-BN-Test.

<i>Napaka</i>	<i>Število</i>
Razpon	3
Manjkajoča entiteta	124
Dodatna entiteta	0
Manjkajoča omenitev	255
Dodatna omenitev	3
Deljena entiteta	399
Mešana entiteta	568

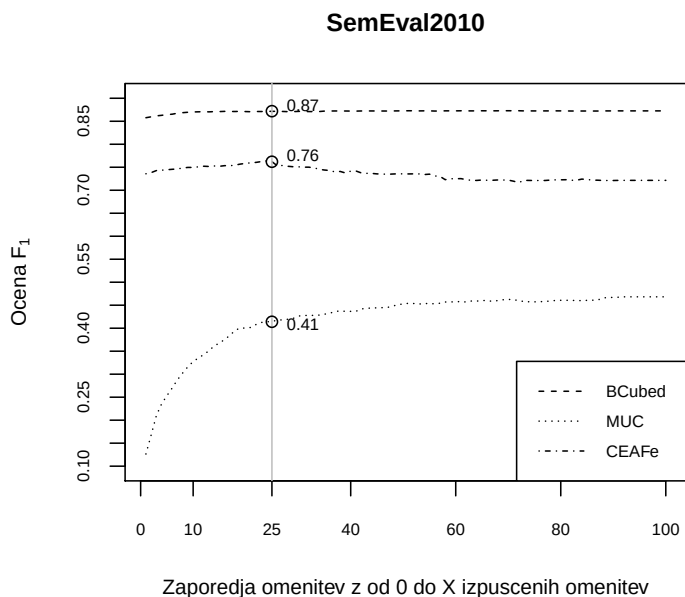
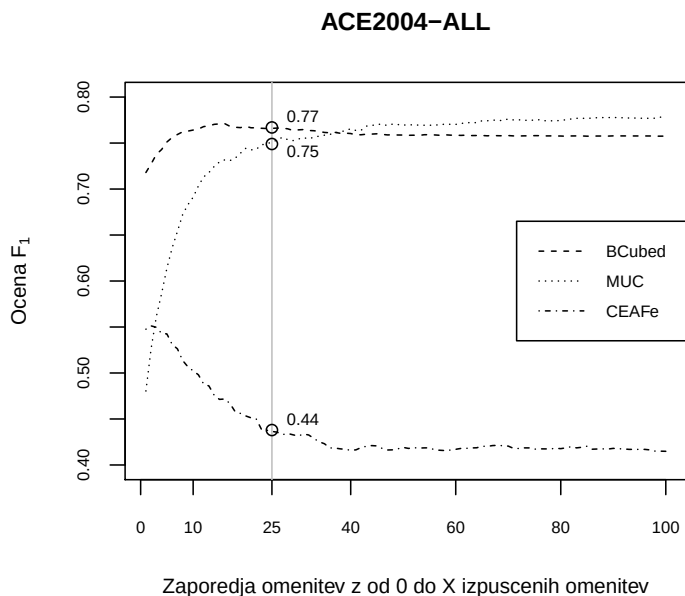
predlagali za uporabo, ko že imamo podatke predstavljene v obliki zaporedij in lahko zaradi tega uporabimo enake atributske funkcije kot pri ostalih nalogah označevanja zaporedij (npr. prepoznavanje imenskih entitet). Pri ročnem pregledu rezultatov smo opazili, da se SkipCor zelo dobro odreže na krajših dokumentih. Sklepamo, da bi lahko torej njegovo uspešnost izboljšali z uteženim upoštevanjem rezultatov različnih modelov. To bomo storili v prihodnjem delu, ko bomo zasnovali inteligentnejši postopek gručenja omenitev.

3.4 Sklep

V poglavju smo predstavili sistem SkipCor za odkrivanje koreferenčnosti, ki za svoje delovanje uporablja le linearno-verižne modele CRE. Z namenom identifikacije vseh možnih koreferenčnosti v besedilu smo razširili osnovni algoritem in mu kot vhod podajali različna zaporedja omenitev. Koreferenčnosti na daljših razdaljah smo tako identificirali s t.i. zaporedji z izpuščenimi omenitvami. V primerjavi z večino obstoječih pristopov nismo preverjali le parov omenitev ali uporabili le enega modela. Predlagan sistem SkipCor je možno popolnoma paralelizirati, kar omogoča izvajanje v linearni časovni zahtevnosti glede na število omenitev v dokumentu.

Predlagan sistem smo evalvirali na podatkovnih množicah, ki vsebujejo označene podatke za problem odkrivanja koreferenčnosti. Rezultate smo nato primerjali z osnovnima izvedenkama sistema SkipCor in ostalimi najboljšimi znanimi pristopi na področju. Dosegli smo popolnoma primerljive rezultate z ostalimi znanimi sistemi in hkrati tu-

di utemeljili, zakaj je uvedba zaporedij izpuščenih omenitev smiselna. Poleg tega smo preverili tudi pričakovano natančnost algoritma nad realnimi podatki, kjer smo ugotovili, da se uspešnost odkrivanja koreferenčnosti zmanjša, če zamenjamo domeno ali slog pisanja besedila.



Slika 3.6

Odkrivanje koreferenčnosti z izpuščenimi omenitvami. Evalvacija predlaganega sistema SkipCor na celotnih podatkovnih množicah ACE 2004 [38] in SemEval2010 [39] z ocenami BCubed [132], MUC [131] and CEAF [133].



Ekstrakcija povezav

4.1 O ekstrakciji povezav

Ekstrakcija povezav želi ekstrahirati povezave med omenitvami v besedilih. V tem poglavju se osredotočimo na iskanje povezav v biomedicinskih znanstvenih člankih [149]. Sistemi za ekstrakcijo povezav [21] odkrivajo semantične povezave med deli besedila. Prvotni sistemi so uporabljali ročno definirana pravila, s katerimi so lahko nad domenskimi besedili ekstrahirali majhen nabor tipov povezav [56]. Kasneje so raziskovalci prilagodili metode strojnega učenja in omogočili razpoznavanje širšega nabora povezav [150–152]. V primerih, kjer ni bilo na voljo označenih podatkov, so predlagali postopke nenadzorovanega učenja, ki odkrijejo le opise povezav in jih ne klasificirajo glede na njihov pomen. Trenutno najboljši znani pristopi uporabljajo kombinacijo metod strojnega učenja in hkrati uporabljajo ročno definirana pravila [153, 154].

V tem poglavju predstavimo sistem za ekstrakcijo povezav za namene izgradnje genskega regulatornega omrežja za organizem *Bacillus subtilis*. Rekonstrukcija genskih regulatornih omrežij je pomemben izziv, ki omogoča izboljšano razumevanje celičnih procesov in molekularnih interakcij v organizmu [155–157]. Za te namene predlagamo sistem z večkratnimi in zaporednimi procesiranjmi, ki uporabljajo linearno-verižne modele CRF in ročno definirana pravila za ekstrakcijo povezav med različnimi geni iz nestrukturiranih besedil (slika 4.5). Uporabo linearno-verižnih modelov CRF smo že uspešno predstavili za odkrivanje koreferenčnosti [15], kjer se omenitve lahko podobno kot pri ekstrakciji povezav pojavljajo na daljših razdaljah (poglavje 3). Med omenitvami smo odkrivanje zakonitosti omogočili s prilagojenimi transformacijami v zaporedja z izpuščenimi omenitvami. S predlaganim sistemom smo na tekmovanju BioNLP 2013 v okviru konference *Association for Computational Linguistics*¹ dosegli najboljši rezultat, ki smo ga kasneje še dodatno izboljšali.

4.1.1 Umestitev v sorodno delo

Večina metod za ekstrakcijo povezav se osredotoča na binarne povezave med dvema entitetama. Predlagani sistemi so pogosto evalvirani na socialni domeni, kjer je cilj odkrivati povezave med ljudmi in organizacijami. Najboljše rezultate dosegajo z uporabo metod iz strojnega učenja, kot so na primer prikriti markovski modeli [158], pogojno naključna polja [90], modeli največje entropije [159] ali splošni binarni klasifikatorji, med katerimi prednjači metoda podpornih vektorjev [160].

¹<http://2013.bionlp-st.org/tasks/gene-regulation-network>

Ista povezava je lahko v prostem besedilu zapisana na mnogo načinov, kar skušajo ustrezno obvladati predvsem metode iz strojnega učenja, ki so naučene nad velikimi podatkovnimi množicami. Poleg tega se povezave pojavljajo redkeje kot na primer omenitve ali imenske entitete, zato so Garcia in sod. [161] predlagali posebne leksikografsko-sintaktične atributske funkcije, ki pomagajo pri identifikaciji potencialnih povezav in pravilni določitvi odvisnosti. V literaturi pogosto zasledimo tudi mešanje terminov ekstrakcija povezav in ekstrakcija opisov povezav (angl. relation descriptors). Pri opisu povezav nas ne zanima celotna povezava z osebkom in predmetom, temveč le delček besedila, ki lahko predstavlja povezavo. V takšnih primerih se uporabljajo enake metode kot pri prepoznavanju imenskih entitet in tudi podatkovne množice so označene z notacijo IOB (angl. inside-outside-beginning) [162], kjer je prva beseda povezave označena kot *B-REL*, vsaka naslednja z *I-REL* in ostale besede z *O*. Li in sod. [63] za ekstrakcijo opisov povezav uporabijo linearno-verižni CRF, pri čemer predhodno označijo vrednosti osebkom in predmetov s posebnimi vrednostmi, da znajo napovedati tudi smer morebitne povezave. V našem delu uporabimo enak model, vendar ga učimo nad zaporedji omenitev in posledično odkrivamo celotno povezavo, vključno z osebkom in predmetom. Banko in Etzioni [60] sta uporabila modele prvega reda za nenadzorovano ekstrakcijo povezav (angl. open relation extraction), kjer sta označila le dele besedila, ki predstavljajo povezavo, brez nadaljnje klasifikacije tipa povezave ali njenih atributov. Model sta naučila nad sintaktičnimi vzorci in z uporabo sopomenk [163]. Njuna metoda je uporabna za obdelavo ogromnih količin podatkov, iz katerih želimo izvesti grobe lastnosti, ki bi jih želeli uporabiti v nadaljnjih postopkih.

Najbolj uporabljana podatkovna množica za evalvacijo sistemov za ekstrakcijo socialnih povezav med ljudmi in organizacijami je ACE 2004 [164–166], kjer so povezave organizirane v dvonivojsko hierarhijo in usmerjene med omenitvami, ki predstavljajo bodisi osebek ali predmet. Wang in sod. [166] so za odkrivanje teh povezav uporabili model SVM s pristopom “ena-proti-ena” z dodatnimi semantičnimi značilkami iz leksikona WordNet [86]. V tem delu odkrivamo povezave v treh nivojih in namesto binarne klasifikacije uporabljamo pravila in linearno-verižni model CRF.

Poleg odkrivanja splošnih povezav se je v zadnjem povečala potreba po ekstrakciji povezav iz biomedicinskih podatkov. Eno prvih organiziranih tekmovanj na tem področju – *Learning Language in Logic* (LLL) se je ukvarjalo z odkrivanjem genskih interakcij, kar je podobno tekmovanju BioNLP GRN 2013, ki vsebuje del dodatno označenih podatkov in na katerem smo s predlagano metodo v tem poglavju sodelo-

vali tudi mi. Giuliano in sod. [59] so kasneje nad podatki iz LLL predlagali izboljšan sistem za ekstrakcijo povezav, pri čemer so uporabili metodo SVM z lokalnim in globalnim jedrom. Pri lokalnem jedru so uporabili lastnosti trenutne omenitve, kot je na primer beseda, lema ali oblikoslovna oznaka, medtem ko je globalno jedro primerjalo besede, ki se pojavljajo levo, desno ali med obema omenitvama. Pri ekstrakciji povezav izbirajo le povedi, ki vsebujejo vsaj dve omenitvi in iz vsake primerne povedi zgenerirajo $\binom{n}{k}$ primerov, kjer je n število omenitev v povedi in k število omenitev, ki lahko tvori povezavo (tj. dve). Smer povezave razpoznavajo s transformacijo ekstrakcije v tri-razredni problem, kjer dva razreda definirata smer povezave in tretji njen neobstoj. Tudi naš pristop preko atributskih funkcij uporablja kontekstualne in sintaktične lastnosti, vendar se ne ukvarja s smerjo povezave, ker jo po ekstrakciji določimo ročno s pomočjo pravil.

Tekmovanja BioNLP

Serijska tekmovanja BioNLP se osredotoča na ekstrakcijo različnih tipov informacij iz biomedicinskih podatkov. Prvo tekmovanje je bilo organizirano leta 2009 [28], ki sta mu sledili še tekmovanja v letu 2011 [167] in 2013 [28]. Vsakič je predstavitev rezultatov in predlaganih pristopov potekala kot samostojna enota v okviru konference *Association for Computational Linguistics* (ACL). Tekmovanje iz leta 2009 je vzbudilo začetno navdušenje nad različnimi nalogami ekstrakcije informacij. Nato je drugo tekmovanje predstavilo dodatne splošne domene, več podatkovnih tipov in podporo različnim tipom dogodkov (tj. dogodek je tip povezave med dvema entitetama), medtem ko je zadnje tekmovanje vključilo še uporabo semantičnih podatkov in odkrivanje molekularnih mehanizmov pri raku, genskih poti, regulatornih omrežij ter avtomatsko dopolnjevanje ontologij.

Del tekmovanja BioNLP 2011 se je že osredotočal na ekstrakcijo povezav med entitetami. Tedaj so Landeghem in sod. predlagali sistem TEES [160], ki je s pomočjo modelov SVM uspešno razpoznaval entitete in povezave med vsemi možnimi pari v isti povedi. Hkrati so ugotovili, da modul za razpoznavo posameznih omenitev zelo vpliva na končno uspešnost sistema. V našem primeru predstavljajo omenitve proteini in geni, ki so predhodno že razpoznani, in je torej naša glavna naloga ekstrakcija dogodkov, povezav in dogodkov, ki so vezani le na eno omenitev.

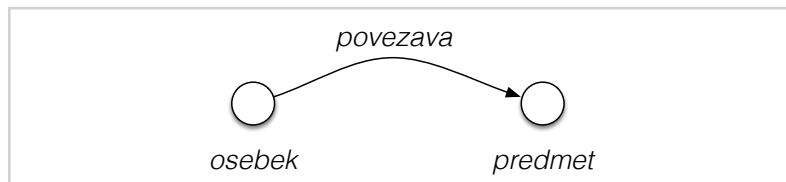
V okviru tekmovanja BioNLP 2013 smo predlagali svoj pristop za odkrivanje genskih regulatornih mrež [13], ki smo ga kasneje še dodatno nadgradili [16] in ga opisuje-

mo v nadaljevanju. Cilj odkrivanja genskih regulatornih mrež iz povzetkov znanstvenih člankov [168] je ekstrakcija interakcij med posameznimi geni z namenom, da se ponostavi postopek odkrivanja vzročno posledičnih vplivov med geni. Ročno označena podatkovna množica vsebuje označene dele besedil, ki predstavljajo omenitve, kot so geni, proteini in reguloni. Poleg tega so označeni še dogodki in interakcije med geni, ki jih je bilo treba ekstrahirati in na podlagi česar so bili vsi sodelujoči sistemi tudi rangirani.

4.2 Predstavitev podatkov

V tem razdelku definiramo predstavitev podatkov, ki je potrebna za ustrezno ekstrakcijo povezav z linearno-verižnimi modeli CRF, vključno z uporabljenimi atributskimi funkcijami. Nazadnje še na reprezentativni povedi iz učne množice prikažemo delovanje ekstrakcije in s tem še dodatno potrdimo ustreznost predlagane predstavitve.

Pri ekstrakciji povezav moramo ugotoviti, med katerima dvema omenitvama obstaja povezava in kakšen je pomen te povezave. Med omenitvami moramo torej poiskati nekatere zakonitosti. Hkrati je potrebno upoštevati, da sta lahko omenitvi, med katerima obstaja povezava, v povedi na poljubnih razdaljah. Predstavitev povezave prikazujemo na sliki 4.1 in kot smo že omenili, le-ta sestoji iz imena ter osebk in predmeta, ki predstavljata atributa povezave.



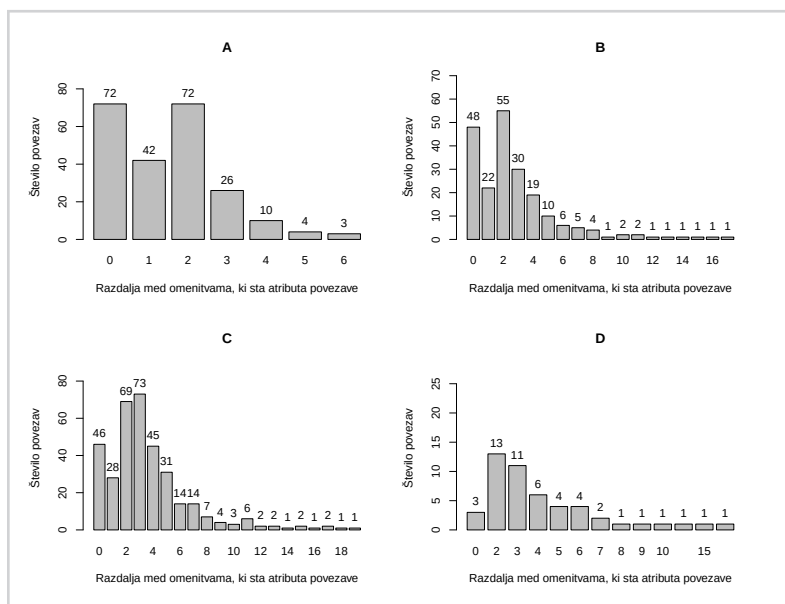
Slika 4.1

Splošna predstavitev povezave. Vsako povezavo (npr. *gerE inhibits cotD*) enolično predstavimo z imenom (npr. *Interaction.Regulation*), osebkom (npr. *gerE*) in predmetom (npr. *cotD*).

Zaradi uporabe linearno-verižnih modelov CRF smo omejeni le na predstavitev podatkov v obliki zaporedij (slika 2.6 (c)). Modeli CRF poskušajo glede na verjetnostno porazdelitev čimbolj pravilno napovedati verjetnost celotnega zaporedja $P(\bar{y}|\bar{x}, \lambda)$, kjer je $\bar{y}$ želeno zaporedje glede na vhodno podano zaporedje $\bar{x}$ in predhodno naučene uteži atributskih funkcij λ . Za ekstrakcijo povezav z zaporedjem $\bar{x}$ predstavimo zaporedje omenitev, ki lahko predstavljajo attribute povezav, in z zaporedjem $\bar{y}$ označimo tip povezave med omenitvama x_{i-1} in x_i . Ciljne oznake $y_i \in \{\text{MOŽNE POVEZAVE}\} \cup$

{O} definiramo z imenom povezave ali z oznako O ugotovimo nepovezanost med omenitvama.

Na sliki 4.2 prikazujemo porazdelitev razdalj med atributi povezav v podatkovni množici GRN BioNLP 2013. Atributi (tj. osebek ali predmet povezave) sestojijo iz omenitve entitet ali dogodkov. Tudi dogodke predstavimo kot tip povezav, ki kot attribute vsebujejo omenitev gena in opis spremembe nad genom. Porazdelitev razdalj med atributi dogodkov je predstavljena v razdelku (A). Razdelek (B) prikazuje razdalje med atributi povezav, ki so v podatkovni množici identificirani kot navadne omenitve (tj. ena ali več zaporednih besed). V učni podatkovni množici obstaja 142 povezav, ki imajo kot enega izmed atributov dogodek – porazdelitev razdalj v razdelku (C). Enajst primerov kot atribut vsebuje drugo povezavo in štiri od njih predstavljajo negacijo, vendar slednje niso upoštevane pri evalvaciji. Za namene evalvacije ni potrebno odkrivati hierarhičnih povezav, zato pri postopku v ekstrakcijo vključimo še dodaten korak, kjer razpoznavamo le povezave med geni *B. subtilis*. Razdalje med omenitvami slednjega tipa so prikazane v porazdelitvi (D).



Slika 4.2

Porazdelitve razdalj med atributi različnih tipov povezav v podatkovni množici BioNLP GRN. (A) Razdalje med omenitvami dogodkov. (B) Razdalje med omenitvami nehierarhičnih povezav. (C) Razdalje med atributi povezav, kjer je vsaj eden od atributov dogodek. (D) Porazdelitev razdalj med omenitvami, ki predstavljajo gene *B. subtilis*.

Glede na porazdelitve opazimo, da ima večina povezav svoje attribute na razdalji dveh omenitev. Podatki torej kažejo, da moramo zaporedja za modele prvega reda predstaviti nekoliko drugače, da jih bomo lahko uspešno odkrivali. V ta namen v nadaljevanju predstavimo zaporedja izpuščenih omenitev (razdelek 4.2.2).

4.2.1 Atributske funkcije

Izbira ustreznih atributskih funkcij zelo vpliva na uspešnost ekstrakcije. Navadno jih definiramo kot predloge in nato končne atributske funkcije (enačba (2.4)) dobimo s sprehomod čez podatkovno množico. Funkcije, ki jih uporabljamo z našo metodo, so predstavljene v tabelah 4.1 in 4.2. V slednji so zapisani generatorji, ki glede na parametre ustvarijo več atributskih funkcij.

4.2.2 Primer ekstrakcije povezav

Pri ekstrakciji povezav želimo razpoznati različne tipe povezav med omenitvami. Na vходу sprejmemo poved z že označenimi omenitvami, kot prikazujemo v sledečem primeru.

“Transcription of *cheV* initiates from a *sigma D*-dependent *promoter* element both in vivo and in vitro, and expression of a *cheV*-lacZ fusion is completely dependent on *sigD*.”²

Iz identificiranih omenitev bi za linerano-verižni model CRF zgradili osnovno zaporedje omenitev, ki je prikazano na sliki 4.3. Zaporedje $\bar{x}$ je sestavljeno iz omenitev (tj. opazovanih vrednosti), saj ne želimo odkrivati le opisov povezav, temveč celotne povezave, kjer omenitve predstavljajo njihove attribute. Zaporedje oznak $\bar{y}$, ki ga želimo napovedati, pa sestoji iz imena povezave med trenutno x_i in prejšnjo omenitvijo x_{i-1} ali oznake O , ki pomeni, da med njima povezava ne obstaja.

V primeru, da obravnavamo le osnovno zaporedje, ne moremo odkriti povezav, ki obstajajo med omenitvami na daljših razdaljah (tj. kjer se vmes nahajajo še druge omenitve). Iz vizualizacije zgornjega primera na sliki 4.3 opazimo, da ne moremo odkriti povezave *Promoter of*, ki obstaja med omenitvama *cheV* in *promoter*, saj v zaporedju nista neposredni sosedi. Zaradi tega in potencialnih omenitev na še daljših razdaljah predlagamo transformacijo osnovnega zaporedja (tj. z nič izpuščenimi omenitvami) v

²Primer povedi iz učne podatkovne množice GRN BioNLP 2013, članek PMID-8169223-S5.

Tabela 4.1

Opis atributskih funkcij. Vse funkcije so uporabljene v vseh modelih CRF v obeh faktorjih. Natančen opis (npr. katere vrednosti ali dodatne oznake omenitev so uporabljene) in implementacija sta dostopna v javnem repozitoriju programske kode [20] (v razredu *FeatureFunctionPackages*).

Naziv	Opis
Porazdelitev ciljnih oznak	Verjetnost ciljne oznake.
Velika začetnica	Ali se trenutna/prejšnja omenitev začne z veliko začetnico.
Dvojna velika začetnica	Ali se obe zaporedni omenitvi začneta z veliko začetnico.
Hearstova pravila	Ali med omenitvami veljajo koreferenčna pravila [126].
Razdalja členov	Število členov, ki se nahajajo med zaporednima omenitvama.
Globina odvisnosti	Globina omenitve v drevesu odvisnosti.
Starševska odvisnost	Oznaka starša v drevesu odvisnosti.
Pot odvisnosti	Oznake v drevesu odvisnosti na poti med omenitvama.
BSubtilis	Če sta obe omenitvi gena v <i>B. subtilis</i> , kakšna je verjetnost protein-protein interakcije v bazi STRING [169], tj. zelo majhna, majhna, srednja, visoka, zelo visoka.
Je BSubtilis	Ali je trenutna omenitev gen v <i>B. subtilis</i> .
Je Bsubtilis par	Katera od dveh zaporednih omenitev je gen v <i>B. subtilis</i> , tj. leva, desna, oba, nobena.

zaporedja z izpuščenimi omenitvami. Število izpuščenih omenitev torej določa, koliko ostalih omenitev se nahaja med dvema zaporednima omenitvama v vhodnem dokumentu. Torej, za ekstrakcijo povezave *Promoter of* z linearno-verižnimi modeli CRF ustvarimo zaporedji z eno izpuščeno omenitvijo, ki nam omogočata, da razpoznamo vse povezave na razdalji ena. Preoblikovano zaporedje, ki to omogoči, je prikazano na sliki 4.4.

Tabela 4.2

Opis generatorjev atributskih funkcij. Vse funkcije so uporabljene v vseh modelih CRF v obeh faktorjih (razen predpona in pripona, ki sta samo tipa unigram). Natančen opis (npr. katere vrednosti ali dodatne oznake omenitev so uporabljene) in implementacija sta dostopna v javnem repozitoriju programske kode [20] (v razredu *FeatureFunctionPackages*).

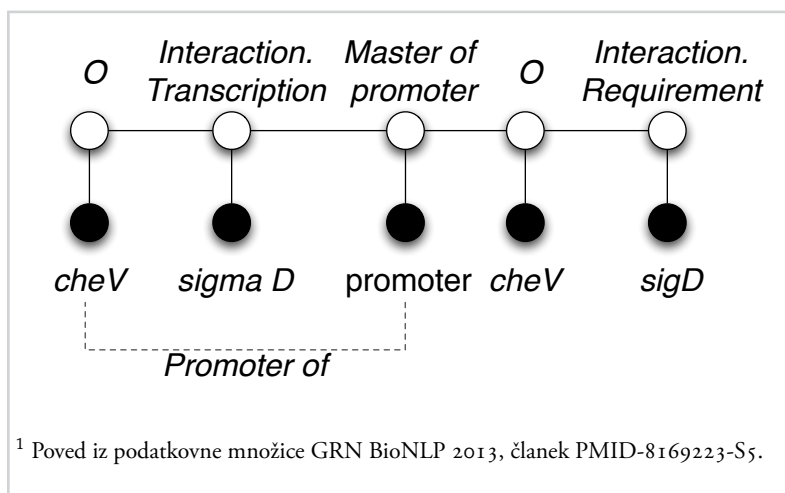
Naziv	Opis
Predpona	Predpone dolžine 2 in 3 na razdaljah od -5 do 5.
Pripona	Pripone dolžine 2 in 3 na razdaljah od -5 do 5.
Zaporedna vrednost	Kombinacija dveh oznak (npr. <i>PDT/NNS</i>) zaporednih omenitev na razdalji od -4 do 4. Uporabljene oznake so beseda, oblikoslovna oznaka, lema, entitetni tip in koreferenčnost.
Trenutna vrednost	Vrednost oznake na razdalji od -4 do 4. Uporabljene oznake so beseda, oblikoslovna oznaka, lema, entitetni tip in koreferenčnost.
Kontekst	Ujemanje v številu n-gramov dolžine 3 na razdalji do 5 členov. Ujemanje je diskretizirano na osem nivojev z Jaccard-ovim koeficientom podobnosti. Različne atributske funkcije primerjajo vrednosti levo / desno / vmes / izzven / unija glede na izbrani omenitvi.
Levo / desno / vmes	Vrednosti členov na levi, desni in med dvema omenitvama na razdalji od 1 do 5. Uporabljene oznake so beseda, oblikoslovna oznaka in lema.
Razbitje	Razbitje omenitve na posamezne člene glede na presledek in upoštevanje prvih dveh členov. Uporabljene oznake so beseda in lema.

4.3 Metoda za ekstrakcijo povezav

V tem razdelku predstavimo sistem z zaporednimi procesiranjmi (slika 4.5), kjer so deli za ekstrakcijo medseboj neodvisni. Sistem sestoji iz devetih komponent, pri čemer prvi dve preoblikujeta podatke v ustrezno predstavitev in nato sledijo glavna procesiranja, ki ekstrahirajo povezave s pomočjo pravil ali linearno-verižnih modelov CRF. V zadnjem delu se podatki še očistijo, popravijo in združijo. Celotna implementacija predlaganega

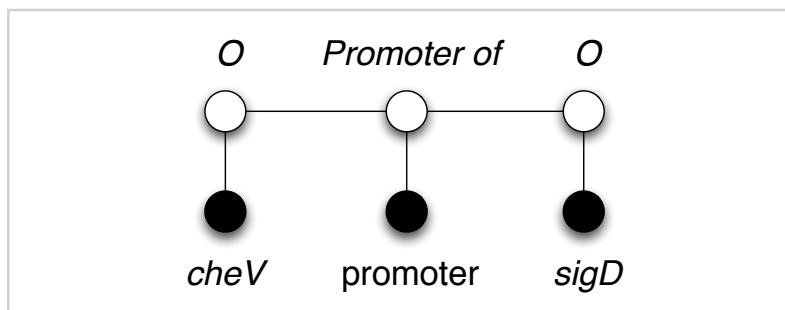
Slika 4.3

Osnovno zaporedje omenitev. Zaporedje vsebuje vse omenitve iz dokumenta "Transcription of *cheV* initiates from a *sigma D*-dependent promoter element both in vivo and in vitro, and expression of a *cheV*-lacZ fusion is completely dependent on *sigD*."¹



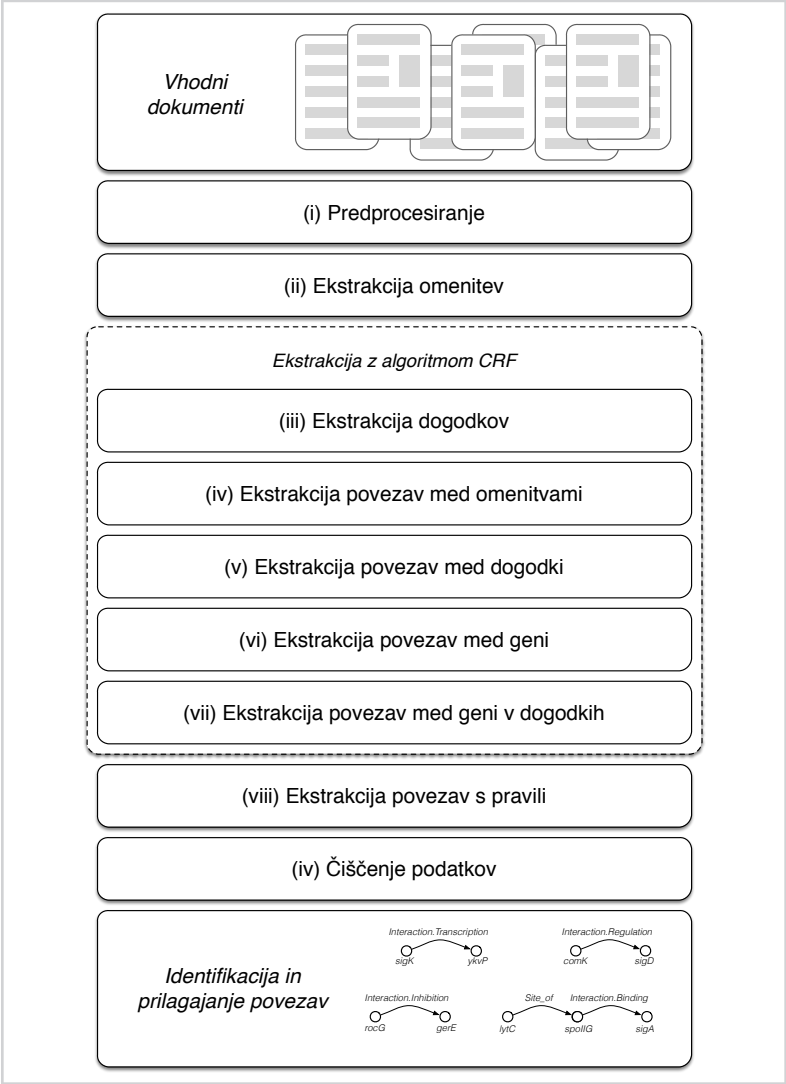
Slika 4.4

Zaporedje z eno izpuščeno omenitvijo. Eno izmed možnih preoblikovanih zaporedij iz osnovnega zaporedja [*cheV*, *sigma D*, *promoter*, *cheV*, *sigD*]. Drugo možno zaporedje, ki se ustvari ob transformaciji je [*sigmaD*, *cheV*].



sistema je na voljo v javnem programskem repozitoriju [20], ki uporablja algoritem CRFSuite [135] za učenje in klasifikacijo z modeli CRF.

Zaporedje procesiranj se izvaja, kot je prikazano na sliki 4.5. Najprej pretvorimo vhodni dokument v ustrezno obliko za nadaljnje procesiranje in ga obogatimo z lemmami, drevesi odvisnosti in oblikoslovnimi oznakami. Nato avtomatsko identificiramo dodaten tip omenitev, ki so lahko atribut dogodkov. Zatem s pomočjo modelov CRF ekstrahiramo dogodke, ki jih predstavimo v obliki povezav. Potem ekstrahiramo različne tipe povezav, vključno s hierarhijami, in nazadnje s pravili uredimo ekstrakcije.



Slika 4.5

Arhitektura predlaganega sistema za ekstrakcijo povezav z večkratnimi procesiranj. Sistem sestoji iz devetih procesiranj, med katerimi prvi dve pripravita podatke, nato naslednjih šest ekstrahira povezave in zadnje procesiranje izvede čiščenje rezultatov. Vsak vhodni dokument procesiramo zaporedno in kot rezultat zanj vrnemo seznam ekstrahiranih povezav.

S pravili ustrezno spremenimo smer povezave (tj. zamenjamo osebke in predmet) in odstranimo duplikate ter nemogoče povezave.

V nadaljnjih podrazdelkih podrobneje opišemo posamezne korake predlaganega pristopa.

4.3.1 Predprocesiranje

V stopnji predprocesiranja ustrezno predstavimo vhodne podatke, razpoznamo povedi in jih razbijemo na posamezne člene. Dodatno še obogatimo podatke z oblikoslovnimi oznakami, drevesi odvisnosti [33] in lemmami [170].

4.3.2 Razpoznavanje omenitev

Omenitve sestojijo iz sledečih tipov: *Protein*, *GeneFamily*, *ProteinFamily*, *ProteinComplex*, *PolymeraseComplex*, *Gene*, *Operon*, *mRNA*, *Site*, *Regulon*. V tem koraku avtomatsko identificiramo dodatne omenitve (npr. *inhibits*, *co-transcribes*), ki predstavljajo enega izmed atributov pri dogodkih. Iz učne množice preberemo vse takšne omenitve in jih nato glede na ujemanje v njihovi lemi odkrivamo v novih dokumentih.

4.3.3 Ekstrakcija dogodkov

Dogodek definira spremembo stanja ene ali več biomolekul, na primer: “*expression of a cheV-lacZ fusion is completely dependent on sigD*”. V našem primeru dogodka predstavimo kot poseben tip povezav z imenom *EVENT*. V učni podatkovni množici so osebki dogodkov predstavljeni v obliki enega izmed sledečih tipov *Protein*, *GeneFamily*, *PolymeraseComplex*, *Gene*, *Operon*, *mRNA*, *Site*, *Regulon* in *Promoter*, medtem ko so predmeti omenitve, ki jih odkrijemo v predhodnem koraku. Dogodke odkrivamo z linearno-verižnimi modeli CRF in nato uporabimo pravila, da ustrezno nastavimo osebke in predmete razpoznanih dogodkov.

4.3.4 Ekstrakcija povezav

Različne tipe povezav ekstrahiramo v štirih procesiranjih (iv, v, vi, vii), kar nam omogoča ekstrakcijo hierarhičnih povezav (tj. povezave, ki vsebujejo drugo povezavo kot enega izmed atributov). Procesiranja uporabljajo linearno-verižne modele CRF in se izvajajo v sledečem zaporedju (primeri ekstrakcij se nanašajo na dokument iz prejšnjega razdelka):

- (iv) Najprej ekstrahiramo povezave, ki kot attribute vsebujejo navadne ali razpoznane omenitve (npr. *sigma D* $\rightarrow$ *Master_of_Promoter* $\rightarrow$ *promoter*). Navadne omenitve predstavljajo gene, proteine in njihove agregate, medtem ko razpoznane omenitve predstavljajo enega izmed atributov pri dogodkih (npr. *transcription*).
- (v) V drugi fazi ekstrahiramo le povezave, ki vsebujejo dogodek kot enega izmed atributov (npr. *expression cheV* $\rightarrow$ *Interaction.Requirement* $\rightarrow$ *sigD*). Pred ekstrakcijo teh povezav najprej dogodka spremenimo v omenitve, ki so sestavljene iz besed njihovih atributov. Nato jih dodamo v seznam ostalih omenitev in uredimo glede na indeks najbolj zgodnje besede v novi omenitvi. Ekstrakcijo povezav izvedemo na enak način kot v prejšnjem koraku.
- (vi) Glavna naloga tekmovanja je ekstrahirati interakcije (tj. povezave tipa *Interaction.**) med geni organizma *B. Subtilis*. Pred ekstrakcijo le teh vrst povezav zato najprej izberemo omenitve, ki predstavljajo gene *B. Subtilis* in nato izvedemo ekstrakcijo s pomočjo linearno-verižnih modelov CRF. Pri izbiri ustreznih omenitev uporabljamo seznam znanih genov iz seznama NCBI³.
- (vii) Evalvacija nad rezultati in dosežen uspeh sta odvisna le od razpoznanih interakcij med geni. Torej v primeru, da obstaja povezava med genom G1 in dogodkom E, potem je ekstrahirano gensko regulatorno omrežje enako, kot če bi razpoznali povezavo med G1 in G2, kjer je G2 predmet dogodka E. Zaradi tega ustrezno preoblikujemo učno množico in naučimo model CRF nad preoblikovanimi podatki (npr. *sigD* $\rightarrow$ *Interaction.Requirement* $\rightarrow$ *cheV*, kjer je *cheV* predmet nekega dogodka).

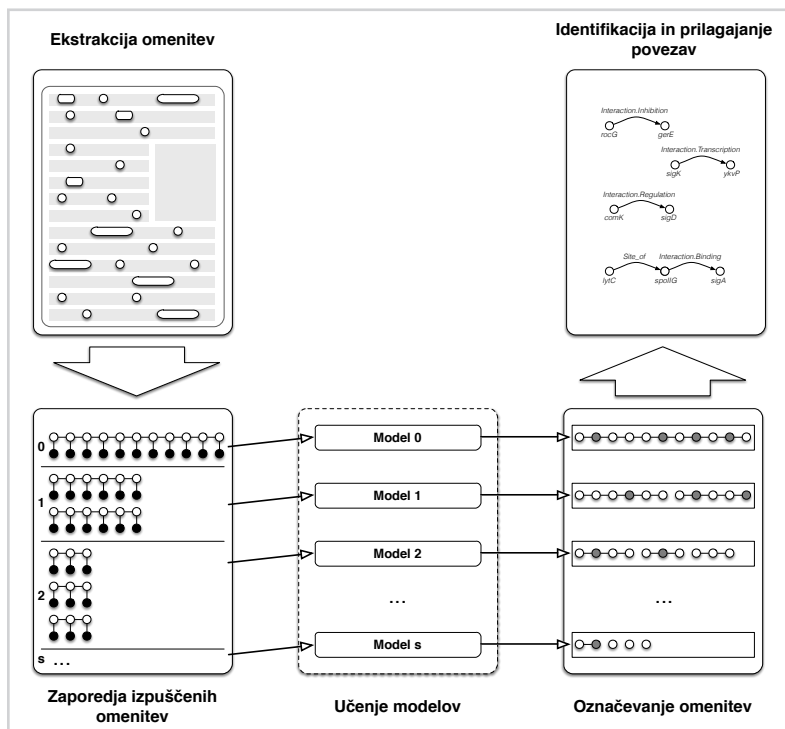
Učna podatkovna množica vsebuje tudi sedem povezav, ki vsebujejo drugo povezavo kot enega izmed atributov. Zaradi premajhnega števila učnih podatkov tega tipa hierarhij nismo odkrivali, vendar pomanjkljivost nadomestimo z ekstrakcijo povezav med geni (vi, vii).

Poleg tega obstajajo še štiri negativne povezave, vendar tekmovanje negativnih povezav ne upošteva v končnem rezultatu. Glede na učno podatkovno množico bi lahko dodali dodatno procesiranje za razpoznavo negativnih povezav, saj morajo biti najprej razpoznane kot pozitivne in nato dodatno označene kot negativne. Pri tem bi uporabili pravila in besede, ki označujejo negativen sentiment (npr. *not*, *whereas*, *neither* ali *nor*).

³<http://www.ncbi.nlm.nih.gov/nuccore/AL009126>

Postopek ekstrakcije povezav z linearno-verižnimi modeli pogojnih naključnih polj

Pri ekstrakciji povezav z linearno-verižnimi modeli CRF najprej zgeneriramo $\lceil \frac{n}{s} \rceil$ zaporedij s $s + 1$ izpuščenimi omenitvami za vsak s , ki je podan kot parameter (primer v razdelku 4.2.2). Nato za vsak s naučimo model in izvedemo označevanje, kar lahko poteka vzporedno, saj so modeli med seboj neodvisni. Za naše namene uporabljamo modele, ki so naučeni na zaporedjih od nič do deset izpuščenih omenitev. V zadnjem koraku iz rezultatov preberemo povezave in jih predstavimo v obliki neusmerjenega grafa. Visokonivojska arhitektura toka podatkov je predstavljena na sliki 4.6.



Slika 4.6

Tok podatkov v procesiranjih za ekstrakcijo povezav z linearno-verižnimi modeli CRF. Najprej transformiramo osnovno zaporedje omenitev v zaporedja z izpuščenimi omenitvami. Na vsakem novem tipu zaporedij naučimo svoj linearno-verižni model CRF in z njim označimo neznane dokumente. Povezave nato preberemo iz pridobljenih oznak in jih vrnemo kot rezultat.

4.3.5 Uporaba ročno definiranih pravil

V zadnjem koraku ekstrakcije definiramo ročna pravila, ki smo jih pridobili s pregledom učne podatkovne množice ali z upoštevanjem sinonimov. Glede na tip pravila jih delimo v štiri razrede:

Trojčki omenitev. Metoda pregleda vse zaporedne trojčke omenitev, ki v sredini vsebujejo natanko eno razpoznano omenitev. Kot vhod ji podamo regularni izraz, s katerim se mora pričeti razpoznan omenitev in ciljni tip povezave.

Zaporedne omenitve. Metoda pregleda vse zaporedne omenitve in preveri, če se besedilo med njima ujema z določenim regularnim izrazom za ekstrakcijo določene povezave. Privzeto predvideva ekstrakcijo iz stavkov v tvorniku, da ustrezno določi osebek in predmet, sicer ju zamenja.

Seznam zaporednih omenitev. Metoda je posplošitev prejšnjih dveh, pri čemer zahteva, da se na levi ali na desni strani nahaja seznam omenitev, ki so med seboj ločene z “,”, “, and” ali “and”. Na primer, iz stavka “*rsfA* is under the control of both $\sigma(F)$ and $\sigma(G)$ ” ta metoda ekstrahira dve povezavi (tj. “ $rsfA \rightarrow Interaction.Activation \rightarrow \sigma(F)$ ” in “ $rsfA \rightarrow Interaction.Activation \rightarrow \sigma(G)$ ”).

Stavki zaporednih omenitev. Zadnja metoda je posplošitev druge in pri iskanju ujemanja ne upošteva podrednih stavkov med omenitvama. Vmesne stavke definiramo le kot del besedila med dvema vejicama. Na primer, metoda ekstrahira povezavo $\sigma(F) \rightarrow Interaction.Activation \rightarrow \sigma(G)$ iz povedi “ $\sigma(F)$ factor turns on about 48 genes, including the gene for *RsfA*, and the gene for $\sigma(G)$ ”.

V biomedicinski literaturi lahko za navedbo ene povezave uporabljamo več izrazov. Na primer, nekaterim je bližje *to repress* v primerjavi z *to inactivate* ali *to inhibit* in obratno. Zaradi upoštevanja uveljavljenih sinonimov še dodatno izboljšamo kvaliteto ekstrakcij. V tabeli 4.3 predstavimo natančen seznam pravil in njihovih parametrov, ki jih uporabimo.

4.3.6 Čiščenje rezultatov

V zadnjem procesiranju izvajamo čiščenje podatkov, kjer odstranimo zanke in redundantne povezave.

Tabela 4.3

Parametri za ekstrakcijo povezav s pravili. Vsaka od metod kot parameter zahteva ciljni tip povezave in regularni izraz. Nekatere zahtevajo tudi ali naj se uporabi aktivni ali pasivni način.

<i>Regularni izraz</i>	
<i>Trojčki omenitev</i>	
Transcription	transcrib
<i>Zaporedne omenitve</i>	
Transcription	.*directs transcription.*
Inhibition	.*inactivate.*
Inhibition	.*inhibits.*
Inhibition	.*repressor to.*
Inhibition ¹	.*is negatively regulated by.*
Activation ¹	.*is governed by.*
Activation ¹	.*essential.*activat.*
Activation	.*to.*activat.*
Activation	.*turns on.*
Requirement ¹	.*requires.*
Requirement	.*required.*
Binding	.*binds.*to.*
Binding	-binding.*
<i>Seznam zaporednih omenitev</i>	
Transcription	.*under.*control.*of.*
Activation ¹	.*is governed by.*
Inhibition	.*represses.*
Inhibition	.*to repress.*
<i>Stavki zaporednih omenitev</i>	
Activation	.*turns on.*
Inhibition	.*repressed.*

¹ Metoda upošteva pasivni način.

Povezava je zanka, če njeni atributi predstavljajo isto entiteto (tj. omenitvi sta koreferenčni). Na primer, stavek "... *sigma D* element, while cheV-lacZ depends on *sigD*

...” vsebuje omenitvi *sigma D* and *sigD*, ki ne moreta biti v interakciji, saj predstavljata isti gen. Odstranitev zank zmanjša število napak (tj. vstavitev), medtem ko odstranitev redundantnih povezav ne vpliva na končen rezultat.

4.4 Analiza rezultatov

Tekmovanje v ekstrakciji povezav za izgradnjo genskega regulatornega omrežja (GRN) predvideva, da sodelujoči razpoznajo povezave med omenitvami in pravilno določijo osebek in predmet povezave. Kot pravilno ekstrahirane povezave štejejo le povezave, ki se ujemajo v imenu in obeh atributih.

Naš pristop temelji na večkratnih procesiranjih vhodnih podatkov, kjer v vsakem procesiranju razpoznavamo določen tip povezav. V nadaljevanju najprej predstavimo splošne rezultate ter analiziramo ekstrakcije preko posameznih procesiranj in različnih podmnožic atributskih funkcij.

4.4.1 Ocenjevalni kriterij

Uradni ocenjevalni kriterij tekmovanja temelji na primerjavi genskega regulatornega omrežja organizma *B. subtilis*, ki ga odkrijemo s pomočjo avtomatske ekstrakcije povezav na podlagi predhodno ročno označenega omrežja iz podatkovne množice. Uspešnost sistema za ekstrakcijo povezav je določena z oceno SER [171] (angl. slot error rate)

$$SER = (S + I + D)/N,$$

ki predstavlja razmerje med vsoto zamenjanih povezav (S), vstavljenih povezav (I) in odstranjenih povezav (D) ter številom povezav v pravilnem omrežju (N). Na kratko, sistemi ki napovejo enako število napačnih in pravičnih povezav, dosežejo vrednost ena ocene SER. Bolj natančni kot so sistemi, manj zamenjanih, vstavljenih ali odstranjenih povezav ustvarijo ter s tem hkrati dosežejo nižjo oceno SER.

4.4.2 Podatkovna množica BioNLP 2013

Podatkovna množica GRN sestoji iz povzetkov člankov iz baze PubMed, ki opisujejo gensko regulatorno omrežje organizma *B. subtilis*. Besedilo je razbito po stavkih z označenimi omenitvami, biokemijskimi dogodki in povezavami, ki opisujejo celične procese na molekularnem nivoju. Povezave, ki so vezane na interakcije, so urejene

hierarhično in klasificirane v tipe *regulation*, *inhibition*, *activation*, *requirement*, *binding* in *transcription*.

Tabela 4.4 prikazuje osnovne statistike podatkovnih množic za učenje, pripravo parametrov (angl. dev) in testiranje. Za testno množico nimamo označenih podatkov in torej zanjo ne moremo prikazati rezultatov analize po posameznih procesiranjih. Ostali dve množici vsebujeta označene omenitve, dogodke in povezave.

Tabela 4.4

Opis podatkovnih množic BioNLP 2013 GRN. Prikazani so podatki za učno množico (train), množico za nastavljanje parametrov (dev) in testno množico (test).

Ime podatkovne množice	<i>dev</i>	<i>train</i>	<i>test</i>
Število dokumentov	48	86	67
Število členov	1321	2380	1874
Število pravih omenitev	205	422	290
Število razpoznanih omenitev	55	102	86
Število dogodkov	72	157	—
Število povezav	105	254	—
Število <i>Interaction</i> . * povezav	71	159	88 ¹

¹ Število je bilo pridobljeno iz uradnega spletnega servisa BioNLP GRN za evalvacijo.

4.4.3 Rekonstrukcija genskega regulatornega omrežja

Predlagani sistem smo testirali na tekmovalni učni množici GRN BioNLP 2013 s prečnim preverjanjem po metodi izloči-enege in dosegli oceno SER 0.74 z o zamenjavami, 36 odstranitvami, 14 vstavitvami in 31 ujemanji.

Testna podatkovna množica vsebuje 290 omenitev v 67 povedih. Ko ekstrahiramo povezave na testni množici, vedno naučimo modele CRF skupaj na učni množici in množici za nastavljanje parametrov. Rezultati, ki smo jih dosegli na tekmovanju, so prikazani v tabeli 4.5. Glede na oceno SER smo (U. of Ljubljana) [13] dosegli prvo mesto. Ostali sodelujoči so bili še K. U. Leuven [172], TEES-2.1 [173], IRISA-TexMex [174] in EVEX [175]. Vsi sodelujoči smo stremeli k čimmanjšemu številu zamenjav, odstranitvev in vstavitev, medtem ko smo želeli maksimizirati število ujemanj. Dosegli smo najmanjše število zamenjav in dobre rezultate v ostalih treh indikatorjih, s čimer smo skupaj dosegli najboljši rezultat. Visoke številke v številu nerazpoznanih povezav

pomenijo, da so povezave v besedilih zapisane v različnih oblikah in jih je zato težje razpoznavati. Sistem IRISA-TeXMex je dosegel najmanjše število odstranitvev in največje število ujemanj, vendar je po drugi strani izgubljal z večjim številom vstavitev in zamenjav ter se posledično uvrstil slabše.

Tabela 4.5

Uradni rezultati tekmovanja BioNLP 2013 GRN. Tabela prikazuje število zamenjav (S), odstranitvev (D), vstavitev (I), ujemanj (M) in oceno "slot error rate" (SER).

<i>Udeleženec</i>	<i>S</i>	<i>D</i>	<i>I</i>	<i>M</i>	<i>SER</i>
U. of Ljubljana	8	50	6	30	0.73
K. U. Leuven	15	53	5	20	0.83
TEES-2.1	9	59	8	20	0.86
IRISA-TeXMex	27	25	28	36	0.91
EVEX	10	67	4	11	0.92

Od oddaje rezultatov na tekmovanju smo predstavili dodatne atributske funkcije in dodatno procesiranje podatkov. Slednje ekstrahira povezave med omenitvami genov *B. Subtilis* iz hierarhičnih povezav – med geni v dogodkih. O izboljšanih rezultatih poročamo v tabeli 4.6. Vsi rezultati so pridobljeni z novimi atributskimi funkcijami in so grupirani glede na vključenost novega procesiranja (vii) in čiščenja podatkov. Brez obeh procesiranj že dodatno izboljšamo naš rezultat iz tekmovanja za 1% z oceno SER 0.72. Nove atributske funkcije torej ekstrahirajo več povezav s povečano natančnostjo. Zanimivo je, da vključenost dodatnega procesiranja pokvari rezultat za 4%, ker razpozna nekaj nepravilnih povezav, vendar kljub temu odkrije dodatna ujemanja. Najboljši rezultat je torej dosežen s čiščenjem podatkov in brez dodatnega procesiranja (vii). Glede na oddan rezultat pa smo ga torej še dodatno izboljšali za 5% SER, kar je že značilna razlika.

Na sliki 4.7 prikazujemo napovedano gensko regulatorno omrežje s povezavami, ki smo jih ekstrahirali. V primerjavi z oddanim omrežjem na tekmovanju [13], smo izboljšali sistem in odkrili dodatni povezavi (tj. *spoIVFB* → *Inhibition* → *spoIVFA*, *sigE* → *Transcription* → *gerE*) ter izbrisali eno povezavo (tj. *sigB* → *Transcription* → *yvyD*). V primeru, da je izbrisana povezava pravilna, bi lahko oba rezultata združili in dosegli oceno 0.67 SER s 4 zamenjavami, 50 odstranitvami, 5 vstavitvami in 34 ujemanji od skupno 88 povezav v testni množici. Čeprav je to, v kolikor vemo, trenutno najboljši

Tabela 4.7

Ekstrahirane povezave glede na vrsto procesiranja na podatkovni množici za nastavljanje parametrov (dev) in testni množici (test). Rezultati čiščenja podatkov predstavljajo število zaničnih povezav in število podvojenih povezav (ločeno s poševnico). Ocene SER so kumulativne.

<i>Procesiranje</i>	<i>Dev</i>		<i>Test</i>	
	#	SER	#	SER
Ekstrakcija dogodkov (iii)	29	1.00	32	1.00
Ekstrakcija povezav med omenitvami (iv)	44	0.87	12	1.00
Ekstrakcija povezav med dogodki (v)	11	0.84	2	1.00
Ekstrakcija povezav med geni (vi)	14	0.84	5	0.97
Ekstrakcija povezav med geni v dogodkih (vii) ¹	26	0.73	15	0.96
Ekstrakcija povezav s pravili (viii)	12	0.75	53	0.76
Čiščenje podatkov (ix)	22/20	0.75	14/5	0.73

¹ Ekstrakcija povezav med geni v dogodkih (vii) daje boljše rezultate na množici za nastavljanje parametrov kot na testni množici, zato je ne uporabimo za pridobitev končnega rezultata.

V fazi ekstrakcije dogodkov (iii) odkrijemo dogodke, ki jih predstavimo v obliki povezav. Dogodki niso upoštevani pri evalvaciji, zato njihove ekstrakcije ne vplivajo neposredno na rezultat SER. Ekstrahirani dogodki predstavljajo le vhod za kasnejši procesiranje (v) in (vii).

Prvi dve procesiranji povezav (slika 4.5) že dosežeta obetajoče rezultate na podatkovni množici za nastavljanje parametrov – množica dev, medtem ko na testni množici ekstrahirata sedem pravilnih in sedem nepravilnih povezav, zato vrednost SER ostaja enaka 1.0. Naslednji dve procesiranji ekstrahirata več pravilnih povezav na testni množici in še dodatno izboljšata rezultate na množici dev. Ekstrakcija povezav med dogodki občutno izboljša rezultate na množici dev, vendar je izboljšanje na testni le malenkostno. Najnižja končna ocena SER je dosežena, če tega procesiranja ne uporabljamo. Naslednja ekstrakcija povezav z uporabo pravil ne izboljša rezultatov na množici dev, čeprav odkrije pomembno število povezav pred vključitvijo ekstrakcije povezav med dogodki. Obe ekstrakciji odkrijeta veliko enakih povezav. Na testni množici se izkaže ravno nasprotno, saj pravila odkrijejo bistveno več povezav kot predhodno procesiranje.

4.4.5 Uspešnost ekstrakcije glede na atributske funkcije

Izbira najbolj informativnih atributskih funkcij je v strojnem učenju ključnega pomena za povečevanje kvalitete rezultatov. V tabeli 4.8 prikazujemo rezultate ekstrakcij z uporabo različnih naborov atributskih funkcij. Atributske funkcije smo razdelili od bolj splošnih (A-C) do bolj specifičnih (D-H). Kot pričakovano, ugotovimo, da se rezultati z več uporabljenimi atributskimi funkcijami izboljšujejo. V primeru, da uporabimo le najbolj osnovne funkcije (A), sistem razpozna le eno povezavo, ki je napačna in je posledično ocena SER večja od 1.0. Z uporabo atributskih funkcij, ki so povezane z organizmom *B. Subtilis* (C), ne dosežemo nobenega izboljšanja. Ko dodamo še dodatne funkcije (D), pa značilno izboljšamo rezultate za 0.12 v oceni napake. Hkrati predvidevamo, da so za izboljšanje odgovorne tudi predhodne atributske funkcije, ki skupaj z dodanimi tvorijo odvisnosti ali pa je to le rezultat tega, ker so generatorske funkcije in tvorijo veliko različnih atributskih funkcij v primerjavi s predhodnimi (A-C). Tudi skupina (E) še dodatno izboljša rezultat. To je pričakovano, saj sklepamo, da v množici za nastavljanje parametrov in v učni množici obstajajo povezave istega tipa. V primeru, da uporabimo le skupine atributskih funkcij (D) in (E), dosežemo oceni SER 0.98 in 0.87, kar pomeni, da so za uspešnost pomembne tudi predhodne funkcije. Če pa množici (D) in (E) uporabimo skupaj, dosežemo oceno SER 0.81. Prav tako se zdi, da podmnožica (H) ne prispeva k rezultatu. To ne drži, saj če odstranimo podmnožico (G), bomo dosegli oceno SER 0.74. Torej, kot pričakovano ugotovimo, da je pomembno izbrati čimvečji nabor atributskih funkcij, čeprav na prvi pogled izgleda da njihove posamezne podmnožice ne vplivajo na rezultat.

4.5 Sklep

V pričujočem poglavju smo predstavili sistem z večkratnimi in zaporednimi procesiranjmi za namene ekstrakcije povezav iz besedil. Sistem pri svojem delovanju uporablja le linearno-verižne modele CRF in domensko specifična pravila. Podobno kot v prejšnjem poglavju za odkrivanje koreferenčnosti, tudi tu na inovativen način predstavimo povezave z zaporedji omenitev in prav tako omogočimo razpoznavanje povezav med atributi na daljših razdaljah z zaporedji izpuščenih omenitev. S predlaganim pristopom smo dosegli tudi najboljši rezultat na tekmovanju BioNLP 2013 za namene odkrivanja genskih regulatornih mrež. V tem poglavju smo opisali tudi izboljšave predlaganega pristopa, ki še dodatno izboljšajo rezultat. V prihodnjem delu bomo poskusili prilag-

Tabela 4.8

Ekstrahirane povezave z različnimi nabori atributskih funkcij. Kratice pomenijo število zamenjav (S), odstranitvev (D), vstavitev (I), ujemanj (M) in oceno "slot error rate" (SER). Rezultati so evalvirani na podatkovni množici za nastavljanje parametrov in z uporabo le procesiranja, ki uporabljajo modele CRF.

<i>Nabor atributskih funkcij</i>	<i>Dev</i>				
	<i>S</i>	<i>D</i>	<i>I</i>	<i>M</i>	<i>SER</i>
A	0	67	1	0	1.01
A – B	1	64	2	2	1.00
A – C	1	64	2	2	1.00
A – D	0	52	7	15	0.88
A – E	0	41	12	26	0.79
A – F	1	38	12	28	0.76
A – G	0	37	12	30	0.73
A – H	0	37	12	30	0.73

Nabori atributskih funkcij so izbrani sledeče: (A) porazdelitev ciljnih oznak, velika začetnica, dvojna velika začetnica, Hearst-ova pravila, razdalja členov, (B) globina odvisnosti, starševska odvisnost, pot odvisnosti, (C) BSubtilis, Je BSubtilis, Je BSubtilisP par, (D) predpona, pripona, (E) zaporedna vrednost, trenutna vrednost, (F) kontekst, (G) levo / desno / vmes, in (H) razbitje. Natančen opis atributskih funkcij se nahaja v tabeli 4.1 in v tabeli 4.2.

goditi postopek ekstrakcije povezav z linearno-verižnimi modeli CRF in zaporedji izpuščenih omenitev za uporabo na drugih domenah, da pokažemo splošno uporabnost predlagane metode.



*Prepoznavanje imenskih
entitet*

5.1 O prepoznavanju imenskih entitet

Pri prepoznavanju imenskih entitet želimo posameznim členom (tj. besedam) v besedilu določiti njihov pomen ali tip. To je ena izmed najstarejših in najbolj raziskanih nalog ekstrakcije informacij in prav tako tudi najbolj rešena. Prvi sistemi so se osredotočali na splošne tipe podatkov, kot so imena oseb ali organizacij [21]. V zadnjem času se veliko raziskav na področju procesiranja naravnega jezika usmerja na biomedicinsko področje, kjer hranijo ogromno količino znanstvene literature, v kateri so opisana različna odkritja in odvisnosti. Avtomatsko pridobivanje in obdelava teh podatkov omogoča dostop do novih odkritij ali pohitritev pri izvajanju poskusov. Že samo osnovna kategorizacija besedil je v začetku omogočila odkrivanje zakonitosti med geni in boleznimi [176–179]. Trenutno znani postopki omogočajo natančnejšo ekstrakcijo podatkov in združevanje virov, kar se uporablja za nove načine poizvedovanja, organizacije, analiziranja in vizualizacije podatkov.

V tem poglavju predstavimo modularen sistem za razpoznavanje kemijskih spojin in zdravil [14], s katerim smo sodelovali na tekmovanju CHEMDNER 2013 in dosegli sedmo mesto med 27 sodelujočimi ekipami. Tekmovalci smo morali odkriti več razredov kemijskih snovi – *systematic*, *identifiers*, *formula*, *trivial*, *abbreviation*, *family* in *multiple* – in nato ekstrakcije glede na lastno verjetno pravilnosti tudi rangirati.

5.1.1 Umestitev v sorodno delo

Večina raziskovalcev na področju razpoznavanja imenskih entitet se ukvarja z identifikacijo osebnih imen, lokacij, organizacij, datumov, dogodkov, ipd. Tradicionalno se pristopi delijo na slovarske, morfološke in kontekstualne [180]. V predlagani metodi preko atributskih funkcij posredno uporabimo vse tri pristope, ko pridobivamo dodatno znanje iz vhodnih podatkov. Poleg tega uporabimo tudi dodatne podatkovne baze in atributske funkcije, ki uporabljajo slovarske vire.

V literaturi obstaja nekaj pristopov, ki kombinirajo pogojna naključna polja (CRF) [90] in metodo podpornih vektorjev (SVM) [181] za razpoznavanje imenskih entitet. Linearno-verižni modeli CRF se uporabljajo za označevanje zaporedij besed, medtem ko modeli SVM bolje klasificirajo posamezne primere besed v vnaprej določene razrede ali v zvezno vrednost. Nekateri raziskovalci so pokazali, da se za namene razpoznavanja entitet modeli CRF izkažejo bolje kot SVM-ji [182]. Slednje še posebej drži, ker za probleme obdelave besedil navadno generiramo veliko število značilk, ki jih znajo

modeli CRF zelo dobro upoštevati. Skupna uporaba modelov SVM in CRF je bila predlagana tudi na drugih področjih [183], kjer raziskovalci najprej uporabijo model SVM in nato še CRF. V primerjavi s tem pristopom mi uporabimo oba modela v obratnem vrstnem redu, kjer model SVM služi kot meta-klasifikator med različnimi tipi ekstrakcij. Najbolj podoben pristop so predlagali Cai in sod. [184], ki najprej z modelom CRF odkrijejo ločnice med potencialnimi omenitvami entitet v besedilu in nato uporabijo model SVM, da napovejo končne ekstrakcije. V našem pristopu najprej z več atributskimi funkcijami uporabimo več modelov CRF in nato z modelom SVM združimo ocene, da se odločimo o končni vrednosti.

Področje ekstrahiranja informacij se je hitreje začelo razvijati v 1990-tih letih s serijo konferenc Message Understanding Conference (MUC) od 1987 (MUC-1) do 1997 (MUC-7), od katerih so se predvsem začetne zelo usmerile v prepoznavanje imenskih entitet. Kasneje so se oblikovale iniciative in tekmovanja, kot so Automatic Content Extraction (ACE) [38], Semantic Evaluation (SemEval) [39] in Conference on Natural Language Learning (CoNLL) Shared Tasks [40], v okviru katerih so pripravili več označenih podatkov. V zadnjem času so začeli pristope prilagajati tudi za uporabo na biomedicinskih domeni in zato organizirali evalvacije v okviru konferenc BioCreative I in II [41], ki sta se osredotočali na prepoznavanje genov in proteinov v znanstvenih člankih. Poleg tega so se na področju pojavila tudi druga tekmovanja – LLL [42] in BioNLP [28], ki so poskušala izboljšati še ostale naloge ekstrakcije informacij.

5.1.2 Tekmovanje CHEMDNER 2013

Tekmovanje v razpoznavanju kemijskih spojin in zdravil – Chemical compound and drug name recognition (CHEMDNER) challenge – sledi seriji tekmovanj BioCreative, ki so bila orientirana v razpoznavanje genov in proteinov. Tekmovanje CHEMDNER 2013 je osredotočeno na domeno kemijskih sestavin, saj na tem področju še ni bilo razvitih uporabnih sistemov za razpoznavanje. Cilj tekmovanja je (a) predstaviti zmožnosti avtomatske ekstrakcije informacij, (b) izboljšati trenutno najboljše sisteme za ekstrakcijo, (c) testirati predlagane sisteme na neodvisni podatkovni množici, (d) prilagoditi obstoječe sisteme za ekstrakcijo na izbrani domeni, (e) primerjati med različnimi strategijami predlaganih sistemov, (f) izboljšati skalabilnost sistemov in (g) raziskati možnosti integracije z obstoječimi sistemi ter različnimi domenami. Posebej za tekmovanje je bila pripravljena nova podatkovna množica, ki vsebuje 10.000 povzetkov iz PubMed člankov s skupno 84.355 identificiranimi omenitvami. Tekmo-

vanje je bilo razdeljeno v nalogi razpoznavanja omenitev kemijskih spojin – chemical entity mentions (CEM) – in rangiranja razpoznanih omenitev – chemical document indexing (CDI). Pri nalogi CEM morajo sodelujoči z začetnim in končnim znakom v besedilu določiti omenitve kemijskih snovi in pri nalogi CDI morajo razpoznane omenitve rangirati od najbolj do najmanj verjetnih. Prepoznanih omenitev ni potrebno klasificirati v posamezne tipe kemijskih snovi in tekmovalci lahko uporabijo katerekoli dodatne vire, ki jih želijo (angl. open challenge setting). Kljub zelo velikemu številu prvotnih prijav na tekmovanje, je rezultate oddalo le 27 tekmovalnih ekip s skupno 87 sodelujočimi raziskovalci.

Večina sodelujočih tekmovalcev je uporabila modele CRF od enega do četrtega reda z zelo različnimi metodami razčlenjevanja in z bogatim naborom lingvističnih, ortografskih, morfoloških ali slovarskih atributskih funkcij. Nekaj sistemov je celo temeljilo na ročno definiranih pravilih za ekstrakcijo in identifikaciji okrajšav. Med dodatnimi viri jih je največ uporabilo znane podatkovne baze CTD, PubChem, ChEBI, Jochem, periodni sistem elementov in korpus Brown. Zmagovalci tekmovanja [185] so uporabili dva modela CRF, enega na posameznih členih in drugega na črkah. Poleg tega so si pomagali še s korpusom Brown in komponento za združevanje ocen.

5.2 *Metoda za razpoznavanje imenskih entitet*

V tem razdelku definiramo predstavitev podatkov, ki jo uporabimo pri razpoznavanju kemijskih snovi z linearno-verižnimi modeli CRF, vključno z uporabljenimi atributskimi funkcijami. Nato prikažemo delovanje predlagane metode na izbranem primeru skozi ves postopek priprave podatkov, posameznih ekstraktov, združevanja rezultatov in meta-klasifikacijo.

5.2.1 *Predstavitve podatkov*

Pri razpoznavanju imenskih entitet moramo za vsak člen določiti njegov tip. Členi so rezultat členjenja in predstavljajo besede, ločila in posebne znake. Tipi členov so odvisni od zaporedja besed in njihovih pomenov v besedilu, zato se odločimo za uporabo modela CRF. Za vsak člen pridobimo tudi dodatne informacije, kot so oblikoslovne oznake, leme ali drevesa odvisnosti, ki jih uporabimo v atributskih funkcijah.

Atributske funkcije

Izbira ustreznih atributskih funkcij je ključnega pomena za uspešnost napovedovanja s katerimkoli algoritmom iz strojnega učenja. Definiramo jih kot predloge in nato končne atributske funkcije pridobimo s sprehodom čez podatkovno množico. V tabelah 5.1 in 5.2 predstavimo atributske funkcije, ki jih uporabimo v naši metodi.

Tabela 5.1

Opis atributskih funkcij. Vse funkcije so uporabljene v obeh faktorjih modelov CRE. Implementacije so dostopne v javnem repozitoriju programske kode [20] (v razredu *FeatureFunctionPackages*).

Naziv	Opis
Porazdelitev ciljnih oznak	Verjetnost ciljne oznake.
Velika začetnica	Ali se trenutna omenitev začne z veliko začetnico.
Dvojna velika začetnica	Ali se obe zaporedni omenitvi začneta z veliko začetnico.
Oblikoslovna oznaka	Oblikoslovna oznaka prvega člena v omenitvi.
Števk	Člen vsebuje dve števki.
Veliki začetnici	Ali člen vsebuje dve veliki začetnici.
Pomišljaj	Ali člen vsebuje pomišljaj.
Pika	Ali člen vsebuje piko.
Vejica	Ali člen vsebuje vejico.
Grški znak	Člen vsebuje grški znak.
Kemijski element	Člen ustreza elementu v periodnem sistemu elementov.
Blagovna znamka [186]	Člen predstavlja blagovno znamko podjetja.
IUPAC [186]	Člen ustreza IUPAC imenu zdravila.
Imena [186]	Člen predstavlja splošno ime zdravila.
Sopomenke [186]	Člen ustreza sinonimu zdravila.
CTD [187]	Člen ustreza kemijski spojini ali njeni sopomenki v CTD podatkovni bazi.

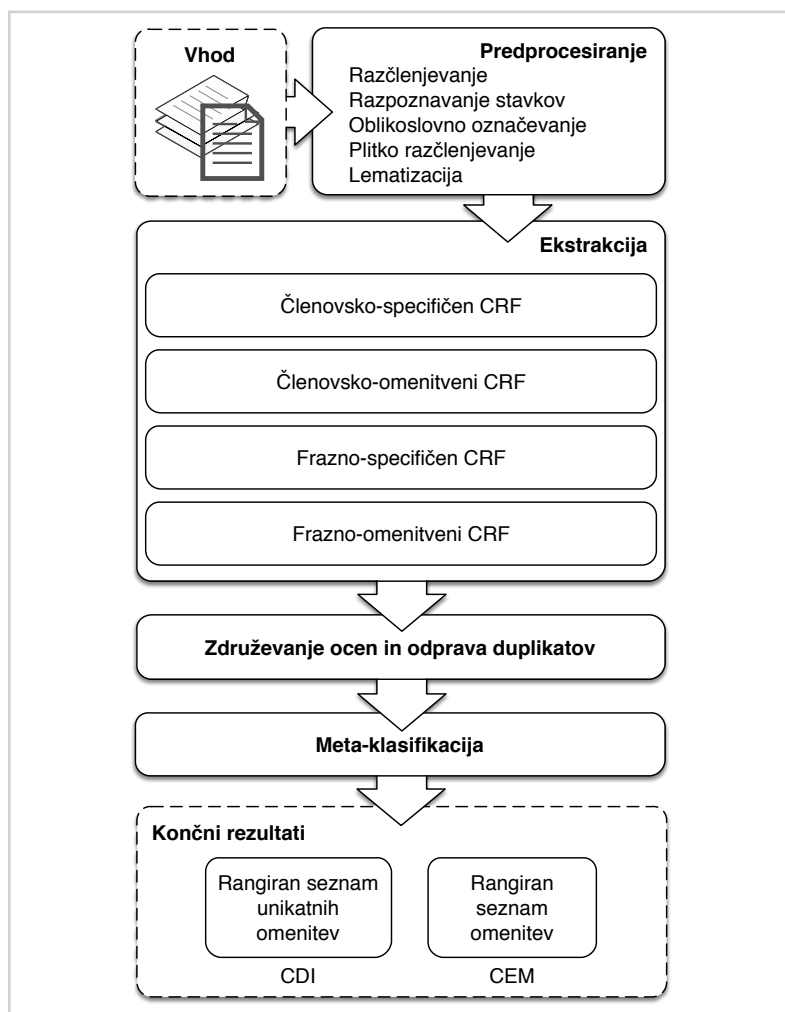
Tabela 5.2

Opis generatorjev atributskih funkcij. Vse funkcije so uporabljene v obeh faktorjih modelov CRF (razen predpona in pripona, ki sta samo tipa unigram). Implementacije so dostopne v javnem repozitoriju programske kode [20] (v razredu *FeatureFunction-Packages*).

Naziv	Opis
Predpona	Predpona dolžine 2 in 3 na razdaljah od -5 do 5.
Pripona	Pripona dolžine 2 in 3 na razdaljah od -5 do 5.
Zaporedna vrednost	Kombinacija dveh oznak (npr. <i>B-NP/I-NP</i>) zaporednih členov na razdalji od -1 do 1. Uporabljene oznake so beseda, oblikoslovna oznaka, lema in oznaka plitkega razčlenjevanje.
Trenutna vrednost	Vrednost oznake na razdalji od -1 do 1. Uporabljene oznake so beseda, oblikoslovna oznaka, lema in oznaka plitkega razčlenjevanja.
Predhodna / naslednja vrednost	Kombinacija oznake predhodnega in naslednjega člena na razdaljah od -2 do 2. Uporabljene so oblikoslovne oznake.
Vrednosti levo / desno	Vrednosti oznak levo ali desno od člena na razdalji od 1 do 3 členov. Uporabljene oznake so besede in oblikoslovne oznake.
N-gram	N-gram vrednosti trenutnega člena dolžine od 2 do 4 znakov.

5.2.2 Opis predlaganega sistema

V tem razdelku predstavimo sistem za razpoznavanje kemijskih snovi z uporabo več modelov CRF za ekstrakcijo (slika 5.1), ki so med seboj neodvisni. Sistem sestoji iz štirih komponent. Prva predstavi podatke v ustrezni obliki in izvede predprocesiranje, ki dodatno obogati podatke. Glavno komponento tvori več modelov za razpoznavanje imenskih entitet, čemur sledi združevanje rezultatov in odprava duplikatov. Nazadnje komponenta z metaklasifikatorjem upošteva vse rezultate in vrne seznam končnih odločitev. Celotna implementacija predlaganega sistema je na voljo v javnem programskem repozitoriju [20], ki uporablja knjižnici CRFSuite [135] in Weka [188].



Slika 5.1

Arhitektura predlaganega sistema za razpoznavanje kemijskih spojin in zdravil. Sistem sestoji iz štirih komponent, kjer prva uvozi in obogati podatke, nato štirje modeli ekstrahirajo podatke, ki so v predzadnjem koraku združeni in nazadnje potrjeni z meta-klasifikatorjem. Vhod predstavljajo besedilni dokumenti in kot izhod je vrnjen rezultat v formatu CHEMDNER 2013.

Vhod

Vhodni podatki sestojijo iz nestrukturiranih besedil, iz katerih želimo razpoznati kemijske spojine in zdravila (naloga CEM). Razpoznane entitete nato še rangiramo glede

na verjetnost pravilnosti ekstrakcij (naloga CDI).

Dokumenti v okviru tekmovanja CHEMDNER 2013 predstavljajo naslove in povzetke člankov iz bibliografske baze PubMed. Primer vhodne povedi je sledeč (tj. poved iz PubMed-ovega članka 23444833):

“The lystabactins are composed of serine (Ser), asparagine (Asn), two formylated/hydroxylated ornithines (FOHOrn), dihydroxy benzoic acid (Dhb), and a very unusual nonproteinogenic amino acid, 4,8-diamino-3-hydroxyoctanoic acid (LySta).”

Predprocesiranje

V fazi predprocesiranja pretvorimo vhodne podatke v ustrezno obliko in jih obogatimo z dodatnimi oznakami. Najprej razpoznamo povedi in izvedemo razčlenjevanje [31] z ročno definiranimi pravili. Poleg tega še dodatno razbijemo člene, ki sestojijo iz mešanih simbolov – števk in črk (npr. *100mg* v *100 mg*, *Thr172* v *Thr 172*), kemijskih spojin (npr. *CuBL* v *Cu BL*) ali členov, ki vsebujejo grške znake (npr. μ *LIF* v μ *LIF*). Razčlenjena zaporedja nato še obogatimo z oblikoslovnimi oznakami, oznakami plitkega razčlenjevanja [33] in lemmami [170].

Ekstrakcija

Komponenta za ekstrakcijo razpozna kemijske spojine v besedilu. Štirje različni modeli CRF uporabljajo dva različna tipa členitev in dva tipa označevanja. Za vsak označen člen vrnejo verjetnost oznake in verjetnost celotnega označenega zaporedja v povedi.

Glede na tip členjenja definiramo naslednja modela:

Členovski. Model v zaporedju predstavi vse člene – besede in simbole, kot so bili razpoznani v fazi predprocesiranja.

Frazni. Pred oddajo rezultatov za tekmovanje CHEMDNER 2013 smo namesto členov v zaporedju uporabili samostalniške besedne zveze, ki smo jih pridobili iz razčlenjevalnika odvisnosti. Ker razpoznavne niso pripomogle k izboljšanju končnega rezultata [14], smo kasneje uporabili le plitek razčlenjevalnik [17] in prav tako izbrali le samostalniške besedne zveze. V obeh primerih zaporedje vsebuje le samostalniške besede ali besedne zveze iz posameznih povedi.

Cilj tekmovanja CHEMDNER 2013 je bil identificirati le dele besedila, ki predstavljajo kemijske spojine. Tekmovalcem torej ni bilo treba eksplicitno označiti razpoznavnih delov s specifično ciljno oznako (npr. *FORMULA* ali *FAMILY*), zato tudi iz tega stališča zasnujemo dva tipa modelov:

Specifičen. Model vsakemu členu v zaporedju priredi specifično oznako kemijske spojine ali oznako *O* v nasprotnem primeru.

Omenitveni. Model člene v zaporedju označuje z oznakama *M* in *O*, pri čemer *M* pomeni, da člen predstavlja kemijsko spojino ali zdravilo in *O* pomeni nasprotno.

Pri označevanju ne uporabljamo notacije “znotraj-zunaj-začetek” (angl. inside-outside-beginning notation – IOB), saj je bilo pokazano, da takšen tip označevanja ne izboljša rezultatov.

S kombinacijo obeh zgornjih možnosti sestavimo štiri različne pristope za ekstrakcijo, ki uporabljajo linearno-verižne modele CRF. Vključno s primerom so to sledeči:

Členovsko-specifičen: The/*O* lystabactins/*FAMILY* are/*O* composed/*O* of/*O* serine/*TRIVIAL* (/ *O* Ser/*FORMULA*)/*O* ,/*O* asparagine/*TRIVIAL* ...

Členovsko-omenitveni: The/*O* lystabactins/*M* are/*O* composed/*O* of/*O* serine/*M* (/ *O* Ser/*M*)/*O* ,/*O* asparagine/*M* ...

Frazno-specifičen: The lystabactins/*FAMILY* serine/*TRIVIAL* Ser/*FORMULA* asparagine/*TRIVIAL* Asn/*FORMULA* two/*O* formylated/hydroxylated ornithines/*MULTIPLE* ...

Frazno-omenitveni: The lystabactins/*M* serine/*M* Ser/*M* asparagine/*M* Asn/*M* two/*O* formylated/hydroxylated ornithines/*M* ...

Združevanje ocen in odprava duplikatov

Rezultat komponente za ekstrakcijo so deli besedil, ki jih je kot kemijsko spojino prepoznal vsaj eden od modelov. Pri združevanju ocen in odpravi duplikatov identificiramo ekstrakcije posameznih modelov in pripravimo podatke za meta-klasifikacijo. Ocene razpoznavnih imen združimo, če se dve imeni prekrivata za vsaj za en znak in v tem primeru kot rezultat upoštevamo unijo znakov. V primeru, da eden od modelov ekstrahira več imen za izbrano omenitev (tj. možno pri členovskih modelih), kot verjetnost imena izberemo člen z maksimalno verjetnostjo oznake.

Tabela 5.3

Združene ocene ekstrakcij. Dvojice predstavljajo verjetnost oznake in verjetnost celotnega zaporedja, ki je rezultat posameznega modela. Uporabljeni modeli po vrsti so (a) členovsko-specifičen, (b) členovsko-omenitveni, (c) frazno-specifičen in (d) frazno-omenitveni.

Besedilo	Verjetnost oznak in zaporedij	Razred
“lystabactins”	(0.40, 0.20)(0.38, 0.55)(0.50, 0.35)(0.60, 0.51)	yes
“serine”	(0.45, 0.24)(0.37, 0.48)(0.53, 0.38)(0.63, 0.58)	yes
...	...	...
“nonproteinogenic”	(0.20, 0.30)(0.22, 0.37)(0.00, 0.00)(0.00, 0.00)	no
...	...	...

V tabeli 5.3 prikazujemo združene ekstrakcije iz zgornjega primera. Vsak zapis predstavlja en učni podatek, ki ga meta-klasifikator uporabi za končno napoved.

Meta-klasifikacija

V meta-klasifikaciji uporabljamo algoritem SVM, ki poda končno odločitev ali določena ekstrakcija predstavlja kemijsko spojino oziroma zdravilo ali ne. Model SVM je primeren za reševanje regresijskih in klasifikacijskih problemov. Izbrali smo ga, ker naš problem ustreza standardnemu binarnemu klasifikacijskem problemu in ker modeli SVM empirično dosegajo dobre rezultate na različnih področjih (npr. razpoznavanje slik, bioinformatika ali procesiranje besedil).

Klasifikacijski primer iz prejšnjega koraka sestoji iz rezultatov štirih modelov CRE, od katerih hranimo verjetnost oznake in verjetnost celotnega napovedanega zaporedja za vsako potencialno razpoznano ime. Torej, model kot vhod obravnava osem zveznih atributov (tj. verjetnosti med nič in ena) in delčki besedil, ki so razpoznani kot pozitivni, so vključeni v končen rezultat.

Končen rezultat

Tekmovanje CHEMDNER 2013 je sestavljeno iz dveh nalog – CEM in CDI, ki sta predstavljena v različnih formatih. Za nalogo CDI vrnemo seznam unikatnih ekstrakcij v dokumentu, ki so rangirane glede na največjo verjetnost kateregakoli zaporedja. Za nalogo CEM vrnemo indeks prvega in končnega znaka ekstrakcije in jih rangiramo na enak način kot pri nalogi CDI.

Tabela 5.4

Struktura končnih rezultatov za nalogi CEM in CDI v okviru tekmovanja CHEMDNER 2013. Stolpci predstavljajo ID dokumenta v bazi PubMed, besedilo ekstrahirane kemijske spojine ali zdravila ter verjetnost pravilnosti, ki jo je ocenila predlagana metoda. Pri nalogi CEM so rangi in verjetnosti nepomembni, saj se ocenjuje le pravilnost ekstrakcij. Poleg tega se z znakom *T* ali *A* določi, ali smo ime ekstrahirali iz naslova ali povzetka vhodnega dokumenta.

<i>CEM results</i>			
Dokument	Besedilo	—	—
23444833	A:318:324	1	0.5
23444833	A:289:301	2	0.5
23444833	A:448:464	3	0.5
...			
<i>CDI results</i>			
Dokument	Besedilo	Rang	Verjetnost
23444833	serine	1	0.63
23444833	lystabactins	2	0.60
23444833	nonproteinogenic	3	0.22
...			

Primer končnih rezultatov prikazujemo v tabeli 5.4.

5.3 Analiza rezultatov

Tekmovanje za razpoznavo kemijskih spojin in zdravil (CHEMDNER 2013) predvideva, da sodelujoči identificirajo njihova imena, in rangirajo ekstrakcije glede na lastno oceno verjetnosti za pravilnost ekstrakcije.

Naš pristop temelji na modularni zgradbi, ki vsebuje več modelov CRF za ekstrakcijo in meta-klasifikator, ki določi končen rezultat. V nadaljevanju najprej predstavimo splošne rezultate, analiziramo ekstrakcije glede na posamezen model in nazadnje pregledamo uspešnost ekstrakcij glede na uporabljene množice atributskih funkcij.

5.3.1 Ocenjevalni kriterij

Uradni ocenjevalni kriterij tekmovanja temelji na uspešnosti sistema v prepoznavanju kemijskih spojin in zdravil, vendar ne ocenjuje ekstrakcije specifičnih tipov imen.

Uspešnost razpoznavanja imenskih entitet je evalvirana s pomočjo ocene F_1

$$F_1 = \frac{2 \cdot \text{natančnost} \cdot \text{priklic}}{\text{natančnost} + \text{priklic}},$$

ki je harmonična sredina med natančnostjo in priklicem. Natančnost meri, kakšen delež ekstrakcij je pravilen, medtem ko priklic pomeni, kakšen delež ekstrakcij je razpoznan glede na celotno bazo. Mikro-povprečena (angl. micro-averaged) ocena F_1 je izračunana kot kumulativno število ekstrakcij čez vse dokumente in makro-povprečena (angl. macro-averaged) ocena F_1 izračuna povprečje ocen med vsemi vhodnimi dokumenti. Prva ocen je uporabna, ko želimo vedeti, kako se sistem obnaša v splošnem. Drugo pa navadno uporabimo, če velikost dokumentov zelo variira.

Sodelujoče skupine so bile rangirane glede na mikro-povprečeno oceno F_1 na večkratnih vzorcih testne množice, velikosti 1000 dokumentov.

5.3.2 Podatkovna množica CHEMDNER 2013

Podatkovna množica CHEMDNER 2013 sestoji iz naslovov in povzetkov člankov iz baze PubMed, ki imajo označene kemijske spojine in zdravila. Oznake vključujejo sistematična imena, skupna ali generična imena spojin in zdravil, marketinška imena, identifikatorje, kratice, referenčne številke spojin in formul. Članki so bili izbrani tako, da predstavljajo reprezentativno podatkovno množico iz biomedicinskega področja.

Tabela 5.5 prikazuje osnovne statistike učne, testne in podatkovne množice za nastavljanje parametrov (dev). Vsak vhodni dokument je opremljen z identifikatorjem v bazi PubMed, z naslovom in s povzetkom.

5.3.3 Uspešnost razpoznavanja zdravil in kemijskih spojin

Predlagan pristop smo testirali na množici za nastavljanje parametrov (dev) z uporabo modelov, ki so bili naučeni na učni množici. Dosegli smo 0.83 ocene F_1 za nalogo CEM in 0.84 ocene F_1 za nalogo CDI (tabela 5.9). Glede na rezultate, ki smo jih poročali ob oddaji tekmovalnih rezultatov, smo izboljšali sistem za 10.2 in 9.2 odstotkov [14]. Izboljšave smo dosegli zaradi dodanih atributskih funkcij, podatkovnih virov in dodatnega označevanja v fazi predprocesiranja.

Tekmovalna testna podatkovna množica je vsebovala 20,000 povzetkov člankov iz baze PubMed, od katerih jih je 3,000 označenih in 2,478 jih vsebuje vsaj eno omenitev kemijske spojine ali zdravila. Slednji dokumenti (pred oddajo rezultatov tekmovalcem

Tabela 5.5

Opis učne (train), testne (test) in podatkovne množice za nastavljanje parametrov (dev) na tekmovanju CHEMDNER 2013. Vrednosti *Povzetki s CEM* pomenijo, koliko povzetkov je vsebovalo vsaj eno kemijsko oznako.

	<i>train</i>	<i>dev</i>	<i>test</i> ¹
<i>Število povzetkov</i>	3,500	3,500	3,000
<i>Število povzetkov s CEM</i>	2,916	2,907	2,478
<i>Število revij</i>	193	188	188
<i>Število spojin</i>	8,520	8,677	7,563
<i>Število omenitev</i>	29,478	29,526	25,351

¹ Pred oddajo rezultatov so tekmovalci dobili testno množico z 20.000 povzetki iz baze PubMed, od katerih je bilo le izbranih 3.000 upoštevanih pri evalvaciji. S tem so organizatorji tekmovanja onemogočili goljufanje, da bi kdo ročno pregledoval testno množico.

neznani) so bili uporabljeni za rangiranje sodelujočih skupin in so bili po tekmovanju razkriti. Za namene razpoznavanja označenih imenskih entitet smo modele naučili skupaj na množiti dev in učni množici ter dosegli rezultate, ki so prikazani v tabeli 5.6. Glede na mikro-povprečno oceno F_1 smo dosegli sedmo mesto [189] od skupaj 27 sodelujočih ekip, ki so oddale rezultate. Zmagovalna ekipa [185] je dosegla oceno F_1 0.87 (CEM) in 0.88 (CDI), kar je približno devet odstotkov več od nas. V primerjavi z našim sistemom so pri ekstrakciji uporabili dva modela CRF in združili njune rezultate. Prvi model CRF so (tako kot mi) naučili na posameznih členih in dodatno uporabili korpus *Brown*. Njihov drugi model pa predstavlja novost, saj so ga naučili na črkah besed od zadaj naprej.

Po oddaji tekmovalnih rezultatov smo dodali nove atributske funkcije, plitko razčlenjevanje in uporabili BioLemmatizer [170] v fazi predprocesiranja ter redefinirali frazna modela CRF. Izboljšani rezultati so zapisani v tabelah 5.7 in 5.8 ter so urejeni glede na podmnožice atributskih funkcij in tip modela za ekstrakcijo. Najboljši novi rezultati izboljšajo predhodne rezultate za približno 7% na 0.83 ocene F_1 (CEM) in 0.84 ocene F_1 (CDI). Slednji pristop in pristop, ki smo ga uporabili na tekmovanju sta značilno različna, kar smo preverili s primerjavo modelov, zgrajenih na poljubnih učnih podmnožicah v velikosti 1,000 povzetkov. Oba rezultata sta pridobljena s kombinacijo

Tabela 5.6

Uradni rezultati tekmovanja CHEMDNER 2013. Tabela prikazuje število ekstrahiranih kemijskih spojin ali zdravil (Pred.), natančnost (P), priklje (R) in oceno F_1 . Modeli so bili naučeni na učni množici in množici dev ter evalvirani na testni množici. Uporabili smo členovsko-specifičen (ČS) in členovsko-omenitveni (ČM) model ter kombinacijo obeh (ČSM).

Model	Pred.	Mikro-povprečje		
		P	R	F
<i>Rezultati CDI</i>				
ČS (run1)	12381	0.83	0.75	0.79
ČM (run2)	12783	0.81	0.76	0.78
ČSM (run3)	13172	0.80	0.77	0.79
<i>Rezultati CEM</i>				
ČS (run1)	20438	0.87	0.70	0.77
ČM (run2)	21109	0.85	0.71	0.77
ČSM (run3)	21562	0.85	0.72	0.78

modelov CRF, kar malenkostno izboljša rezultate v primerjavi z uporabo posameznih. Frazni modeli se odrežejo najslabše, saj dosežejo najnižje rezultate, zato jih v združenih rezultatih ne uporabimo.

5.3.4 Analiza rezultatov glede na model učenja

V tabeli 5.9 prikazujemo rezultate ekstrakcij po posameznih modelih. Vsak model razpozna kemijske spojine ali zdravila neodvisno od drugih in v primeru, ko je uporabljenih več modelov, so rezultati združeni. Iz rezultatov opazimo, da združena modela dosežeta najboljše rezultate.

Členovsko-specifičen in členovsko-omenitveni model v primerjavi z njunim združenim modelom dosežeta slabšo oceno F, medtem ko združen model izboljša njen rezultat iz 0.73 na 0.79 F_1 za nalogo CEM in iz 0.75 na 0.81 F_1 za nalogo CDI. Tudi primerjavi z modeli, ki smo jih oddali v okviru tekmovanja, in z modeli z novimi atributskimi funkcijami, doseže združen model najboljše rezultate v vseh ocenah. Modeli, ki so členovskega tipa, dosežejo najboljše natančnosti, saj se pri njih bolj upoštevajo sosednje besede, ki kontekstno vplivajo na razpoznavo kemijskih spojin ali zdravil.

Vključitev dodatnih atributskih funkcij, ki smo jih predstavili po tekmovanju, izboljša rezultate na 0.83 ocene F_1 za nalogo CEM in na 0.84 ocene F_1 za nalogo CDI. V tem primeru modele naučimo na učni podatkovni množici in jo testiramo na množici

za nastavljanje parametrov (dev) (tabela 5.10), kjer dosežemo malenkost boljše rezultate kot na testni množici. Pri evalvaciji na testni množici modele naučimo na obeh preostalih množicah in dosežemo že dovolj dobre rezultate le na majhnem naboru atributskih funkcij – podmnožica atributskih funkcij A v tabelah 5.7 in 5.8. Iz slednjega bi lahko sklepali, da se učna množica razlikuje v primerjavi z učno množico in množico dev, čeprav naj bi bili članki izbrani popolnoma naključno.

5.3.5 Uspešnost razpoznavanja glede na atributske funkcije

Izbira najbolj informativnih atributskih funkcije je ključna za doseganje boljših rezultatov z metodami iz strojnega učenja. V tabelah 5.7 in 5.8 prikazujemo rezultate ekstrakcij na testni množici z uporabo različnih podmnožic atributskih funkcij. Podmnožice razdelimo v skupine od bolj splošnih (A) do bolj specifičnih (E). V nasprotju z našimi pričakovanji pa nismo dosegli izboljšav pri uporabi več atributskih funkcij, saj sistem že pri uporabi podmnožice A odkrije največ kemijskih spojin ali zdravil. Ko k množici A dodamo še uporabo leksikonov in posebnih znakov, rezultati ostanejo enaki. To je delno tudi posledica tega, ker veliko kemijskih spojin vsebuje posebne znake in zato v fazi členjenja razpadejo na več členov in zato posamezen člen pogosto ni najden v slovarju. Ko dodajamo še ostale podmnožice atributskih funkcij, se uspešnost sistema zniža na nivo rezultatov, ki smo jih oddali v okviru tekmovanja. Rezultati torej kažejo, da lahko najboljše rezultate dosežemo že z zelo spošnimi ortografskimi in morfološkimi atributskimi funkcijami.

5.4 Sklep

V pričujočem poglavju smo predstavili modularen sistem za razpoznavanje kemijskih spojin in zdravil iz besedil. Sistem sestoji iz štirih različnih modulov, ki za ekstrakcijo uporabljajo linearno-verižne modele CRF, in meta-klasifikatorja, ki združi njihove ocene ter vrne končen rezultat. Z opisanim pristopom smo sodelovali na tekmovanju CHEMDNER 2013 in dosegli sedmo mesto med 27 sodelujočimi ekipami. Po tekmovanju smo sistem še dodatno izboljšali in analizirali posamezne komponente ter različne nabore atributskih funkcij, na podlagi česar smo uspeli sistem izboljšati za dodatne 4% ocene F. Ker predlagan pristop ne vsebuje domensko specifičnih pravil, ga lahko naučimo za delovanje nad poljubno domeno.

Tabela 5.7

Prepoznavanje kemijskih spojin in zdravil z različnimi podmnožicami atributskih funkcij – CEM. Tabela prikazuje število pravih razpoznav (TP), napačnih razpoznav (FP), zgrešenih razpoznav (FN), natančnost (P), priklíc (R) in oceno F_1 . Uporabili smo členovsko-specifičen (ČS) in členovsko-omenitveni (ČM) model, kombinacijo slednjih (ČSM) ter frazno-specifičen (FS) in frazno-omenitveni (FM) model. Podmnožice atributskih funkcij so sledeče: (A) porazdelitev ciljnih oznak, velika začetnica, dvojna velika začetnica, predpona, pripona, (B) grški znak, kemijski element, blagovna znamka, IUPAC, imena, sopomenke, CTD, (C) zaporedna vrednost, trenutna vrednost, predhodna / naslednja vrednost, (D) številki, veliki začetnici, pomišljaj, pika, vejica, ngram in (E) vrednosti levo / desno. Njihov opis se nahaja v tabelah 4.1 in 4.2.

	Makro-povprečje						Mikro-povprečje		
	TP	FP	FN	P	R	F	P	R	F
A									
ČS	19142	2092	6209	0.88	0.77	0.81	0.90	0.76	0.82
ČM	19596	2382	5755	0.87	0.79	0.81	0.89	0.77	0.83
ČSM	19827	2488	5524	0.87	0.79	0.81	0.89	0.78	0.83
FS	14466	3710	10885	0.76	0.60	0.66	0.80	0.57	0.66
FM	14655	4271	10696	0.75	0.61	0.65	0.77	0.58	0.66
A – B									
ČS	19142	2092	6209	0.88	0.77	0.81	0.90	0.76	0.82
ČM	19596	2382	5755	0.87	0.79	0.81	0.89	0.77	0.83
ČSM	19827	2488	5524	0.87	0.79	0.81	0.89	0.78	0.83
FS	14466	3710	10885	0.76	0.60	0.66	0.80	0.57	0.66
FM	14655	4271	10696	0.75	0.61	0.65	0.77	0.58	0.66
A – C									
ČS	19018	2091	6333	0.88	0.77	0.80	0.90	0.75	0.82
ČM	19459	2443	5892	0.87	0.78	0.81	0.89	0.77	0.82
ČSM	19685	2558	5666	0.87	0.79	0.81	0.89	0.78	0.83
FS	14357	3739	10994	0.76	0.60	0.65	0.79	0.57	0.66
FM	14508	4288	10843	0.74	0.61	0.65	0.77	0.57	0.66
A – D									
ČS	18449	2318	6902	0.87	0.75	0.78	0.89	0.73	0.80
ČM	18803	2817	6548	0.85	0.76	0.78	0.87	0.74	0.80
ČSM	19131	2925	6220	0.85	0.77	0.79	0.87	0.75	0.81
FS	13873	3926	11478	0.74	0.58	0.63	0.78	0.55	0.64
FM	14000	4539	11351	0.73	0.58	0.63	0.76	0.55	0.64
A – E									
ČS	17264	2428	8087	0.85	0.70	0.74	0.88	0.68	0.77
ČM	17856	3013	7495	0.83	0.72	0.75	0.86	0.70	0.77
ČSM	18147	3094	7204	0.82	0.72	0.75	0.85	0.72	0.78
FS	12839	4006	12512	0.72	0.53	0.59	0.76	0.51	0.61
FM	13069	4761	12282	0.70	0.54	0.59	0.73	0.52	0.61

Tabela 5.8

Prepoznavanje kemijskih spojin in zdravil z različnimi podmnožicami atributskih funkcij – CDI. Tabela prikazuje število pravih razpoznav (TP), napačnih razpoznav (FP), zgrešenih razpoznav (FN), natančnost (P), priklic (R) in oceno F_1 . Uporabili smo členovsko-specifičen (ČS) in členovsko-omenitveni (ČM) model, kombinacijo slednjih (ČSM) ter frazno-specifičen (FS) in frazno-omenitveni (FO) model. Podmnožice atributskih funkcij so sledeče: (A) porazdelitev ciljnih oznak, velika začetnica, dvojna velika začetnica, predpona, pripona, (B) grški znak, kemijski element, blagovna znamka, IUPAC, imena, sopomenke, CTD, (C) zaporedna vrednost, trenutna vrednost, predhodna / naslednja vrednost, (D) številki, veliki začetnici, pomišljaj, pika, vejica, ngram in (E) vrednosti levo / desno. Njihov opis se nahaja v tabelah 4.1 in 4.2.

	<i>TP</i>	<i>FP</i>	<i>FN</i>	<i>Makro-povprečje</i>			<i>Mikro-povprečje</i>		
				<i>P</i>	<i>R</i>	<i>F</i>	<i>P</i>	<i>R</i>	<i>F</i>
<i>A</i>									
ČS	11088	1540	2602	0.87	0.83	0.84	0.88	0.81	0.84
ČM	11306	1791	2384	0.86	0.84	0.84	0.86	0.83	0.84
ČSM	11430	1914	2260	0.86	0.84	0.84	0.86	0.83	0.85
FS	6959	7174	6731	0.52	0.56	0.52	0.49	0.51	0.50
FM	7052	7755	6638	0.51	0.56	0.51	0.48	0.52	0.49
<i>A – B</i>									
ČS	11088	1540	2602	0.87	0.83	0.84	0.88	0.81	0.84
ČM	11306	1791	2384	0.86	0.84	0.84	0.86	0.83	0.84
ČSM	11430	1914	2260	0.86	0.84	0.84	0.86	0.83	0.85
FS	6959	7174	6731	0.52	0.56	0.52	0.49	0.51	0.50
FM	7052	7755	6638	0.51	0.56	0.51	0.48	0.52	0.49
<i>A – C</i>									
ČS	11061	1536	2629	0.87	0.82	0.84	0.88	0.81	0.84
ČM	11260	1833	2430	0.86	0.84	0.84	0.86	0.82	0.84
ČSM	11394	1951	2296	0.85	0.84	0.83	0.85	0.83	0.84
FS	6948	7143	6742	0.52	0.55	0.52	0.49	0.51	0.50
FM	7026	7717	6664	0.51	0.56	0.51	0.48	0.51	0.49
<i>A – D</i>									
ČS	10770	1715	2920	0.86	0.80	0.82	0.86	0.79	0.82
ČM	10926	2123	2764	0.83	0.82	0.81	0.84	0.80	0.82
ČSM	11109	2240	2581	0.83	0.82	0.81	0.83	0.81	0.82
FS	6737	7186	6953	0.51	0.54	0.50	0.48	0.49	0.49
FM	6804	7810	6886	0.50	0.54	0.50	0.47	0.50	0.48
<i>A – E</i>									
ČS	10284	1835	3406	0.83	0.76	0.78	0.85	0.75	0.80
ČM	10565	2305	3125	0.81	0.78	0.78	0.82	0.77	0.80
ČSM	10715	2400	2975	0.80	0.79	0.78	0.82	0.78	0.80
FS	6333	7007	7357	0.49	0.50	0.47	0.47	0.46	0.47
FM	6457	7774	7233	0.48	0.51	0.47	0.45	0.47	0.46

Tabela 5.9

Prepoznavanje kemijskih spojin in zdravil z različnimi podmnožicami atributskih funkcij na množici dev. Tabela prikazuje število pravih razpoznav (TP), napačnih razpoznav (FP), zgrešenih razpoznav (FN), natančnost (P), priklic (R) in oceno F_1 . Uporabili smo členovsko-specifičen (ČS) in členovsko-omenitveni (ČM) model, kombinacijo slednjih (ČSM) ter frazno-specifičen (FS) in frazno-omenitveni (FO) model.

Model	TP	FP	FN	Makro-povprečje			Mikro-povprečje		
				P	R	F	P	R	F
Rezultati CEM									
Tekmovalne atributske funkcije									
ČS	17945	4113	11581	0.78	0.63	0.67	0.81	0.61	0.70
ČM	18695	4878	10831	0.76	0.65	0.68	0.79	0.63	0.70
ČSM	21865	3860	7661	0.83	0.76	0.77	0.85	0.74	0.79
FS	15940	4884	13586	0.74	0.58	0.63	0.77	0.54	0.63
FM	16098	5670	13428	0.71	0.59	0.62	0.74	0.55	0.63
Dodatne atributske funkcije									
ČS	19380	2222	5971	0.88	0.78	0.81	0.90	0.76	0.83
ČM	19655	2497	5696	0.87	0.79	0.81	0.89	0.78	0.83
ČSM	19970	2636	5381	0.86	0.80	0.81	0.88	0.79	0.83
FS	14729	3779	10622	0.77	0.61	0.66	0.80	0.58	0.67
FM	14700	4315	10651	0.74	0.61	0.65	0.77	0.58	0.66
Rezultati CDI									
Tekmovalne atributske funkcije									
ČS	10642	3050	5453	0.77	0.69	0.71	0.78	0.66	0.71
ČM	10965	3702	5130	0.75	0.71	0.71	0.75	0.68	0.71
ČSM	12804	2871	3291	0.82	0.81	0.80	0.82	0.80	0.81
FS	7824	8345	8271	0.51	0.53	0.50	0.48	0.49	0.49
FM	7876	9116	8219	0.49	0.54	0.49	0.46	0.49	0.48
Dodatne atributske funkcije									
ČS	11110	1605	2580	0.87	0.83	0.84	0.87	0.81	0.84
ČM	11247	1835	2443	0.86	0.84	0.84	0.86	0.82	0.84
ČSM	11415	2003	2275	0.85	0.84	0.83	0.85	0.83	0.84
FS	7043	7270	6647	0.53	0.56	0.52	0.49	0.51	0.50
FSM	7026	7807	6664	0.51	0.56	0.51	0.47	0.51	0.49

Tabela 5.10

Prepoznavanje kemijskih spojin in zdravil z različnimi podmnožicami atributskih funkcij na množici dev z modeli, kot smo jih oddali v okviru tekmovanja. Tabela prikazuje število pravih razpoznav (TP), napačnih razpoznav (FP), zgrešenih razpoznav (FN), natančnost (P), priklíc (R) in oceno F_1 . Uporabili smo členovsko-specifičen (ČS), členovsko-omenitveni (ČM) model in kombinacijo slednjih (ČSM).

Model	TP	FP	FN	Makro-povprečje			Mikro-povprečje		
				P	R	F	P	R	F
Rezultati CEM									
ČS (run1)	18858	3848	10668	0.80	0.66	0.70	0.83	0.64	0.72
ČM (run2)	19477	4619	10049	0.78	0.68	0.70	0.81	0.66	0.73
ČSM (run3)	19790	4839	9736	0.77	0.68	0.70	0.80	0.67	0.73
Rezultati CDI									
ČS (run1)	11324	2884	4771	0.78	0.72	0.73	0.80	0.70	0.75
ČM (run2)	11582	3515	4513	0.76	0.74	0.73	0.77	0.72	0.74
ČSM (run3)	11835	3765	4260	0.75	0.74	0.73	0.76	0.74	0.75



*Iterativna ekstrakcija
informacij s pomočjo
ontologije*

6

6.1 O ekstrakciji informacij

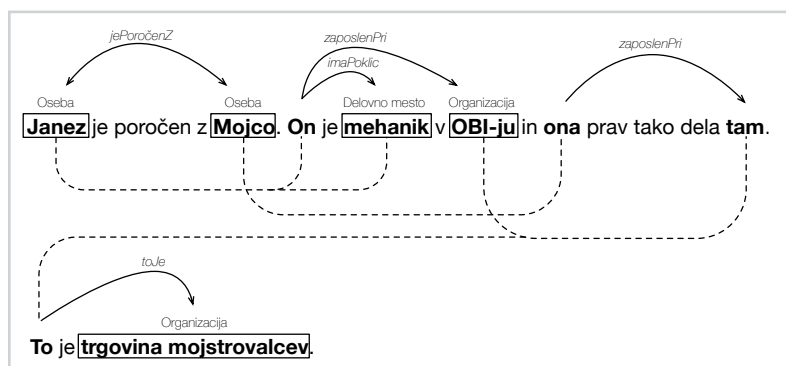
Ekstrakcija informacij se je kot raziskovalno področje pojavilo v 1970-tih letih, ko so se prvotni sistemi osredotočali predvsem na razpoznavanje imenskih entitet iz besedil [25]. Od tedaj dalje so se začeli počasi razvijati tudi ostali postopki, med katerimi so glavni razpoznavanje imenskih entitet, ekstrakcija povezav in odkrivanje koreferenčnosti [21]. Veliko dela je bilo narejenega tudi na področju semantičnih tehnologij, kjer so bili predlagani sistemi za avtomatsko gradnjo ontologij ali pa so izkoriščali različne semantične vire za bolj natančno ekstrakcijo [26]. Čeprav se termin ekstrakcija informacij večinoma povezuje le z obdelavo besedil, v splošnem pomeni pridobivanje podatkov iz poljubnega tipa vira, ki je lahko na primer zvok, video ali slika.

Poleg glavnih nalog ekstrakcije imajo zelo pomembno vlogo tudi tehnike predprocesiranja, s katerimi dodatno obogatimo vhodne podatke in s tem omogočimo bolj natančno ekstrakcijo. Na sliki 6.1 prikazujemo glavne naloge ekstrakcije informacij, ki so (a) razpoznavanje imenskih entitet, (b) ekstrakcija povezav in (c) odkrivanje koreferenčnosti. Pri (a) želimo vsak člen v besedilu označiti s pomenom, kjer nas zanima, katere besede predstavljajo osebe, organizacije, lokacije ipd. Pri (b) želimo med omenitvami (označene odebeljeno) odkriti povezave in pri (c) želimo omenitve združiti tako, da vsaka gruča omenitev predstavlja eno entiteto. Ker večina dosedanjih pristopov združuje naloge v zaporedju ali se osredotoča le na eno izmed nalog, v tem poglavju predlagamo iterativni in skupni sistem za ekstrakcijo informacij, ki združuje vse tri glavne naloge ekstrakcije. Pri tem smo inovativno v vseh primerih uporabili le linearno-verižne modele CRF in upoštevali vmesne rezultate ekstrakcij, ki so pripomogli k izboljšavi skupnih končnih rezultatov.

6.1.1 Umestitev v sorodno delo

Sistemi za ekstrakcijo informacij so se do sedaj osredotočali na posamezen problem ekstrakcije ali na njihovo zaporedno vezavo [21]. Trenutno najboljši objavljeni rezultati kažejo, da je najbolje rešen problem razpoznavanja imenskih entitet, kjer sistemi na splošnih podatkovnih množicah dosegajo okoli 90% ocene F_1 , medtem ko je odkrivanje koreferenčnosti rešeno slabše z ocenami F_1 okoli 70% in ekstrakcija povezav najslabše z okoli 50%.

Razpoznavanje imenskih entitet (poglavje 5) je eno izmed prvih raziskanih področjih v okviru ekstrakcije informacij. Predlagani pristopi se v grobem delijo na pristope



Slika 6.1

Ekstrakcija informacij. Predstavitev rezultata glavnih nalog ekstrakcije informacij – (a) odkrivanje koreferenčnosti (tj. črtkana črta, ki povezuje omenitve na isto entiteto), (b) ekstrakcija povezav (npr. jePoročenZ, zaposlenPri) in (c) razpoznavanje imenskih entitet (npr. oseba, organizacija)

s pravili in pristope z uporabo metod strojnega učenja. Prvi so razpoznavali entitete z uporabo leksikonov, slovarjev ali s pomočjo vnaprej definiranih pravil [180, 190, 191], medtem ko so ostali uporabili indukcijske metode in različne klasifikacijske modele [93, 192], da so vhodnim členom napovedali ciljne oznake. Pri uporabi klasifikacijskih modelov prednjačijo označevalniki zaporedij, kot so skriti markovski modeli, modeli največje entropije ali pogojna naključna polja [91].

Metode za ekstrakcijo povezav (poglavje 4) se v glavnem delijo v dve kategoriji. Prvo predstavljajo modeli, ki uporabljajo nabor leksikalnih, sintaktičnih in semantičnih atributskih funkcij [161, 193], ter drugo pa predstavljajo jedrne metode [165, 194]. Nekateri pristopi namesto ekstrakcije povezav z atributi prevedejo problem na označevanje zaporedij, kjer je cilj označiti del besedila, ki predstavlja povezavo [63]. Razvile so se tudi tehnike odprte ekstrakcije informacij, ki uporabljajo nenadzorovane pristope in poskušajo pridobiti čimveč podatkov, pri čemer se ne osredotočajo na tip povezave ali atributov [60, 195].

Pri odkrivanju koreferenčnosti (poglavje 3) večina tehnik pretvori nalogo v binarni klasifikacijski problem [73, 112] (tj. algoritem preveri koreferenčnost za vsak par omenitev), kar omogoča uporabo standardnih klasifikatorjev iz strojnega učenja. Raziskovalci so zato predlagali mnogo inovativnih in lingvistično bogatih atributskih funkcij [72, 73] z različnimi tipi modelov, kot so modeli največje entropije [115], metoda podpornih vektorjev [114] ali markovska omrežja [116]. Predlaganih je bilo tudi nekaj uspešnih sistemov, ki temeljijo na pravilih [70, 113], vendar ti zahtevajo dobro predhodno poznavanje besedil in so težje prilagodljivi novim domenam. McCallum

in sod. [100] so za reševanje problema predlagali različne tipe modelov CRF, vendar so le osnovnega izmed njih evalvirali na problemu združevanja citatov, ki je soroden problemu odkrivanja koreferenčnosti [101]. Zaradi zahtevnejšega in dolgotrajnega učenja ter napovedovanja s splošnimi modeli CRF, so raziskovalci razvili model, ki je podoben linearno-verižnemu z dodatkom faktorjev, ki lahko nastopajo na daljših razdaljah [93]. Edini, ki pa so uporabili linearno-verižne modele [112], pa so z njimi odkrivali koreferenčnost med množicami omenitev in ne med posameznimi omenitvami.

Tekom razvoja metod za ekstrakcijo informacij se je razvilo še posebno podpodročje, ki pri ekstrakciji uporablja ali celo polni ontologijo [26]. Ontologije lahko tako predstavljajo dodaten vir podatkov, ki ga lahko uporabimo kot leksikon, seznam pravil ali celo kot mehanizem sklepanja v vseh fazah ekstrakcije. Poleg tega lahko tudi rezultate predstavimo z njeno shemo in jih ob dovolj veliki natančnosti shranimo ter tako sproti bogatimo bazo znanja [78, 196, 197]. Večina novejših sistemov za ekstrakcijo informacij uporablja lastno ontologijo [198], ki pa je lahko povezana še z množico ostalih prosto dostopnih ontologij.

V primerjavi z izvajanjem le enega tipa ekstrakcije ali kombiniranjem več tipov v zaporedju, želimo s skupno ekstrakcijo več nalog povezati med seboj in izkoriščati vmesne rezultate različnih nalog. Kombiniranje več modelov ekstrakcij je bilo najprej preskušeno na področju razpoznavanja imenskih entitet [199, 200], kjer so raziskovalci izboljšali rezultate z upoštevanjem več različnih ekstrakcij. Nato so nekateri predlagali pristope, ki kombinirajo razpoznavanje imenskih entitet in ekstrakcijo povezav z uporabo ontologije in pravil, kjer so prav tako dosegli izboljšave [201, 202]. Roth in Yih [203] sta predlagala induktivni sistem, ki upošteva ročno definirana pravila med posameznimi ekstrakcijami, kar so izboljšali Yao in sod. [204] z avtomatsko pridobitvijo takšnih pravil iz Wikipedije. Edino delo, ki pa je podobno našemu, so predstavili Singh in sod. [205], ki so razvili skupen model CRF za razpoznavanje imenskih entitet, ekstrakcijo povezav in odkrivanje koreferenčnosti. Zaradi kompleksne strukture modela so razvili tudi izboljšano tehniko izmenjave oznak in pokazali, da se rezultati s skupnim modelom malenkostno izboljšajo v primerjavi s samostojnimi modeli. Nekateri raziskovalci so poskusili kombinirati del predprocesiranja s posamezno nalogo ekstrakcije [206] ali pa so nekatere razvite tehnike ekstrakcije informacij aplicirali na druge domene [207, 208] ali na sorodne naloge, kot so segmentacija ali ujemanje citatov [101, 209, 210].

6.2 Predstavitev podatkov

V tem razdelku definiramo predstavitev podatkov, ki omogoča ekstrakcijo informacij za vse tri naloge le z linearno-verižnimi modeli CRF. Pri razpoznavanju imenskih entitet za označevanje uporabljamo le osnovno zaporedje členov, medtem ko pri odkrivanju koreferenčnosti in ekstrakciji povezav preoblikujemo osnovna zaporedja omenitev v zaporedja z izpuščenimi omenitvami.

6.2.1 Pogojna naključna polja

Pogojna naključna polja (angl. conditional random fields – CRF) so zasnovana kot tip modelov, ki se osredotočajo na napovedovanje zaporedij oznak (razdelek 2.4.1). Predstavitev odvisnosti pri podanem problemu oziroma struktura modela nam določa kompleksnost učenja in napovedovanja z modelom ter izbiro ustreznega algoritma za delo z izbranim modelom. Zaradi enostavnosti in hitrosti smo se odločili uporabiti linearno-verižne modele CRF, pri katerih smo omejeni na predstavitev podatkov v obliki zaporedij (slika 2.6 (c)) in z njimi poskušamo rešiti vse tri glavne naloge ekstrakcije informacij. Izbrani modeli poskušajo glede na verjetnostno porazdelitev čimbolj pravilno napovedati verjetnost celotnega zaporedja $P(\bar{y}|\bar{x}, \lambda)$, kjer je $\bar{y}$ zeleno zaporedje glede na vhodno podano zaporedje $\bar{x}$ in predhodno naučene uteži atributskih funkcij λ .

Pri odkrivanju koreferenčnosti z zaporedjem $\bar{x}$ predstavimo zaporedje vseh omenitev v dokumentu in z zaporedjem $\bar{y}$ definiramo, ali je omenitev x_i koreferenčna z omenitvijo x_{i-1} . Ciljne oznake $y_i \in \{C, O\}$ definiramo tako, da oznaka C določa koreferenčnost s predhodno omenitvijo in oznaka O določa nasprotno.

Za ekstrakcijo povezav z zaporedjem $\bar{x}$ predstavimo zaporedje omenitev v povedi, ki lahko predstavljajo attribute povezav, in z zaporedjem $\bar{y}$ označimo tip povezave med omenitvama x_{i-1} in x_i . Ciljne oznake $y_i \in \{MOŽNE\ POVEZAVE\} \cup \{O\}$ pomenijo ime povezave med atributoma ali neobstojeve povezave med njima (tj. oznaka O).

V primeru razpoznavanja imenskih entitet zaporedje $\bar{x}$ predstavlja vse člene v povedi, pri čemer želimo vsak člen označiti s pomensko oznako $y_i \in \{MOŽNA\ IMENA\} \cup \{O\}$. Možna imena so vnaprej znani tipi, kot na primer OSEBA, ORGANIZACIJA ali LOKACIJA. Vsi ostali členi brez znanega tipa pa so označeni z oznako O.

Slika 6.2

Primer atributske funkcije, ki pridobi tipe povezav med dvema konceptoma v ontologiji. Prikazana je SPARQL poizvedba, ki za dva primerka tipov *entity1* in *entity2* pridobi vse možne tipe povezav med njima.

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ie: <http://zitnik.si/ontos/owl/iobie.owl#>
SELECT DISTINCT ?result
WHERE {
    ?entity1 ie:name "%s"^^xsd:string .
    ?entity2 ie:name "%s"^^xsd:string .
    { ?entity1 ?result ?entity2 } UNION
    { ?entity2 ?result ?entity1 } .
    FILTER (STRSTARTS(STR(?result),
        "http://zitnik.si/ontos/owl/iobie.owl#")) .
}
```

Atributske funkcije

Čeprav je bilo v literaturi predlaganih že mnogo atributskih funkcij [67, 71, 72, 123–125], jih razdelimo v kategorije in predlagamo dodatne. Atributske funkcije na področju procesiranja naravnih besedil navadno definiramo kot predloge in nato končne atributske funkcije (enačba (2.4)) pridobimo s prehodom skozi podatkovno množico. Natančna implementacija vseh atributskih funkcij, ki jih uporabljamo, je dostopna v javnem repozitoriju programske kode [20] (v razredu *FeatureFunctionPackages*).

Predlagamo naslednje kategorije atributskih funkcij:

Predprocesiranje. Atributske funkcije te kategorije predstavimo v razdelku 3.2.1.

Lokacija. Atributske funkcije te kategorije predstavimo v razdelku 3.2.1.

Oblika omenitve. Atributske funkcije te kategorije predstavimo v razdelku 3.2.1.

Semantični pomen. Nekaj teh atributskih funkcij predstavimo že v razdelku 3.2.1. Tu dodamo še nove, ki so vezane na uporabo ontologije. Ontologija v predlaganem sistemu predstavlja shemo, množico pravil in vsebuje tudi primerke izbranih konceptov. Atributske funkcije, vezane na ontologijo, so *konceptualni tip v ontologiji*, *starševski konceptualni tip v ontologiji* ali *možni tipi povezav med dvema primerkoma v ontologiji* (primer na sliki 6.2).

Iterativnost. Iterativne atributske funkcije omogočajo skupno ekstrakcijo informacij in upoštevajo oznake modelov, ki so bili naučeni v predhodni iteraciji. Atributske funkcije torej iz vmesnih rezultatov upoštevajo *prepoznane imenske entitete, ekstrahirane povezave med dvema omenitvama, ekstrahirano ime povezave in vrednosti atributov, oznake drugega atributa v razpoznani povezavi in oznake ostalih omenitev v isti koreferenčni gruči.*

6.2.2 Odkrivanje koreferenčnosti

Pri odkrivanju koreferenčnosti želimo združiti koreferenčne omenitve – omenitve, ki predstavljajo isto entiteto. Zaradi uporabe le linearno-verižnih modelov CRF lahko identificiramo le koreferenčne omenitve, ki so si v dokumentu neposredno zaporedne. Torej za rešitev tega problema predlagamo ustrezne transformacije osnovnega zaporedja omenitev, da lahko odkrivamo koreferenčne omenitve na daljših razdaljah. Celoten postopek je podrobneje opisan v poglavju 3.

Osnovno zaporedje omenitev $\bar{x}$ vključuje vse omenitve iz enega dokumenta, ki so urejene glede na mesto pojavitve v njem. V primerih, kjer se omenitve prekrivajo med seboj, jih uredimo v leksikografskem vrstnem redu. Za naš izbrani primer dokumenta (slika 6.1) je osnovno zaporedje omenitev (tj. zaporedje brez izpuščenih omenitev) sestavljeno sledeče: $\bar{x} = [\text{JANEZ}, \text{MOJCO}, \text{ON}, \text{MEHANIK}, \text{OBI-JU}, \text{ONA}, \text{TAM}, \text{To}, \text{TRGOVINA MOJSTROVALCEV}]$.

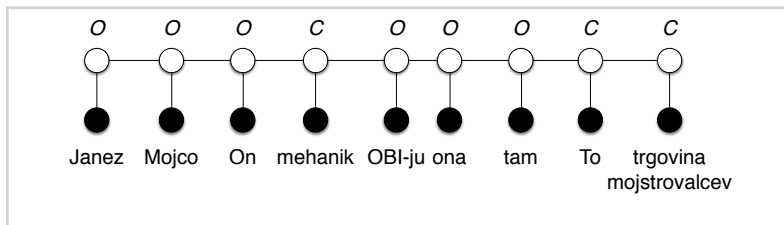
Glede na izbran dokument moramo torej odkriti tri gruče omenitev oziroma entitete: $x_{\text{JANEZ}} = \{\text{JANEZ}, \text{ON}, \text{MEHANIK}\}$, $x_{\text{MOJCA}} = \{\text{MOJCO}, \text{ONA}\}$, $x_{\text{OBI}} = \{\text{OBI-JU}, \text{TAM}, \text{To}, \text{TRGOVINA MOJSTROVALCEV}\}$.

Na sliki 6.3 prikazujemo zaporedje omenitev $\bar{x}$, ki ga lahko uporabimo za učenje verjetnostnih modelov prvega reda. Zaporedje imenujemo *zaporedje z nič izpuščenimi omenitvami*, saj vsebuje vse omenitve iz izbranega dokumenta in med njimi ni (tj. nič) drugih omenitev. Za odkrivanje koreferenčnosti zaporedje označimo za oznaki $\{O, C\}$, kjer C pomeni, da je trenutna omenitev koreferenčna s prejšnjo in O pomeni nasprotno. Ko nad takimi zaporedji naučimo linearno-verižni model CRF, ga lahko potem uporabimo za označevanje zaporedij, za katere vnaprej ne poznamo oznak. Na takšen način lahko iz osnovnega zaporedja omenitev odkrijemo le sledeči koreferenčni gruči omenitev $\{\text{ON}, \text{MEHANIK}\}$ in $\{\text{TAM}, \text{To}, \text{TRGOVINA MOJSTROVALCEV}\}$. Da bi lahko razpoznali še ostale koreferenčnosti (npr. OBI-JU in TAM), moramo uporabiti drug model ali predlagati drugačno predstavitev podatkov. V nadaljevanju se

odločimo za slednje in predstavimo nov način transformacije podatkov.

Slika 6.3

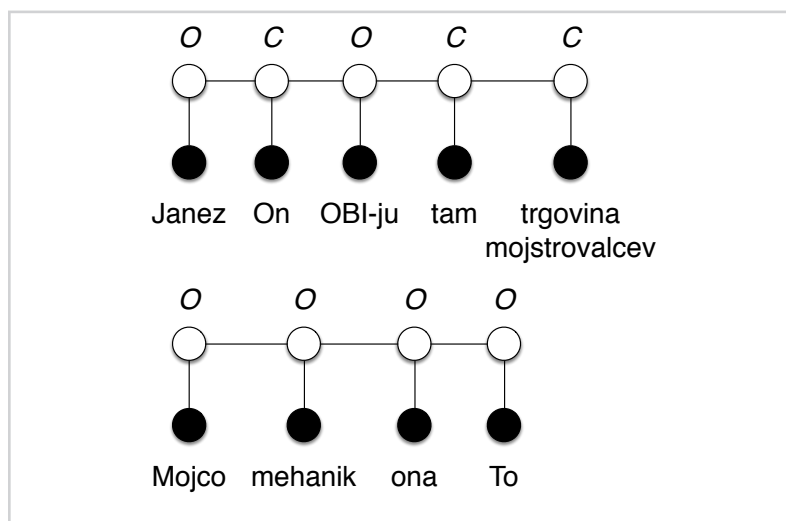
Zaporedje brez izpuščenih omenitev. Osnovno zaporedje omenitev, ki vsebuje vse omenitve iz dokumenta. Če je trenutna omenitev koreferenčna s prejšnjo, je označena s C in sicer z O.



Z dodatnimi transformacijami osnovnega zaporedja omenitev tvorimo več različnih tipov zaporedij in s tem omogočimo razpoznavo vseh treh gruč omenitev iz dokumenta x . Novi tipi zaporedij so zgrajeni iz osnovnega zaporedja $\bar{x}$ in ustrezno označeni z oznakama $\{O, C\}$. Na primer, da se odločimo uporabiti razdalje izpuščenih omenitev od nič do tri, pretvorimo podatke v štiri tipe zaporedij – zaporedja z nič, eno, dvema in s tremi izpuščenimi omenitvami. Za vsak tip zaporedja naučimo ločen model CRE, ki nam omogoča odkrivanje koreferenčnih omenitev na določeni razdalji. Iz zgornjega dokumenta x bi zaporedji z eno izpuščeno omenitvijo izgledali, kot je prikazano na sliki 6.4. S tem bi trenutne rezultate razširili z dvema novima paroma $\{JANEZ, ON\}$ in $\{OBI-JU, TAM\}$. Po uporabi zaporedij z dvema in s tremi izpuščenimi omenitvami pa bi lahko dokončno odkrili še manjkajoči par $\{MOJCO, ONA\}$. Nad pridobljenim rezultatom bi nazadnje izvedli še gručenje in vrnili gruče omenitev x_{JANEZ} , x_{MOJCA} in x_{OBI} .

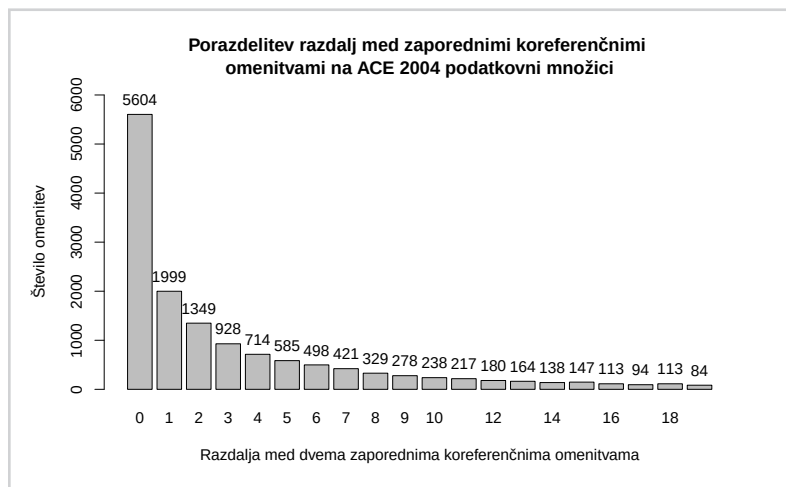
Našo idejo dodatno podpremo s prikazom porazdelitve razdalj med dvema zaporednima koreferenčnima omenitvama na podatkovni množici ACE 2004, kar prikazujemo na sliki 6.5 (množico SemEval2010 podobno analiziramo na sliki 3.1). Podatki so dovolj reprezentativni, da ilustrirajo splošno stanje v ostalih podatkovnih množicah in prav tako v prostih besedilih. Glede na distribucijo je neposredno koreferenčnih le 35% omenitev. Kumulativno do razdalje 20 pa lahko zajamemo 90% vseh koreferenčnih parov. Podobno lahko do razdalje 50 zajamemo okoli 97% parov omenitev. Čeprav lahko z uporabo daljših razdalj teoretično odkrijemo vse možne koreferenčnosti, temu ni tako, saj se pri večjih razdaljah znižuje stopnja natančnosti in povečuje prilik, zato se stopnja uspešnosti ne povečuje. Zaradi takšnih problemov empirično določimo največjo razdaljo 25, do katere izgradimo modele (slika 3.6).

Transformacija v različne tipe zaporedij vrne več zaporedij. Iz osnovnega zaporedja



Slika 6.4

Zaporedji z eno izpuščeno omenitvijo. Zaporedje omenitev vsebuje vsako drugo omenitev iz vhodnega dokumenta. Če je trenutna omenitev koreferenčna s prejšnjo, je označena s C in sicer z O.



Slika 6.5

Porazdelitev razdalj med zaporednimi koreferenčnimi omenitvami na podatkovni množici ACE 2004 [39]. Razdalja x med dvema zaporednima omenitvama pomeni, da med njima obstaja x ostalih omenitev. Razdalje 20 ali več zaradi lepše predstavitve niso prikazane (tj. skupno 1533 omenitev).

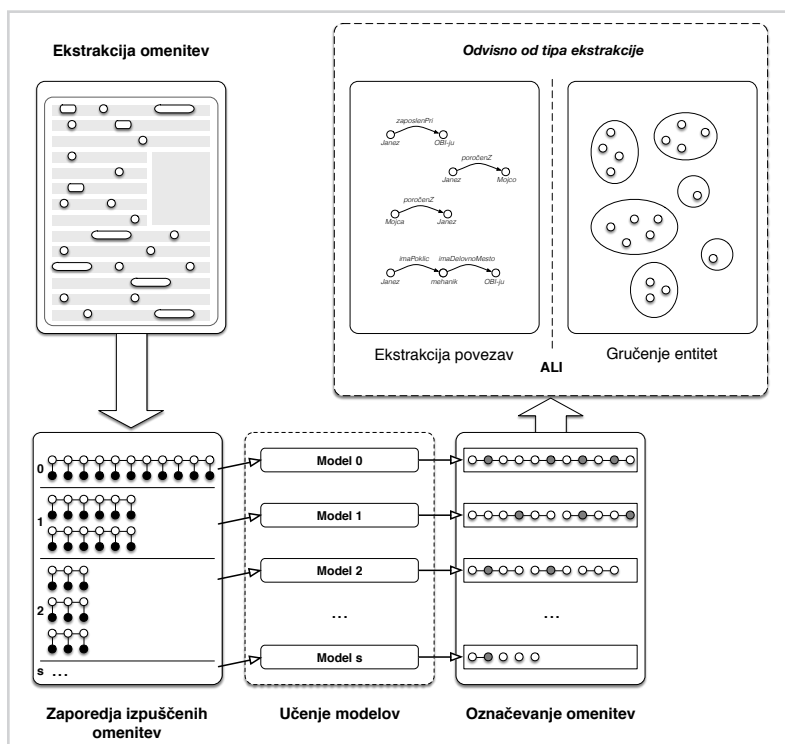
na razdalji ena dobimo dve zaporedji, na razdalji dve tri, na razdalji tri štiri ipd. Število povezav za vsak tip zaporedij se ohranja, poleg tega pa lahko modele gradimo in z njimi

napovedujemo vzporedno. Bolj natančna časovna analiza je opisana v razdelku 3.2.4.

Enak tip transformacij podatkov in postopek za gradnjo modelov CRF uporabljamo pri odkrivanju koreferenčnosti in ekstrakciji povezav. Glavna razlika je v različnih oznakah in predstavitvi označenih podatkov. Za ekstrakcijo povezav (podrobneje opisana v naslednjem razdelku) odkrivamo povezave med sosednjimi omenitvami v zaporedju, medtem ko pri koreferenčnosti označimo, če obe zaporedni omenitvi predstavljata isto entiteto. Poleg tega moramo pri odkrivanju koreferenčnosti po koncu iz označenih rezultatov sestaviti gruče omenitev, ki predstavljajo posamezne razpoznanne entitete. Visokonivojsko predstavitev toka podatkov obeh tipov ekstrakcije predstavimo na sliki 6.6.

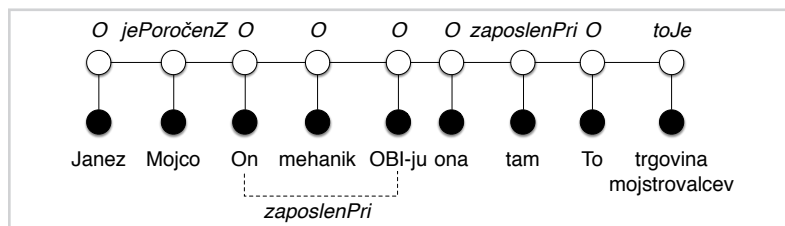
Slika 6.6

Tok podatkov pri ekstrakciji povezav in odkrivanju koreferenčnosti z linearno-verižnimi modeli CRF. Najprej transformiramo osnovno zaporedje omenitev v zaporedja z izpuščenimi omenitvami. Na vsakem novem tipu zaporedij naučimo svoj linearno-verižni model CRF in z njim označimo neznane dokumente. Pri ekstrakciji povezav povezave preberemo iz pridobljenih oznak in jih vrnemo kot rezultat, medtem ko pri odkrivanju koreferenčnosti izvedemo še gručenje omenitev in kot rezultat vrnemo seznam gruč, ki predstavljajo entitete.



6.2.3 Ekstrakcija povezav

Pri ekstrakciji povezav želimo odkriti, če med dvema omenitvama obstaja povezava. Na sliki 6.7 prikazujemo osnovno zaporedje omenitev $\bar{x}$ iz vhodnega dokumenta (slika 6.1). Zaradi uporabe linearno-verižnih modelov CRF ne moremo odkriti povezav med omenitvami, ki ni niso neposredno zaporedne (npr. *zaposlenPri*). Podobno kot pri odkrivanju koreferenčnosti predlagamo dodatne transformacije osnovnega zaporedja, da omogočimo ekstrakcijo povezav med atributi na daljših razdaljah. Celoten postopek je podrobneje opisan v poglavju 4.



Slika 6.7

Zaporedje brez izpuščenih omenitev za ekstrakcijo povezav. Zaporedje vsebuje vse omenitve iz vhodnega dokumenta (slika 6.1).

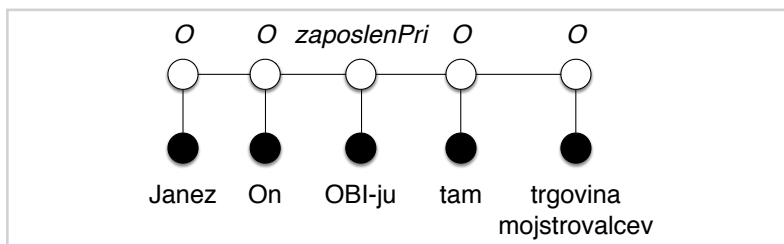
Zaporedje oznak $\bar{y}$ označimo s simbolom O , če med trenutno in prejšnjo omenitvijo nismo odkrili povezave. V nasprotnem primeru trenutno omenitev označimo z imenom povezave (npr. *zaposlenPri* ali *poročenZ*), ki jo tvori s prejšnjo omenitvijo.

Osnovno zaporedje omenitev na sliki 6.7 ne more modelirati povezav med oddaljenimi omenitvami. Na primer med omenitvama ON in OBI-JU obstaja povezava *zaposlenPri*, ki je ne moremo razpoznati z linearno-verižnim modelom CRF. Zaradi identificiranega problema tudi tu transformiramo osnovno zaporedje v ustrezna zaporedja z izpuščenimi omenitvami. Na sliki 6.8 zato prikazujemo primer zaporedja z eno izpuščeno omenitvijo, ki omogoči, da razpoznamo povezavo, ki je pred tem nismo mogli.

slika 6.9 prikazuje porazdelitev razdalj med atributi povezav (tj. osebkom in predmetom) na podatkovni množici ACE 2004. Iz podatkov opazimo, da večina povezav obstaja na razdalji dveh omenitev, kar še dodatno podkrepi potrebo po transformaciji osnovnega zaporedja omenitev. V nasprotnem primeru bi lahko v najboljšem primeru odkrili le povezave, katerih atributi so neposredno zaporedni (tj. 2126 povezav).

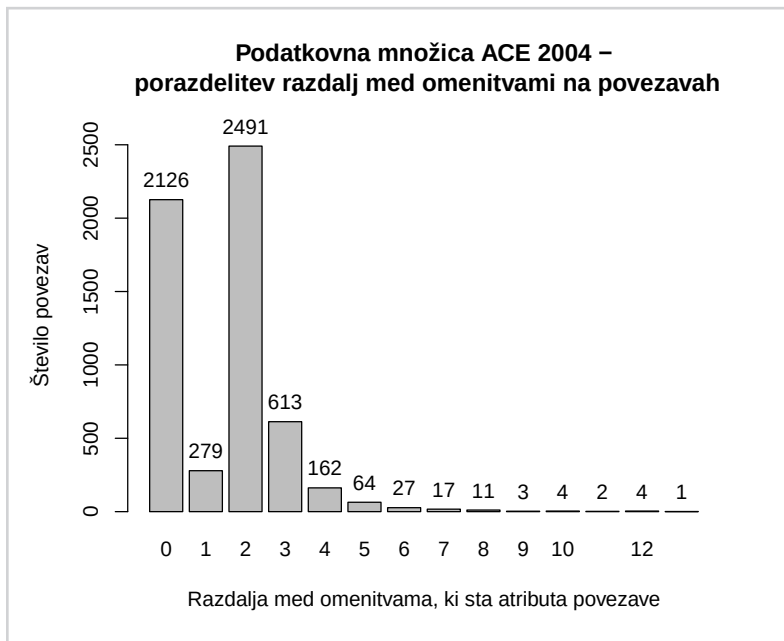
Slika 6.8

Zaporedje z eno izpuščeno omenitvijo za ekstrakcijo povezav. Zaporedje je zgenerirano iz osnovnega (slika 6.7). Drugo možno zaporedje z eno izpuščeno omenitvijo sestoji iz omenitev [MOJCO, MEHANIČ, ONA, To].



Slika 6.9

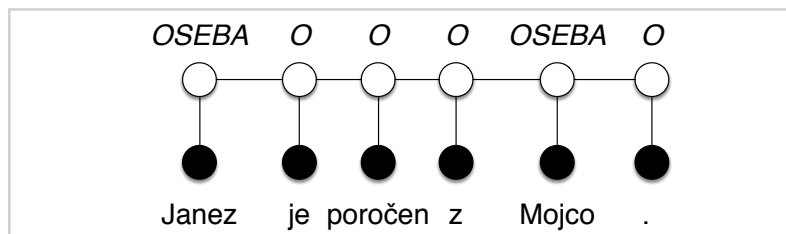
Porazdelitev razdalj med atributi povezav (tj. osebkom in predmetom) na podatkovni množici ACE 2004. Razdalja x pomeni, da je med atributom x ostalih omenitev.



6.2.4 Prepoznavanje imenskih entitet

Razpoznavanje imenskih entitet je ena izmed najstarejših nalog ekstrakcije informacij, ki želi za vsak člen v besedilu določiti tip njegove imenske entitete. Z vhodnim zaporedjem $\bar{x}$ predstavimo zaporedje členov v izbrani povedi. Vsakega izmed členov želimo označiti z ustrežno oznako imenske entitete (npr. *Oseba* ali *Organizacija*) ali z oznako

O, če zanj nimamo ustrezne predstavitve. Na sliki 6.10 prikazujemo primer označevanja prve povedi iz prejšnjega dokumenta (slika 6.1). Pri označevanju uporabljamo tradicionalen pristop z linearno-verižnimi modeli CRF, ki jih vključimo v iterativen pristop skupaj z ekstrakcijo povezav in odkrivanjem koreferenčnosti.



Slika 6.10

Zaporedje členov za razpoznavanje imenskih entitet.

6.3 Sistem za iterativno ekstrakcijo informacij

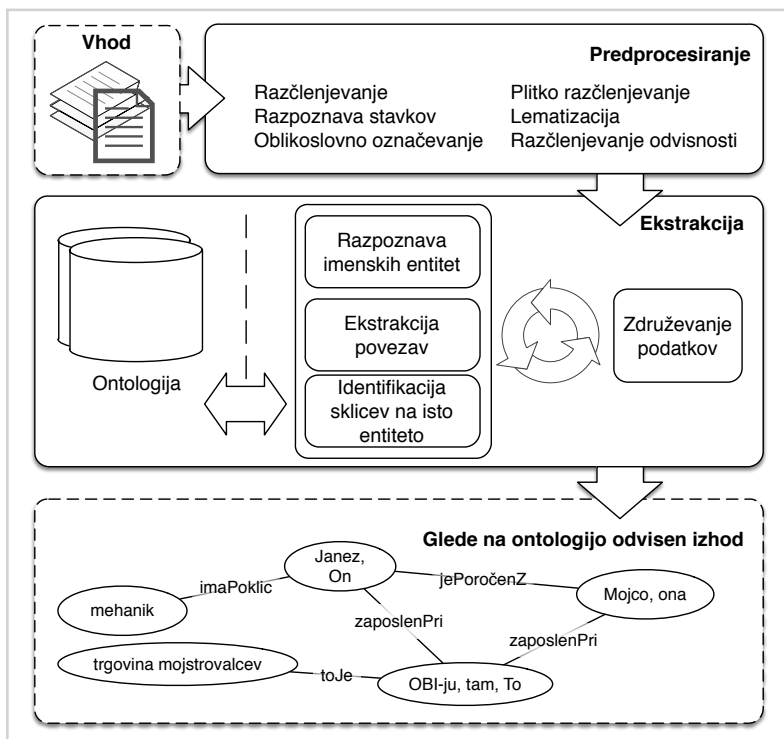
V tem razdelku predstavimo sistem za skupno in iterativno ekstrakcijo informacij. Modularni celostni sistem za ekstrakcijo informacij (angl. end-to-end information extraction system) v iterativnem postopku združuje vse tri glavne naloge ekstrakcije – prepoznavanje imenskih entitet, ekstrakcijo povezav in odkrivanje koreferenčnosti. Glavni komponenti sistema sta modul za predprocesiranje, v katerem obogatimo vhodne podatke in jih preoblikujemo v pravo obliko, in modul za ekstrakcijo, ki združuje vmesne rezultate ekstrakcij med iteracijami in kot zunanji vir podatkov uporablja ontologijo.

Celotna implementacija sistema je dostopna v javnem repozitoriju programske kode [20], ki uporablja linearno-verižne modele CRF iz knjižnice CRFSuite [135].

Predprocesiranje

Modul za predprocesiranje obogati vhodne podatke z dodatnimi atributi, ki jih kasneje potrebujejo metode ekstrakcije informacij. Modul razbije vhodno besedilo na posamezne povedi in člene (tj. besede) [31], jim doda leme [82], oblikoslovne oznake [33], odvisnostna drevesa [31], izvede plitko razčlenjevanje [33] in odkrije omenitve (s pomočjo pravil [70]).

Ta del postopka je edini, ki je zelo odvisen od jezika, v katerem je napisano besedilo. V primeru, da za izbran jezik ni razvitih nobenih postopkov predprocesiranja, moramo zaradi ustreznega delovanja ekstrakcije odkriti vsaj posamezne povedi in člene.



Slika 6.11

Sistem za iterativno in celostno ekstrakcijo informacij. Sistem sestoji iz modula za predprocesiranje in komponente za ekstrakcijo informacij. Slednja med seboj iterativno povezuje modul za razpoznavanje imenskih entitet, ekstrakcijo povezav in odkrivanje koreferenčnosti. Poleg tega na koncu vsake iteracije združi vmesne rezultate in zagotavlja povezavo do ontologije s podatki.

Ekstrakcija

Modul za ekstrakcijo sestoji iz iterativne metode za ekstrakcijo in metode za združevanje vmesnih rezultatov. V iterativni metodi ločeno izvajamo razpoznavanje imenskih entitet, ekstrakcijo povezav in odkrivanje koreferenčnosti. Vsi tipi ekstrakcije se izvajajo na istih podatkih in v vseh primerih so uporabljeni linearno-verižni modeli CRE. Zaradi uporabe enakih modelov ne potrebujemo dodatnih transformacij v predstavitvi podatkov in tudi atributske funkcije so lahko večkrat uporabljene. Omenitve (tj. zaporedja enega ali več členov) so tako vezane na člene in se lahko sklicujejo na sosednosti v izvornem besedilu. Njihovi obogateni atributi so enaki atributom prvega člena, ki ga vsebuje določena omenitev. V obratni smeri moramo prenesti le oznake imenskih entitet, saj razpoznavanje imenskih entitet deluje nad posameznimi členi, oba ostala

tipa ekstrakcije pa delujeta nad omenitvami. Ob koncu vsake iteracije metoda za združevanje podatkov združi rezultate ekstrakcij in jih predstavi kot graf, v katerem vozlišča predstavljajo entitete (tj. gruče omenitev) in povezave ekstrahirane povezave med entitetami. Dve entiteti sta povezani, če sta povezani njune omenitvi. Združene rezultate nato v naslednji iteraciji uporabljajo atributske funkcije za morebitno izboljšavo ekstrakcij.

Ontologija je v času ekstrakcije uporabljena v treh različnih kontekstih. Predstavlja vsaj shemo domene, za katero so naučeni modeli za ekstrakcijo informacij (tj. entitete in povezave so predstavljene kot njeni primerki konceptov in lastnosti). Poleg tega definira poljubne ostale koncepte, omejitve in pravila (npr. razdalja med koncepti, sosednost konceptov, regularni izraz, ki velja za primerke ali seznam znanih primerkov), ki jih uporabljajo atributske funkcije. Zaradi tega lahko uporabnik s spreminjanjem podatkov v ontologiji delno vpliva na ekstrakcijo informacij.

Rezultati celotnega sistema se torej iterativno posodabljaajo. Vrnjeni so kot semantični podatkovni graf, ki je označen skladno s sistemsko ontologijo.

6.4 Analiza rezultatov

Predlagani sistem smo evalvirali na podatkovni množici *Automatic Content Extraction (ACE) 2004* [38], saj edina vsebuje ročno označene imenske entitete, povezave in koreferenčnosti. Uporabili smo vse različne novičarske tipe besedil, ki jih vsebuje angleška različica podatkovne množice. Izbrana besedila sestojijo iz 443 dokumentov s predvidoma 191,387 členi in 7,518 povedmi. Bolj natančne statistike uporabljenih podatkov so predstavljene v tabeli 6.1. Lastna imena, ki jih moramo ekstrahirati, lahko predstavljajo osebe, organizacije, objekte (tj. imena stavb), lokacije in zemljepisna imena. Med njimi ekstrahiramo pet tipov povezav, bolj natančneje povezave vloga (tj. vloga, ki jo oseba ima v neki organizaciji), delež (tj. del neke celote ali večje enote), lokacija (tj. nahajanje na neki lokaciji ali prebivališču), bližina (tj. nahajanje v bližini druge lokacije) in družbenost (tj. družbene povezave med osebami). Vsaka izmed teh tipov povezav je nato razdeljena v bolj specifične povezave, ki jih je skupaj 24. Pri evalvaciji sistema uporabljamo že razpoznane omenitve in podatkovno množico razdelimo v učni (80%) in testni (20%) del.

V tabelah 6.2, 6.3 in 6.4 prikazujemo rezultate po vsaki izmed petih iteracij. Po prvi iteraciji dosežemo enake rezultate, kot bi jih dosegli, če bi vsak tip ekstrakcije izvajali neodvisno od drugih. Algoritmi za ekstrakcijo uporabljajo atributske funkcije,

Tabela 6.1

Opis učne (train) in testne (test) podatkovne množice ACE 2004.

	<i>Train</i>	<i>Test</i>
<i>Število dokumentov</i>	3 54	89
<i>Število povedi</i>	5,789	1,638
<i>Število členov</i>	140,237	35,490
<i>Število omenitev</i>	22,079	5,679
<i>Število entitet</i>	9,544	2,488
<i>Število povezav</i>	4,619	1,185

ki so natančno opisane v poglavjih 3, 4 in 5. Glede na prikazane porazdelitve razdalj med atributi povezav in koreferenčnimi omenitvami se odločimo, da za ekstrakcijo povezav zgradimo zaporedja z izpuščenimi omenitvami od nič do tri in za odkrivanje koreferenčnosti od nič do 25.

Tabela 6.2

Rezultati razpoznavanja imenskih entitet. Uporabljene mere so delež zmanjšanja napak v % (Napaka), klasifikacijska točnost (CA), makro-povprečna natančnost (MaP), makro-povprečen priklic (MaR), makro-povprečna ocena F (MaF) in mikro-povprečna ocena F (MiF).

<i>Model</i>	<i>Napaka (%)</i>	<i>CA</i>	<i>MaP</i>	<i>MaR</i>	<i>MaF</i>	<i>MiF</i>
Neodvisen	—	97.0	54.0	30.4	38.9	90.8
Druga iteracija	10.3	97.2	55.0	33.2	41.4	91.6
Tretja iteracija	15.0	97.8	55.2	33.5	41.7	92.2
Četrta iteracija	15.0	97.8	55.2	33.5	41.7	92.2
Peta iteracije	15.0	97.8	55.2	33.5	41.7	92.2

Rezultate predstavimo z metrikami, ki jih v rezultatih uporablja tudi večina raziskovalcev. Pri prepoznavanju imenskih entitet tako uporabimo makro in mikro-povprečno oceno F, saj je to problem napovedovanja v več razredov (tabela 6.2). Klasifikacijska točnost 97.8% pove malo o doseženih rezultatih, saj je v podatkovni množici ogromno členov, ki ne pripadajo nobenemu tipu imenske entitete. Zanimivo je le videti, da se do tretje iteracije delež napak zmanjša za 15%, tako kot podobno poročajo tudi Singh in sod. [205]. Med makro-povprečno oceno in mikro-povprečno oceno je velika razlika, saj makro-povprečje daje enako težo posameznim razredom v vseh dokumentih,

medtem ko mikro-povprečje povpreči rezultate iz vseh dokumentov hkrati. Makro-povprečje je nizko zato, ker so nekateri razredi imenskih entitet zelo slabo prepoznani. Cilj, ki smo ga hoteli doseči, pa lepo kaže, da se uspešnost metode tekom iteracij malenkostno izboljša.

Tabela 6.3

Rezultati ekstrakcije povezav. Uporabljene mere so delež zmanjšanja napak v % (Napaka), natančnost (P), priklic (R) in ocena F.

<i>Model</i>	<i>Napaka (%)</i>	<i>P</i>	<i>R</i>	<i>F</i>
Neodvisen	—	54.3	55.2	54.7
Druga iteracija	0.8	55.1	55.6	55.3
Tretja iteracija	2.4	54.8	55.6	55.2
Četrta iteracija	2.4	55.0	55.4	55.2
Peta iteracija	2.0	54.2	55.7	54.9

Pri ekstrakciji povezav uporabimo standardno oceno F, s katero merimo, ali smo povezavo napovedali pravilno ali ne - povezava se mora ujemati v imenu in v obeh atributih. Ocena SER, ki jo uporabimo v poglavju 4 je v praksi zelo redko uporabljena, a je bila izbrana kot ocenjevalni kriterij za tekmovanje BioNLP. Tudi pri ekstrakciji povezav dosežemo cilj in tekom iteracij vsaj malenkostno izboljšamo rezultate (tabela 6.3). Delež napak pri ekstrakciji se zmanjša, vendar z občutno manjšo razliko kot pri prepoznavanju lastnih imen. Pri prepoznavanju imenskih entitet opazimo, da se rezultati od tretje iteracije dalje ne izboljšujejo, medtem ko pri ekstrakciji povezav opazimo, da se v peti iteraciji rezultati celo malenkostno poslabšajo.

Tabela 6.4

Rezultati odkrivanja koreferenčnosti. Sistemi za odkrivanje koreferenčnosti se primerjajo glede na ocene MUC [131], BCu-bed [132] in CEAF [133].

<i>Model</i>	<i>MUC</i>	<i>BCubed</i>	<i>CEAF</i>
Neodvisen	73.2	73.9	49.8
Druga iteracija	73.8	73.5	50.0
Tretja iteracija	74.0	74.1	52.9
Četrta iteracija	74.3	73.8	52.8
Peta iteracija	74.3	73.8	52.8

Pri odkrivanju koreferenčnosti v primerjavi z ostalimi pristopi dosežemo 1.9% slabšo oceno MUC in 5.2% slabšo oceno BCubed kot trenutno najboljši znani rezultati, ki sta jih objavila Haghighi in Klein [113]. Kljub temu tako kot v prejšnjih primerih, tudi tu skupni rezultati kažejo podoben trend izboljšanj kot delo Singh-a in sod. [205], ki so pred kratkim predlagali skupen model CRF za prepoznavanje imenskih entitet, ekstrakcijo povezav in odkrivanje koreferenčnosti. Rezultati po peti iteraciji so v primerjavi z algoritmom SkipCor slabši za 0.7% ocene MUC in 4.8% ocene BCubed, kjer smo iz podatkovne množice uporabili vse ročno označene povezave in imenske entitete (6.4). Od tretje iteracije dalje se oceni BCubed in CEAf malenkostno poslabšata, medtem ko se rezultat za oceno MUC izboljša.

Opazimo lahko, da se rezultati vsake izmed nalog malenkostno izboljšajo po vsaki iteraciji ekstrakcije informacij, kar je bil tudi naš glavni cilj. Predvidevamo, da je v nadaljnjem delu smiselno uporabiti le tri iteracije, saj sklepamo, da se v drugi iteraciji lahko medsebojno uporabijo rezultati različnih nalog in se v tretji lahko kvečjemu propagirajo preko dveh klasifikatorjev za uporabo v tretjem – na primer v prvi iteraciji prepoznamo imensko entiteto, kar nam v drugi iteraciji omogoči pravilno ekstrakcijo povezave in nato v tretji iteraciji izboljšamo odkrivanje koreferenčnosti za izbrano imensko entiteto. Izboljšanje dosežemo le s skupno ekstrakcijo informacij preko več iteracij. Najbolj očitno se izboljšave opazijo pri prepoznavanju imenskih entitet, kjer po tretji iteraciji zmanjšamo število napačnih oznak za 15%. Izvedli smo tudi analizo, pri čemer smo onesposobili enega izmed ekstraktorjev, tako da smo pri prepoznavanju imenskih entitet vse člene označili kot ostale, da pri ekstrakciji povezav nismo razpoznali nobene povezave in da smo pri odkrivanju koreferenčnosti odkrili le gruče z eno omenitvijo. Ker so razlike v uspešnosti metod med iteracijami že tako zelo majhne, nismo mogli izvedeti, kateri klasifikator najbolj vpliva na izboljšanje ostalih dveh, saj so bile v teh primerih razlike med iteracijami še manjše. V nadaljnjem raziskovanju bomo zato poskušali dodatno izboljšati razlike med iteracijami, da bomo lahko pokazali značilne vplive med metodami ekstrakcije.

Zaključki

7

7.1 *Zaključki*

V disertaciji smo inovativno rešili vse tri glavne naloge ekstrakcije informacij in jih nazadnje povezali v iterativno metodo, pri čemer za vsako posebej uporabimo linearno-verižne modele pogojnih naključnih polj. Najprej smo predstavili sistem za odkrivanje koreferenčnosti – SkipCor, kjer smo zaradi omejitve izbranih modelov predlagali transformacijo zaporedij omenitev v zaporedja z izpuščenimi omenitvami. Za razliko od večine ostalih predlaganih pristopov, SkipCor odkriva koreferenčnosti nad zaporedji omenitev in ne le med pari. Sistem smo evalvirali na sedmih podatkovnih množicah, kjer smo dosegli primerljive ali boljše rezultate z ostalimi pristopi. Poleg tega smo preverili še pričakovano uspešnost algoritmov na realnih podatkih in ugotovili, da način označevanja podatkovne množice in izbor domene predstavlja veliko vlogo. Podobno kot pri sistemu SkipCor, smo zaporedja z izpuščenimi omenitvami prilagodili za problem ekstrakcije povezav. Razvili smo več različnih členitev vhodnih podatkov, s čimer smo omogočili ekstrakcijo hierarhičnih povezav. Skupaj z ročno izbranimi pravili za ekstrakcijo povezav smo dosegli najboljši rezultat v odkrivanju genskega regulatornega omrežja v okviru tekmovanja BioNLP 2013. Prav tako smo skušali transformacijo zaporedij uporabiti tudi pri prepoznavanju imenskih entitet, vendar nismo dosegli izboljšav. Kljub temu smo uporabili standarden pristop za ekstrakcijo nad zaporedji členov, kjer smo naučili več modelov ter združili njihove ocene. S tem pristopom smo na tekmovanju v odkrivanju kemijskih spojin in zdravil dosegli sedmo mesto. Nazadnje smo vse pristope združili v arhitekturi za celostno ekstrakcijo informacij, ki kot vhod sprejme besedilo in vrne označene podatke na podlagi sistemske ontologije. Ontologijo uporabimo na tri načine, a ključnega pomena je uporaba njene sheme za boljšo interoperabilnost ekstrahiranih podatkov. Vse tri glavne naloge ekstrakcije informacij združimo v iterativni metodi in pokažemo, da zaradi odvisnosti med njimi preko iteracij izboljšamo rezultate.

Glavni znanstveni doprinos v tem delu je predlog novega načina predstavitve podatkov, ki posledično omogoča, da lahko problema odkrivanja koreferenčnosti in razpoznavanja povezav učinkovito implementiramo s pomočjo verjetnostnih grafovskih modelov prvega reda. Omenjeni modeli so bili že sedaj zelo uspešni pri prepoznavanju imenskih entitet, kar smo tudi sami potrdili na biomedicinski domeni. Ker smo uspeli z enako podatkovno strukturo in istim tipom modela rešiti vse tri glavne probleme ekstrakcije informacij, smo nazadnje predlagali iterativni sistem za ekstrakcijo

informacij, kjer lahko iste atributske funkcije uporabimo za različne naloge in enostavno uporabimo vmesne rezultate med različnimi ekstrakcijami. Slednji sistem je pokazal malenkostno izboljšanje rezultatov preko več iteracij, kar želimo v prihodnosti še izboljšati. Vse predlagane metode so splošne in uporabne za poljubno domeno, za katero lahko pridobimo označeno podatkovno množico za izgradnjo klasifikatorjev. Podobno velja za jezik besedila, za katerega moramo najti le ustrezne obstoječe metode za predprocesiranje in ustrezne podatkovne vire, ki jih uporabljajo atributske funkcije – na primer, sezname entitet ali vrednosti v ontologiji.

V nadaljnjih raziskavah želimo še bolj natančno raziskati odvisnosti med posameznimi nalogami ekstrakcije. Poleg tega bomo pri posameznih nalogah preverili, kako zanesljivi so rezultati posameznih zaporedij z izpuščenimi omenitvami in predlagali metodo za bolj inteligentno združevanje rezultatov. Nazadnje bomo predlagali še metriko za ocenjevanje uspešnosti celostnih sistemov za ekstrakcijo informacij, ki združujejo vse tri glavne naloge.



Slovarček izrazov

A

A.1 Angleško-slovenski slovar izrazov

Področje ekstrakcije informacij za slovenski jezik je še zelo novo, zato prevodi angleških terminov še niso uveljavljeni ali dovolj znani, da bi jih vsi slovenski raziskovalci nedvoumno uporabljali. V tabelah A.1 in A.2 predstavimo prevode najbolj pogostih terminov, ki jih uporabljamo v disertaciji.

Tabela A.1

Prevodi terminov na področju ekstrakcije informacij iz angleškega jezika v slovenski jezik – 1/2.

<i>Angleški izraz</i>	<i>Slovenski prevod</i>
information retrieval	informacijsko poizvedovanje
information extraction	ekstrakcija informacij
relation extraction	ekstrakcija povezav
coreference resolution	odkrivanje koreferenčnosti
clustering	gručenje
classification	klasifikacija
part-of-speech tag	oblikoslovna oznaka
part-of-speech tagging	oblikoslovno označevanje
dependency treebank	drevesnica odvisnosti
dependency parser	razčlenjevalnik odvisnosti
named entity recognition	prepoznavanje imenskih entitet
entity extraction	ekstrakcija entitet
feature function	atributska funkcija
feature	značilka
factor graph	faktorski graf
conditional random fields (CRF)	pogojna naključna polja (CRF)
linear-chain CRF	linearno-verižni CRF
token	člen
tokenization	razčlenjevanje
natural language processing	procesiranje naravnega jezika
F score	ocena F

Tabela A.2

Prevodi terminov na področju ekstrakcije informacij iz angleškega jezika v slovenski jezik – 2/2.

<i>Angleški izraz</i>	<i>Slovenski prevod</i>
recall	priklic
precision	natančnost
hidden Markov models	prikriti markovski modeli
probabilistic graphical models	verjetnostni grafovski modeli
dataset, corpus	podatkovna množica
labeling, tagging	označevanje
subject	osebek
predicate	povezava
object	predmet
mention	omenitev
mention extraction	ekstrakcija omenitev
sequence	zaporedje
skip-mention sequence	zaporedje izpuščenih omenitev
shallow parsing	plitko razčlenjevanje
coreference	koreferenca
maximum entropy models	modeli največje entropije
semantic role labeling	semantično označevanje vlog
relationship descriptor	opis povezave
anaphora resolution	odkrivanje anafor
apposition	antonomazija
development dataset	podatkovna množica za nastavljanje parametrov



LITERATURA

- [1] Vannevar Bush. As We May Think. *The Atlantic*, July 1945.
- [2] Oren Etzioni. Search needs a shake-up. *Nature*, 476 (7358):25–26, 2011.
- [3] C. Chang, M. Kayed, M. R. Girgis, K. F. Shaalan, et al. A survey of web information extraction systems. *IEEE transactions on knowledge and data engineering*, 18(10):1–18, 2006.
- [4] Nilesh Dalvi, Ashwin Machanavajhala, and Bo Pang. An analysis of structured data on the web. *Proceedings of the VLDB Endowment*, 5(7):680–691, 2012. doi: [10.14778/2180912.2180920](https://doi.org/10.14778/2180912.2180920).
- [5] Peter Mika. Microsearch: An interface for semantic search. *Semantic Search*, pages 79–88, 2008.
- [6] Boštjan Pajntar and Marko Grobelnik. Searchpoint-a new paradigm of web search. In *17th World Wide Web (WWW) Conference-Developers Track*, pages 1–2, 2008.
- [7] F. Shaikh, U.A Siddiqui, I Shahzadi, S.I Jami, and Z.A Shaikh. SWISE: Semantic Web based intelligent search engine. In *International Conference on Information and Emerging Technologies (ICIET)*, pages 1–5, June 2010. doi: [10.1109/ICIET.2010.5625670](https://doi.org/10.1109/ICIET.2010.5625670).
- [8] Miriam Fernández, Iván Cantador, Vanesa López, David Vallet, Pablo Castells, and Enrico Motta. Semantically enhanced information retrieval: An ontology-based approach. *Web Semantics*, 9(4): 434–452, 2011. doi: [10.1016/j.websem.2010.11.003](https://doi.org/10.1016/j.websem.2010.11.003).
- [9] Portal slovensčina.eu, 2013. Dostopno na <http://www.slovenscina.eu/> (zadnjič obiskano 24. julija 2014).
- [10] Mihael Arčan and Špela Vintar. Avtomatično prepoznavanje lastnih imen. In *Konferenca za jezikovne tehnologije*, pages 1–6, 2012.
- [11] Slavko Žitnik. Collective ontology-based information extraction using probabilistic graphical models. In *CAISE-DC-2012 : proceedings of CAISE 2012 Doctoral Consortium*, pages 1–10, Gdansk, Poland, 2012.
- [12] Slavko Žitnik, Lovro Šubelj, Dejan Lavbič, Aljaž Zrnc, and Marko Bajec. Collective information extraction using first-order probabilistic models. In *BCI 2012 : proceedings*, pages 279–282, Novi Sad, Serbia, 2012.
- [13] Slavko Žitnik, Marinka Žitnik, Blaž Zupan, and Marko Bajec. Extracting Gene Regulation Networks Using Linear-Chain Conditional Random Fields and Rules. In *Proceedings of BioNLP Shared Task 2013 Workshop*, pages 178–187, 2013.
- [14] Slavko Žitnik and Marko Bajec. Token- and constituent-based linear-chain crf with svm for named entity recognition. In *BioCreative Challenge Evaluation Workshop*, volume 2, pages 144–151, Washington, 2013.
- [15] Slavko Žitnik, Lovro Šubelj, and Marko Bajec. Skip-Cor: Skip-mention coreference resolution using linear-chain conditional random fields. *PLoS ONE*, 9(6):e100101, June 2014. doi: [10.1371/journal.pone.0100101](https://doi.org/10.1371/journal.pone.0100101).
- [16] Slavko Žitnik, Marinka Žitnik, Blaž Zupan, and Marko Bajec. Sieve-based relation extraction of gene regulatory networks from biological literature. *BMC Bioinformatics*, 2014. ČAKA NA OBJAVO.
- [17] Slavko Žitnik and Marko Bajec. Multiple linear-chain conditional random fields with a meta-classifier for extracting chemical entity mentions. *Journal of Cheminformatics*, 2014. V RECENZIJU.
- [18] Slavko Žitnik, Lovro Šubelj, Dejan Lavbič, Olegas Vasilecas, and Marko Bajec. General Context-Aware Data Matching and Merging Framework. *Informatica*, 24(1):119–152, 2013.

- [19] Slavko Žitnik and Marko Bajec. Text mining in medicine. In Goran Rakocevic, Tijana Djukic, Nenad Filipovic, and Veljko Milutinović, editors, *Computational Medicine in Data Mining and Modeling*, pages 105–134. Springer New York, 2013. ISBN 978-1-4614-8784-5.
- [20] Slavko Žitnik. Intelligent Ontology-based Information Extraction - IOBIE, repozitorij programske kode, 2014. Dostopno na <https://bitbucket.org/szitnik/iobie> (zadnjič obiskano 24. julija 2014).
- [21] Sunita Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.
- [22] Hamed Hassanzadeh and MohammadReza Keyvanspour. A machine learning based analytical framework for semantic annotation requirements. *International Journal of Web & Semantic Technology*, 2(4):27–38, 2011.
- [23] Katharina Kaiser and Silvia Miksch. Information extraction: A Survey. Technical report, Vienna University of Technology, Institute of Software Technology and Interactive Systems, 2005.
- [24] Lawrence Reeve and Hyoil Han. Survey of semantic annotation platforms. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, SAC '05, pages 1634–1638, New York, NY, USA, 2005. ACM. doi: [10.1145/1066677.1067049](https://doi.org/10.1145/1066677.1067049).
- [25] Peggy M. Andersen, Philip J. Hayes, Alison K. Huettner, Linda M. Schmandt, Irene B. Nirenburg, and Steven P. Weinstein. Automatic extraction of facts from press releases to generate news stories. In *Proceedings of the third conference on Applied natural language processing*, pages 170–177, Pennsylvania, 1992. Association for Computational Linguistics.
- [26] Daya C. Wimalasuriya and Dejing Dou. Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science*, 36(3):306–323, 2010.
- [27] Illhoi Yoo and Min Song. Biomedical ontologies and text mining for biomedicine and healthcare: A survey. *Journal of Computing Science and Engineering*, 2(2): 109–136, 2008.
- [28] Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. Overview of BioNLP Shared Task 2013. In *Proceedings of BioNLP Shared Task 2013 Workshop*, pages 1–7, 2013.
- [29] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.
- [30] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.
- [31] Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. Parsing With Compositional Vector Grammars. In *Proceedings of the Association for Computational Linguistics*, pages 1–11, Pennsylvania, 2013. Association for Computational Linguistics.
- [32] Steven Abney and Steven P. Abney. *Parsing By Chunks*. Kluwer Academic Publishers, 1991.
- [33] Apache OpenNLP: a machine learning based toolkit for the processing of natural language text. Dostopno na <http://opennlp.apache.org/> (zadnjič obiskano 24. julija 2014).
- [34] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: [10.3115/1118108.1118117](https://doi.org/10.3115/1118108.1118117).
- [35] Song Mao, Azriel Rosenfeld, and Tapas Kanungo. Document structure analysis algorithms: A literature survey. volume 5010, pages 197–207, 2003.
- [36] D. W. Embley, Y. Jiang, and Y.-K. Ng. Record-boundary discovery in web documents. *SIGMOD Record*, 28(2):467–478, 1999. doi: [10.1145/304181.304223](https://doi.org/10.1145/304181.304223).
- [37] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. VIPS: A vision-based page segmentation algorithm. Technical report, Microsoft technical report, MSR-TR-2003-79, 2003.
- [38] George Doddington, Alexis Mitchell, Mark Przybicki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program—tasks, data, and evaluation. In *Proceedings of LREC*, volume 4, pages 837–840. Paris: European Language Resources Association, 2004.
- [39] Marta Recasens, Lluís Màrquez, Emili Sapena, Antònia M. Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. SemEval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 1–8. Stroudsburg: Association for Computational Linguistics, 2010.
- [40] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task*, pages 1–40. Stroudsburg: Association for Computational Linguistics, 2012.

- [41] Larry Smith, Lorraine K Tanabe, Rie Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Hadlow, Craig A Struble, Richard J Povinelli, Andreas Vlachos, William A Baumgartner, Lawrence Hunter, Bob Carpenter, Richard Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter Adriaans, Christian Blaschke, Rafael Torres, Mariana Neves, Preslav Nakov, Anna Divoli, Manuel Maña-López, Jacinto Mata, and W John Wilbur. Overview of BioCreative II gene mention recognition. *Genome Biology*, 9(Suppl 2):S2, 2008.
- [42] Claire Nédellec. Learning language in logic-genic interaction extraction challenge. In *Proceedings of the 4th Learning Language in Logic Workshop (LLL05)*, volume 7, pages 1–7. Bonn: ACM, 2005.
- [43] L.F. Rau. Extracting company names from text. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, volume 1, pages 29–32, Feb 1991. doi: [10.1109/CAIA.1991.120841](https://doi.org/10.1109/CAIA.1991.120841).
- [44] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. doi: [10.3115/992628.992709](https://doi.org/10.3115/992628.992709).
- [45] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguistic Investigations*, 30(1):3–26, 2007.
- [46] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, page 147–155, 2009.
- [47] Samet Atdalug and Vincent Labatut. A comparison of named entity recognition tools applied to biographical texts. *arXiv preprint arXiv:1308.0661*, pages 1–6, 2013.
- [48] Nancy Chinchor, P. Robinson, and E. Brown. Hub-4 named entity task definition. In *Proceedings of DARPA Broadcast News Workshop*, 1998.
- [49] Nancy A. Chinchor. Overview of MUC-7/MET-2. In *Proceedings of the 7th Message Understanding Conference*, 1998.
- [50] Satoshi Sekine and Hitoshi Isahara. Irex: Ir and ie evaluation project in japanese. In *Proceedings of International Conference on Language Resources & Evaluation (LREC 2000)*, 2000.
- [51] Erik F. Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, pages 1–4, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
- [52] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [53] Diana Santos, Nuno Seco, Nuno Cardoso, and Rui Vilela. Harem: An advanced ner evaluation contest for portuguese. In *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC'2006*, pages 1986–1991, 2006.
- [54] Saul Kripke. *Naming and Necessity*. Boston: Harvard University Press, 1982.
- [55] Martha Palmer, Daniel Gildea, and Nianwen Xue. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103, 2010.
- [56] Sergey Brin. Extracting patterns and relations from the world wide web. *The World Wide Web and Databases*, 1:172–183, 1999.
- [57] Nguyen Bach and Sameer Badaskar. A review of relation extraction. *Literature review for Language and Statistics II*, pages 1–15, 2007.
- [58] Benjamin Hachey. *Towards generic relation extraction*. PhD thesis, 2009.
- [59] Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pages 401–408. Stroudsburg: Association for Computational Linguistics, 2006.
- [60] Michele Banko and Oren Etzioni. The tradeoffs between open and traditional relation extraction. In *Proceedings of the Association for Computational Linguistics*, pages 28–36, 2008.
- [61] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Proceedings of EMNLP 2011*, pages 1–12, 2011.
- [62] Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, and others. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, page 523–534. Association for Computational Linguistics, 2012.

- [63] Yaliang Li, Jing Jiang, Hai Leong Chieu, and Kian Ming A. Chai. Extracting relation descriptors with conditional random fields. pages 392–400, Thailand, 2011. Asian Federation of Natural Language Processing.
- [64] Pradheep Elango. Coreference resolution: A survey. Technical report, University of Wisconsin, Madison, 2005.
- [65] Jonathan H. Clark and José P. González-Brenes. Coreference resolution: Current trends and future directions. *Language and Statistics II Literature Review*, pages 1–14, 2008.
- [66] Jie Cai and Michael Strube. End-to-end coreference resolution via hypergraph partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 143–151, Pennsylvania, 2010. Association for Computational Linguistics.
- [67] Vincent Ng. Unsupervised models for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 640–649, Pennsylvania, 2008. Association for Computational Linguistics.
- [68] Constantin Orasan, Dan Cristea, Ruslan Mitkov, and António Branco. Anaphora resolution exercise: An overview. In *Proceedings of LREC*, pages 1–5, 2008.
- [69] Xiaoqiang Luo. Coreference or not: A twin model for coreference resolution. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 73–80, New York, 2007. Association for Computational Linguistics.
- [70] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihail Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34, Pennsylvania, 2011. Association for Computational Linguistics.
- [71] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544, 2001.
- [72] Eric Bengtson and Dan Roth. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303, Pennsylvania, 2008. Association for Computational Linguistics.
- [73] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111, Philadelphia, 2002. Association for Computational Linguistics.
- [74] Constantin Orasan and Richard Evans. NP animacy identification for anaphora resolution. *Journal of Artificial Intelligence Research*, 29(1):79–103, 2007.
- [75] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.
- [76] B. Yildiz and S. Miksch. Motivating ontology-driven information extraction. In *International Conference on Semantic Web and Digital Libraries. Indian Statistical Institute Platinum Jubilee Conference Series*, pages 45–53, 2007.
- [77] S. Bloehdorn, P. Cimiano, A. Hotho, and S. Staab. An ontology-based framework for text mining. *GLDV-Journal for Computational Linguistics and Language Technology*, 20:87–112, 2005.
- [78] L. McDowell and M. Cafarella. Ontology-driven information extraction with ontophyon. *The Semantic Web-ISWC 2006*, pages 428–444, 2006.
- [79] Wilson Wong, Wei Liu, and Mohammed Bennaoun. Ontology learning from text: A look back and into the future. *ACM Computer Surveys*, 44(4):20:1–20:36, 2012. doi: [10.1145/2333112.2333115](https://doi.org/10.1145/2333112.2333115).
- [80] Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. Ontogen: Semi-automatic ontology editor. In *Proceedings of the 2007 Conference on Human Interface: Part II*, pages 309–318, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-73353-9.
- [81] Adam Przepiórkowski. Slavonic information extraction and partial parsing. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*, page 1–10, 2007.
- [82] Matjaž Juršič, Igor Mozetič, Tomaž Erjavec, and Nada Lavrač. Lemmagen : multilingual lemmatisation with induced ripple-down rules. *Journal of University Computer Science*, 16(9):1190–1214, 2010.
- [83] Miha Grčar, Simon Krek, and Kaja Dobrovoljc. Obeliks: statistični oblikoskladenjski označevalnik in lematizator za slovenski jezik. In *Zbornik Osmo konference Jezikovne tehnologije*, 2012.
- [84] Kaja Dobrovoljc, Simon Krek, and Jan Rupnik. Skladdenjski razčlenjevalnik za slovenščino. In *Zbornik Osmo konference Jezikovne tehnologije*, 2012.
- [85] Špela Vintar and Darja Fišer. Adding multi-word expressions to slownet. *Research Infrastructure for Digital Lexicography*, page 56, 2009.
- [86] George A. Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

- [87] Zbornik prispevkov konference za jezikovne tehnologije. 2012.
- [88] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007. ISBN 0262072882.
- [89] Charles Sutton. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.
- [90] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. San Francisco: Morgan Kaufmann, 2001.
- [91] Eric Fosler-Lussier, Yanzhang He, Preethi Jyothi, and Rohit Prabhavalkar. Conditional random fields in speech, audio, and language processing. *Proceedings of the IEEE*, 101(5):1054–1075, 2013. doi: [10.1109/JPROC.2013.2248112](https://doi.org/10.1109/JPROC.2013.2248112).
- [92] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4, CONLL '03*, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [93] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Pennsylvania, 2005. Association for Computational Linguistics.
- [94] Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. Efficient, feature-based, conditional random field parsing. In *In Proceedings of ACL/HLT*, pages 1–9, 2008.
- [95] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03*, pages 134–141, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [96] Yanxin Shi and Mengqiu Wang. A dual-layer crfs based joint decoding method for cascaded segmentation and labeling tasks. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07*, pages 1707–1712, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
- [97] Noah A. Smith, David A. Smith, and Roy W. Tromble. Context-based morphological disambiguation with random fields. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05*, pages 475–482, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
- [98] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 230–237, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [99] Trevor Cohn. Efficient inference in large conditional random fields. In *Proceedings of the 17th European conference on Machine Learning*, pages 606–613, Heidelberg, 2006. Springer.
- [100] Andrew McCallum and Ben Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Neural Information Processing Systems*, pages 1–8, 2004.
- [101] Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 593–601, Arlington, 2004. AUAI Press.
- [102] Trevor A. Cohn. *Scaling Conditional Random Fields for Natural Language Processing*. University of Melbourne, Department of Computer Science and Software Engineering, Faculty of Engineering, 2007.
- [103] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. ISSN 0018-9219.
- [104] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00*, pages 591–598, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [105] Léon Bottou. *Une Approche théorique de l'Apprentissage Connexionniste: Applications à la Reconnaissance de la Parole*. PhD thesis, Université de Paris XI, Orsay, France, 1991.
- [106] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions of Information Theory*, 47(2), September 2006. doi: [10.1109/18.910572](https://doi.org/10.1109/18.910572).
- [107] J Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3), September 1986. doi: [10.1016/0004-3702\(86\)90072-X](https://doi.org/10.1016/0004-3702(86)90072-X).

- [108] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI 99, pages 467–475, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.
- [109] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2), November 1999. doi: [10.1023/A:1007665907178](https://doi.org/10.1023/A:1007665907178).
- [110] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, January 2008. doi: [10.1561/2200000001](https://doi.org/10.1561/2200000001).
- [111] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal Scientific Computing*, 16(5):1190–1208, September 1995. doi: [10.1137/0916069](https://doi.org/10.1137/0916069).
- [112] Aron Culotta, Michael Wick, Robert Hall, and Andrew McCallum. First-order probabilistic models for coreference resolution. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 81–88, 2007.
- [113] Aria Haghighi and Dan Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 3, pages 1152–1161, Pennsylvania, 2009. Association for Computational Linguistics.
- [114] Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 2, pages 968–977, 2009.
- [115] Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhata, and Salim Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 136–143, Pennsylvania, 2004. Association for Computational Linguistics.
- [116] Shujian Huang, Yabing Zhang, Junsheng Zhou, and Jiajun Chen. Coreference resolution using markov logic networks. *Advances in Computational Linguistics*, 41:157–168, 2009.
- [117] Veselin Stoyanov, Claire Cardie, Nathan Gilbert, Ellen Riloff, David Buttlar, and David Hysom. Coreference resolution with reconcile. In *Proceedings of the Association for Computational Linguistics 2010 Conference - Short Papers*, pages 156–161, Pennsylvania, 2010. Association for Computational Linguistics.
- [118] Karthik Raghunathan, Heeyoung Lee, Sudarshan Ranganathan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501, Pennsylvania, 2010. Association for Computational Linguistics.
- [119] Cosmin Bejan, Matthew Titsworth, Andrew Hickl, and Sanda Harabagiu. Nonparametric bayesian models for unsupervised event coreference resolution. In *Advances in Neural Information Processing Systems*, pages 73–81, 2009.
- [120] Cosmin Adrian Bejan and Sanda Harabagiu. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422, Stroudsburg, 2010. Association for Computational Linguistics.
- [121] Eugene Charniak. Unsupervised learning of name structure from coreference data. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–7, Stroudsburg, 2001. Association for Computational Linguistics.
- [122] R. Vijay Sundar Ram and Sobha Lalitha Devi. Coreference resolution using tree CRFs. In *Proceedings of Conference on Intelligent Text Processing and Computational Linguistics*, pages 285–296, Heidelberg, 2012. Springer.
- [123] Samuel Broscheit, Massimo Poesio, Simeno Paolo Ponzetto, Kepa Joseba Rodriguez, Lorenza Romano, Olga Uryupina, Yannick Versley, and Roberto Zanoli. BART: a multilingual anaphora resolution system. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 104–107, Pennsylvania, 2010. Association for Computational Linguistics.
- [124] Giuseppe Attardi, Stefano Dei Rossi, and Maria Simi. TANL-1: coreference resolution by parse analysis and similarity clustering. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 108–111, Pennsylvania, 2010. Association for Computational Linguistics.
- [125] Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidú. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Proceedings of CoNLL 2012 Joint Conference on EMNLP and CoNLL*, pages 41–48, Pennsylvania, 2012. Association for Computational Linguistics.
- [126] Mohit Bansal and Dan Klein. Coreference semantics from web features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 389–398. Stroudsburg: Association for Computational Linguistics, 2012.

- [127] Shane Bergsma and Dekang Lin. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 33–40, Pennsylvania, 2006. Association for Computational Linguistics.
- [128] William E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods*, pages 354–359. American Statistical Association, 1990.
- [129] Nancy Chincor. MUC-3 evaluation metrics. In *Proceedings of the 3rd conference on Message understanding*, pages 17–24, Pennsylvania, 1991. Association for Computational Linguistics.
- [130] Nancy Chinchor and Beth Sundheim. MUC-5 evaluation metrics. In *Proceedings of the 5th conference on Message understanding*, pages 69–78, Pennsylvania, 1993. Association for Computational Linguistics.
- [131] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, pages 45–52, Pennsylvania, 1995. Association for Computational Linguistics.
- [132] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 1–7, 1998.
- [133] Xiaoqiang Luo. On coreference resolution performance metrics. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32, Pennsylvania, 2005. Association for Computational Linguistics.
- [134] Lynette Hirschman and Nancy A. Chincor. MUC-7 coreference task definition. In *Proceedings of the Seventh Message Understanding Conference*, pages 1–17, San Francisco, 1997. Morgan Kaufmann.
- [135] Naoaki Okazaki. CRFSuite: a fast implementation of conditional random fields (CRFs), 2007. Dostopno na <http://www.chokkan.org/software/crfsuite> (zadnjič obiskano 24 julija 2014).
- [136] Jenny Rose Finkel and Christopher D. Manning. Enforcing transitivity in coreference resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 45–48, 2008.
- [137] Anders Björkelund and Richárd Farkas. Data-driven multilingual coreference resolution using resolver stacking. In *Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task*, pages 49–55, 2012.
- [138] Chen Chen and Vincent Ng. Combining the best of two worlds: A hybrid approach to multilingual coreference resolution. In *Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task*, pages 56–63, 2012.
- [139] Marcus Stamborg, Dennis Medved, Peter Exner, and Pierre Nugues. Using syntactic dependencies to solve coreferences. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 64–70, 2012.
- [140] Desislava Zhekova, Sandra Kübler, Joshua Bonner, Marwa Ragheb, and Yu-Yin Hsu. UBIU for multilingual coreference resolution in OntoNotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 88–94, 2012.
- [141] Xinxin Li, Xuan Wang, and Xingwei Liao. Simple maximum entropy models for multilingual coreference resolution. In *Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task*, pages 83–88, 2012.
- [142] Emili Sapena, Lluís Padró, and Jordi Turmo. RelaxCor: a global relaxation labeling approach to coreference resolution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 88–91, 2010.
- [143] Hamidreza Kobdani and Hinrich Schütze. SUCRE: a modular system for coreference resolution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 92–95, 2010.
- [144] Desislava Zhekova and Sandra Kübler. UBIU: a language-independent system for coreference resolution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 96–99, 2010.
- [145] Nathan Gilbert and Ellen Riloff. Domain-specific coreference resolution with lexicalized features. pages 1–6, Pennsylvania, 2013. Association for Computational Linguistics.
- [146] Sebastian Martschat and Michael Strube. Recall error analysis for coreference resolution. pages 2070–2081, 2014.
- [147] Jonathan K. Kummerfeld and Dan Klein. Error-driven analysis of challenges in coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1–13, 2013.
- [148] Olga Uryupina. Error Analysis for Learning-based Coreference Resolution. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 1914–1919, 2008.
- [149] Martin Krallinger and Alfonso Valencia. Text-mining and information-retrieval services for molecular biology. *Genome Biology*, 6(7):224–232, 2005.

- [150] Asako Koike, Yoshiyuki Kobayashi, and Toshihisa Takagi. Kinase pathway database: an integrated protein-kinase and NLP-based protein-interaction resource. *Genome Research*, 13(6a):1231–1243, 2003.
- [151] Hans-Michael Müller, Eimear E Kenny, and Paul W Sternberg. Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biology*, 2(11):e309, 2004.
- [152] Zuoshuang Xiang, Tingting Qin, Zhaohui S Qin, and Yongqun He. A genome-wide MeSH-based literature mining system predicts implicit gene-to-gene relationships and networks. *BMC Systems Biology*, 7(Suppl 3):S9, 2013.
- [153] Yan Xu, Kai Hong, Junichi Tsujii, I Eric, and Chao Chang. Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries. *Journal of the American Medical Informatics Association*, 19(5):824–832, 2012.
- [154] Myungjae Kwak, GONDY Leroy, Jesse D Martinez, and Jeffrey Harwell. Development and evaluation of a biomedical search engine using a predicate-based vector space model. *Journal of Biomedical Informatics*, 46(5):929–939, 2013.
- [155] Tatjana Sauka-Spengler and Marianne Bronner-Fraser. A gene regulatory network orchestrates neural crest formation. *Nature reviews Molecular cell biology*, 9(7):557–568, 2008.
- [156] Lesley T MacNeil and Albertha J.M. Walhout. Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. *Genome Research*, 21(5):645–657, 2011.
- [157] Isabelle S Peter and Eric H Davidson. A gene regulatory network controlling the embryonic specification of endoderm. *Nature*, 474(7353):635–639, 2011.
- [158] Dayne Freitag and Andrew McCallum. Information extraction with HMM structures learned by stochastic optimization. In *Proceedings of the National Conference on Artificial Intelligence*, pages 584–589. AAAI Press, 2000.
- [159] Nanda Kambhata. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, pages 1–22. Stroudsburg: Association for Computational Linguistics, 2004.
- [160] Sofie Van Landeghem, Jari Björne, Thomas Abeel, Bernard De Baets, Tápío Salakoski, and Yves Van de Peer. Semantically linking molecular entities in literature through entity relationships. *BMC Bioinformatics*, 13(Suppl 11):S6, 2012.
- [161] Marcos Garcia and Pablo Gamallo. Dependency-based text compression for semantic relation extraction. *Information Extraction and Knowledge Acquisition*, pages 1–8, 2011.
- [162] Lance A. Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, pages 82–94. Netherlands: Springer, 1995.
- [163] Alexander Yates and Oren Etzioni. Unsupervised resolution of objects and relations on the web. In *Proceedings of NAACL HLT*, pages 121–130. Stroudsburg: Association for Computational Linguistics, 2007.
- [164] Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. ACE 2004 Multilingual Training Corpus. *Linguistic Data Consortium, Philadelphia*, 1:1–1, 2005.
- [165] Razvan C. Bunescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731, 2005.
- [166] Ting Wang, Yaoyong Li, Kalina Bontcheva, Hamish Cunningham, and Ji Wang. Automatic Extraction of Hierarchical Relations from Text. In *Proceedings of the 3rd European Conference on The Semantic Web: Research and Applications*, pages 215–229. Berlin: Springer-Verlag, 2006.
- [167] Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. Overview of the ID, EPI and REL tasks of BioNLP shared task 2011. *BMC bioinformatics*, 13(Suppl 11):S2, 2012.
- [168] Robert Bossy, Philippe Bessière, and Claire Nédellec. BioNLP shared task 2013 - an overview of the genic regulation network task. In *Proceedings of BioNLP Shared Task 2013 Workshop*, pages 153–160, 2013.
- [169] A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, J. Lin, P. Minguez, P. Bork, C. von Mering, and L. J. Jensen. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(D1):808–815, 2012.
- [170] Haibin Liu, Tom Christiansen, William A. Baumgartner Jr, and Karin Verspoor. BioLemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of Biomedical Semantics*, 3:1–29, 2012.
- [171] John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel. Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*, pages 249–252. DARPA, 1999.

- [172] Thomas Provoost and Marie-Francine Moens. Detecting relations in the gene regulation network. In *Proceedings of BioNLP Shared Task 2013 Workshop*, pages 135–138, 2013.
- [173] Jari Björne and Tapio Salakoski. TEES 2.1: Automated annotation scheme learning in the bioNLP 2013 shared task. In *Proceedings of BioNLP Shared Task 2013 Workshop*, pages 16–25, 2013.
- [174] Vincent Claveau. IRISA participation to bioNLP-ST13: lazy-learning and information retrieval for information extraction tasks. In *Proceedings of BioNLP Shared Task 2013 Workshop*, pages 188–196, 2013.
- [175] Kai Hakala, Sofie Van Landeghem, Tapio Salakoski, Yves Van de Peer, and Filip Ginter. EVEX in ST'13: Application of a large-scale text mining resource to event extraction and network construction. In *Proceedings of BioNLP Shared Task 2013 Workshop*, pages 26–34, 2013.
- [176] Rosario M Piro and Ferdinando Di Cunto. Computational approaches to disease-gene prediction: rationale, classification and successes. *The FEBS Journal*, 279(5):678–696, 2012.
- [177] Yves Moreau and Léon-Charles Tranchevent. Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nature Reviews Genetics*, 13(8):523–536, 2012.
- [178] Allan Peter Davis, Thomas C Wieggers, Robin J Johnson, Jean M Lay, Kelley Lennon-Hopkins, Cynthia Saraceni-Richards, Daniela Sciak, Cynthia Grondin Murphy, and Carolyn J Mattingly. Text mining effectively scores and ranks the literature for improving chemical-gene-disease curation at the comparative toxicogenomics database. *PLoS One*, 8(4):e58201, 2013.
- [179] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, 1:1–5, 2013.
- [180] Miguel Vazquez, Martin Krallinger, Florian Leitner, and Alfonso Valencia. Text mining for drugs and chemical compounds: Methods, tools and applications. *Molecular Informatics*, 30(6-7):506–519, 2011.
- [181] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [182] Dingcheng Li, Karin Kipper-Schuler, and Guergana Savova. Conditional random fields and support vector machines for disorder named entity recognition in clinical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 94–95. Stroudsburg: Association for Computational Linguistics, 2008.
- [183] Guilherme Hoefel and Charles Elkan. Learning a two-stage svm/crf sequence classifier. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 271–278. New York: ACM, 2008.
- [184] Peng Cai, Hangzai Luo, and Aoying Zhou. Semantic entity detection by integrating CRF and SVM. In *Proceedings of the 11th international conference on Web-Age Information Management*, pages 483–494. Berlin: Springer, 2010.
- [185] Yanan Lu, Xiaoyuan Yao, Xiaomei Wei, and Ji Donghong. WHU-BioNLP CHEMDNER System with Mixed Conditional Random Fields and Word Clustering. In *BioCreative Challenge Evaluation Workshop*, volume 2, pages 129–134. Washington, 2013.
- [186] DrugBank: Open Data Drug and Drug Target Database, . URL <http://www.drugbank.ca/>.
- [187] CTD Database: Comparative Toxicogenomics Database, . URL <http://ctdbase.org/>.
- [188] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [189] Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julien Oyarzabal, and Alfonso Valencia. Overview of the chemical compound and drug name recognition (CHEMDNER) task. In *BioCreative Challenge Evaluation Workshop*, volume 2, pages 2–33. Washington, 2013.
- [190] Bin Liu, Laura Chiticariu, Vivian Chu, H. V. Jagadish, and Frederick R Reiss. Automatic rule refinement for information extraction. In *Proceedings of the VLDB Endowment*, volume 3, 2010.
- [191] Nilesh Dalvi, Ravi Kumar, and Mohamed Soliman. Automatic wrappers for large scale web extraction. *Proceedings of the VLDB Endowment*, 4(4):219–230, 2011. ISSN 2150–8097.
- [192] John Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, pages 151–175, 2012.
- [193] Jing Jiang and ChengXiang Zhai. A systematic exploration of the feature space for relation extraction. In *Proceedings of the HLT-NAACL*, pages 113–120, 2007.
- [194] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106, 2003.
- [195] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. pages 1–7, 2007.

- [196] Philipp Cimiano, Uwe Reyle, and Jasmin Šarić. Ontology-driven discourse analysis for information extraction. *Data & Knowledge Engineering*, 55(1): 59–83, 2005. Natural Language and Database and Information Systems {NLDB} 03.
- [197] Ashwin Ittoo and Gosse Bouma. Minimally-supervised extraction of domain-specific part-whole relations using wikipedia as knowledge-base. *Data & Knowledge Engineering*, 85(0):57–79, 2013. Natural Language for Information Systems: Communicating with Anything, Anywhere in Natural Language.
- [198] Borislav Popov, Atanas Kiryakov, Damyan Ognyanoff, Dimitar Manov, and Angel Kirilov. Kim – a semantic platform for information extraction and retrieval. *Natural Language Engineering*, 10(3-4): 375–392, 2004.
- [199] Sriparna Saha and Asif Ekbal. Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. *Data & Knowledge Engineering*, 85(0):15–39, 2013. Natural Language for Information Systems: Communicating with Anything, Anywhere in Natural Language.
- [200] Razvan Bunescu and Raymond J. Mooney. Collective information extraction with relational markov networks. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.
- [201] Claire Nedellec and Adeline Nazarenko. Ontologies and information extraction. *CoRR*, abs/cs/0609137, 2006.
- [202] Xiaofeng Yu and Wai Lam. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1–9. Association for Computational Linguistics, 2010.
- [203] Dan Roth and Wen-tau Yih. Global inference for entity and relation identification via a linear programming formulation. *Introduction to statistical relational learning*, pages 553–580, 2007.
- [204] Limin Yao, Sebastian Riedel, and Andrew McCallum. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 1013–1023, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [205] Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, pages 1–6, New York, NY, USA, 2013. ACM. doi: [10.1145/2509558.2509559](https://doi.org/10.1145/2509558.2509559).
- [206] Charles Sutton and Andrew McCallum. Joint Parsing and Semantic Role Labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning, CONLL '05*, pages 225–228, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
- [207] Hoifung Poon and Lucy Vanderwende. Joint inference for knowledge extraction from biomedical literature. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 813–821. Association for Computational Linguistics, 2010.
- [208] Sebastian Riedel and Andrew McCallum. Fast and robust joint models for biomedical event extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1–12. Association for Computational Linguistics, 2011.
- [209] Hoifung Poon and Pedro Domingos. Joint inference in information extraction. In *AAAI*, volume 7, pages 913–918, 2007.
- [210] Sameer Singh, Karl Schultz, and Andrew McCallum. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *Machine Learning and Knowledge Discovery in Databases*, pages 414–429, 2009.