Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Miha Bogataj

Odprta ekstrakcija informacij za slovenski jezik

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: doc. dr. Slavko Žitnik

Ljubljana, 2022

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Kandidat: Miha Bogataj
Naslov: Odprta ekstrakcija informacij za slovenski jezik
Vrsta naloge: Diplomska naloga na univerzitetnem programu prve stopnje Računalništvo in informatika
Mentor: doc. dr. Slavko Žitnik

Opis:
Odprta ekstrakcija informacij se ukvarja z nenadzorovano obdelavo besedil, pri čemer se na podlagi hevristik prepoznava trojice, ki lahko sestavljajo semantično mrežo. Kandidat naj pregleda obstoječe pristope odprte ekstrakcije informacij. Na podlagi pregleda naj kandidat izbere primeren pristop ali izdela lastno metodo za odprto ekstrakcijo informacij za slovenščino, ki bo prepoznavala trojice tipov osebek, povezava in predmet.

Title: Open information extraction for Slovenian language

Description:
Open information extraction deals with uncontrolled processing text, identifying triples that can be semantically composed based on heuristics. The candidate should review the existing approach to opening up information extraction. Based on the examination, the candidate should choose the appropriate approach or develop the latest method for open extraction of information for the Slovenian language, which has identified three types of subject, relation, and object.

Zahvaljujem se mentorju doc. dr. Slavku Žitniku za pomoč pri izdelavi diplomske naloge. Zahvaljujem se tudi svojim staršem za podporo pri študiju.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
RDF	Resource Description Framework	Okvir za opis virov
IE	Information Extraction	Ekstrakcija informacij
CLASSLA	CLARIN Knowledge Centre for South Slavic languages	CLARIN Center znanja za južno slovanske jezike
OLLIE	Open Language Learning for Information Extraction	Odprto učenje jezika za ekstrakcijo informacij
POS	Part-of-speech	Besedna vrsta
JSON	JavaScript Object Notation	JavaScript Objektna Notacija
SQL	Structured Query Language	Strukturirani povpraševalni jezik za delo s podatkovnimi bazami
NLP	Natural language processing	Procesiranje naravnega jezika
S	Subject	Osebek
V	Verb	Glagol
O	Object	Predmet
C	Complement	Komplement
A	Adverbial	Prislov

Povzetek

Naslov: Odprta ekstrakcija informacij za slovenski jezik

Avtor: Miha Bogataj

Odprta ekstrakcija informacij je proces procesiranja naravnega jezika, ki iz posameznih povedi izvleče možne odvisnosti. Odvisnosti so sestavljene iz semantične trojice, kjer prvi člen predstavlja subjekt o katerem poizvedujemo, relacije, ki opiše, kako se prvi člen navezuje na tretjega, in objekt. Sistem odprte ekstrakcije informacij za slovenščino temelji na metodi na podlagi pravil. Sistem je sestavljen iz predprocesorja in ekstraktorja. Vloga predprocesorja je obdelava vhodnega besedila s pomočjo sistema CLASSLA, ki slovnično analizira poved, lematizacija in izgradnja semantičnega drevesa. Vloga ekstraktorja je, da z uporabo pravil poišče relacije v povedi. Ta pravila so bolj kompleksna kot v angleščini, ker je v slovenščini besedni red bolj prost. Slovenščina pozna tudi več sklanjatev, ki omogočajo bolj točno določitev subjekta in objekta. Med najdenimi ekstrakcijami je možno iskanje na dva načina: iskanje povedi in dopolnjevanje parametrov. Iskanje povedi zahteva izpolnjene vse parametre semantične trojice in vrne seznam povedi, ki ustrezajo iskani semantični trojici. Dopolnjevanje parametrov zahteva dva izpolnjena parametra, od katerih je relacija obvezna. Ta način vrne seznam možnih vrednosti za manjkajoč parameter.

Ključne besede: ekstrakcija, informacija, slovenščina.

Abstract

Title: Open information extraction for the Slovenian language

Author: Miha Bogataj

Open information extraction is a process of natural language processing that extracts possible dependencies from individual sentences. Dependencies consist of a semantic triple where the first article represents the subject we inquire about, the relations that describe how the first article relates to the third, and the object. The open information extraction system for the Slovenian language is based on a rule-based method. The system consists of a preprocessor and extractor system. The role of the preprocessor is to process input text using the CLASSLA system which grammatically analyzes sentences, lemmatizes, and builds a semantic tree. The role of extractor is to find relationships in sentences using given rules. These rules are more complex than in English because in Slovenian the word order is freer. Slovenian also knows several declensions that enable a more precise definition of the subject and object. It is possible to search for found extractions in two ways: searching for sentences and supplementing the parameters. Sentence search requires that all parameters of the semantic triple are met and returns a list of sentences that match the semantic triple searched for. Complementing the parameters requires two met parameters of which the relation is mandatory. This method returns a list of possible values for the missing parameter.

Keywords: extraction, information, Slovenian language.

Poglavje 1

Uvod

Človeštvo je začelo zapisovati zgodovino že štiri tisočletja pred našim štetjem [9]. To je posledično povzročilo, da poznamo zelo veliko informacij o preteklosti iz vseh najdenih besedil. Zaradi izredno velikega nabora besedil, moramo ustvariti stroj, ki je zmožen razumeti ta besedila. Tukaj nam pomaga področje računalništva Obdelava naravnega jezika. Tako kot človek, se mora tudi računalnik naučiti razumeti besedilo, ki mu ga damo.

V zgodovini človeštva so se razvili mnogi jeziki, ki so med seboj bolj ali manj različni. Različni jeziki lahko uporabljajo različne pisave, različne strukture ali pa lahko izrazijo več/manj kot nek drug jezik. Dober primer različnih struktur je primerjava slovenščine in angleščine. Angleščina ima 12 časov, slovenščina pa 4 (od katerih se eden ne uporablja več). Slovenščina pozna sklanjatve samostalnikov in pridevnikov, Angleščina pa tega ne pozna. Zato je potrebno obravnavati mnoge jezike individualno.

Področje obdelave naravnega jezika se je začelo razvijati v petdesetih letih prejšnjega stoletja, ko je Alan Turing objavil članek z naslovom "Computing Machinery and Intelligence", kjer je predlagal to, kar danes imenujemo Turingov test. Razvoj področja se je nadaljeval s prvim prevajalnikom iz ruščine v angleščino leta 1954, ki je bil sposoben prevesti šestdeset ruskih stavkov v angleščino.

V devetdesetih letih prešnjega stoletja se je področje obdelave naravnega

jezika povsem spremenilo z uvedbo statističnih metod in strojnega učenja. V to kategorijo lahko uvrstimo tudi odprto ekstrakcijo informacij. Od leta 2013 je bil uveden povsem nov način obdelave naravnega jezika, ki temelji na nevronskih mrežah, kar je trenutno "state-of-the-art" na področju obdelave naravnega jezika.

Odprta ekstrakcija informacij je že dobro poznan postopek v mnogih jezikih, ampak v slovenščini je to še nepoznan postopek. V nadaljevanju bomo zato predstavili postopek odprte ekstrakcije informacij iz besedil v slovenskem jeziku.

Poglavje 2

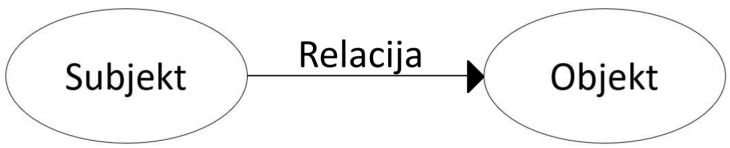
Odprta ekstrakcija informacij

2.1 Definicija problema

Problem odprte ekstrakcije informacij temelji na različnih metodah procesiranja naravnega jezika. Cilj naloge je izbrati primerno metodo za ekstrakcijo informacij v slovenščini in razvoj takega sistema, vključno z iskalnikom za poizvedbe med rezultati.

Kot rezultat ekstrakcije za posamezno poved z uporabo razvitega sistema želimo dobiti množico semantičnih trojic. Semantična trojica je trojica atomarnih entitet v RDF modelu (podatkovni model za metapodatke). Vsebuje tri entitete, ki predstavljajo semantiko povedi.

Format semantične trojice nam omogoča formalno predstavitev znanja v obliki, ki je berljiva stroju[15]. Semantična trojica je zapisana v formatu "(subjekt, relacija, objekt)".



Slika 2.1: Prikaz semantične trojice. Prikazuje, kako povežemo subjekt z objektom preko relacije.

Za ekstrakcije želimo, da bi vsebovale čim manj šumnih primerov. To preverimo tako, da uporabimo razvit iskalnik, s katerim najdemo in analiziramo reprezentativne primere. Zanima nas dopolnjevanje manjkajočih parametrov v semantični trojici, zato za posamezno poizvedbo pregledamo ustreznost dopolnjenih parametrov.

Odprta ekstrakcija informacij (angl. Open IE) je postopek pridobivanja trditev iz velikih korpusov brez potrebe po vnaprej določenem besedišču [5].

Edini vhodni podatek v sistem odprte IE je zbirka besedil, izhod pa je množica izvlečenih relacij. Odprti IE sistem gre čez podatke enkrat, kar omogoča skalabilnost sistema z velikostjo zbirke besedil.

2.2 Izzivi

2.2.1 Avtomatizacija

Sistemi odprte IE se morajo zanašati na nenadzorovane metode. To pomeni, da morajo biti namesto naštevanja ciljnih relacij vnaprej te samodejno zaznane med enim prehodom čez korpus. Ročno delo pri ustvarjanju množice rezultatov mora biti zmanjšano na minimum [12].

2.2.2 Heterogenost korpusa

Heterogenost korpusa je ovira za poglobljena lingvistična orodja, kot so sintaktični in odvisnostni razčlenjevalniki. Taki sistemi pogosto delujejo dobro, ko so učeni in uporabljeni na specifični domeni, ampak so nagnjeni k napakam, ko so uporabljeni v besedilih drugih vrst [12].

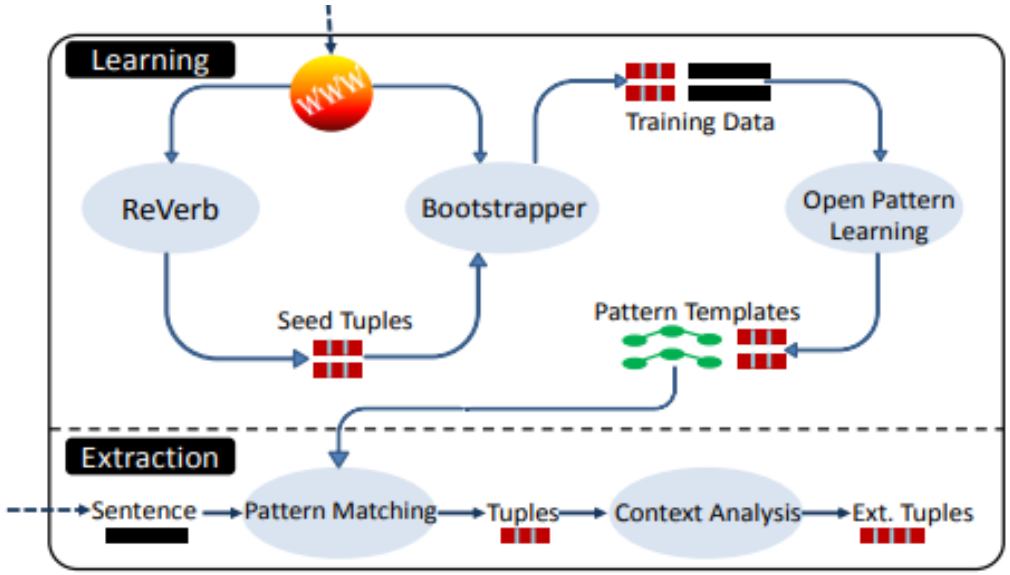
2.2.3 Učinkovitost

Algoritme odprte ekstakcije informacij je enostavno paralelizirati z metodo deli in vladaj, saj so povedi med seboj neodvisne. To se naredi tako, da se razdeli nabor povedi med več skupin in vse skupine sočasno izvajajo ekstrakcijo iz svojega nabora povedi.

2.3 Metode

2.3.1 Metode, ki temeljijo na znanju (prva generacija)

Metode, ki temeljijo na znanju, so metode prve generacije odprte ekstrakcije informacij, ki generirajo vzorce na podlagi učnih podatkov s pomočjo drevesa odvisnosti ali besednih vrst. Primer takega sistema sta TextRunner in OLLIE.



Slika 2.2: Sistemska arhitektura sistema OLLIE. Začne se s sejanjem N-terk iz sistema ReVerb, da ustvari učno množico, iz katere se uči vzorce.

Sistem OLLIE v prvem koraku ustvari množico N-teric nad korpusom z uporabo ReVerb-a. Ta množica je skupaj z originalnimi povedmi vhod v komponento Bootstrapper. Cilj te komponente je avtomatsko ustvariti veliko učno množico, ki vključuje množico različnih načinov, na katere je lahko informacija izražena v besedilu. Opazimo, da je lahko skoraj vsaka relacija prav tako izražena z ReVerb izrazom, ki temelji na glagolu.

Komponenta iz množice N-teric vzame posamezno N-terico, za katero nato poišče vse povedi iz korpusa, ki vsebujejo vse besede iz dane N-terice.

Bootstrapping hipoteza predpostavlja, da vse najdene povedi izražajo informacijo originalne N-terice [16]. Ta hipoteza ni vedno pravilna. Za zmanjšanje števila napak se dodajo dodatne omejitve na odvisnosti v povedih. Dovoljene so samo povedi, katerih vsebinske besede iz argumentov in relacije so med seboj povezane preko linearne poti do velikosti štiri v drevesu odvisnosti.

V naslednjem koraku se sistem iz učne množice mora naučiti vzorcev, ki kodirajo različne načine za izražanje informacij v povedih. Primer takih vzorcev je viden na sliki 2.3. Prepoznana pravila se lahko razlikujejo od korpusa do korpusa.

Extraction Template	**Open Pattern**
1. (arg1; be {rel} {prep}; arg2)	{arg1} ↑nsubjpass↑ {rel:postag=VBN} ↓{prep_*}↓ {arg2}
2. (arg1; {rel}; arg2)	{arg1} ↑nsubj↑ {rel:postag=VBD} ↓dobj↓ {arg2}
3. (arg1; be {rel} by; arg2)	{arg1} ↑nsubjpass↑ {rel:postag=VBN} ↓agent↓ {arg2}
4. (arg1; be {rel} of; arg2)	{rel:postag=NN;**type=Person**} ↑nn↑ {arg1} ↓nn↓ {arg2}
5. (arg1; be {rel} {prep}; arg2)	{arg1} ↑nsubjpass↑ {slot:postag=VBN;**lex** ∈**announce**|**name**|**choose...**} ↓dobj↓ {rel:postag=NN} ↓{prep_*}↓ {arg2}

Slika 2.3: Primer odprtih vzorcev. Prva tri pravila so popolnoma semantična; odebeljena so leksikalno omejena.

Odprti vzorci kodirajo vzorce, s katerimi se lahko relacija izraža v povedih. Za učenje teh vzorcev najprej izvlečemo odvisnostno pot, ki povezuje argumentne in relacijske besede za vsako N-terico in njej povezano poved. Označimo relacijsko vozlišče v odvisnostni poti s točno besedo (leksikalna omejitev) in besedno vrsto. Nato se to vozlišče še normalizira, kar pomeni, da se spremeni oblika glagola "biti" v obliko leme.

Če drevo odvisnosti vsebuje vozlišče, ki ni del vhodne N-terice, se to vozlišče imenuje "odprto vozlišče". Tako vozlišče ne razveljavi N-terice. Tako vozlišče se prav tako označi z besedno vrsto in leksikalno omejitvijo.

Nato se izvede sintaktično preverjanje. Ta preverjanja so omejitve, ki preverijo vsak kandidatni vzorec, da na poti odvisnosti ni odprtih vzorcev, da relacija leži med argumentoma, da se predlog na koncu vzorca ujema s predlogom v relaciji in da pot ne vsebuje odvisnosti "nn" ali "amod".

Taki vzorci so nato uporabljeni pri ekstrakciji informacij, da za posamezno poved pridobi množico možnih N-teric. Za izboljšavo točnosti se nad množico možnih N-teric izvede še kontekstno analizo. To se stori z uporabo razčlenjanja odvisnosti. Algoritem poišče povezave, ki vsebujejo odvisnost tipa vzorčno dopolnilo (ccomp), prislovni modifier (advcl) in jih leksikalno filtriramo [16].

2.3.2 Metode na podlagi pravil (druga generacija)

Metode na podlagi pravil so metode druge generacije odprte ekstrakcije informacij, ki temeljijo na ročno določenih hevurističnih pravilih, ki temeljijo na semantičnih lastnostih, kot so besedne vrste ali drevo odvisnosti. Primera takega sistema sta ClauseIE in ExtrHech [1].

Metode na podlagi pravil se delijo na dve vrsti:

- metode na podlagi pravil in plitke sintakse,
- metode na podlagi pravil in razčlembe odvisnosti.

Metode na podlagi pravil in plitke sintakse se zanašajo na leksiko-skladenjske vzorce in ročno sestavljena pravila glede na besedne vrste. Ta model pridobi odvisnosti glede na omejitve, kjer je vsaka relacija glagol ali glagolska fraza.

Metode na podlagi pravil in razčlembe odvisnosti uporabljajo ročno določena hevristična pravila, ki operirajo na razčlembi odvisnosti.

Primer takega sistema bo tudi sistem OpenIE za slovenščino, ki je bil razvit v okviru tega diplomskega dela. Sistem zaradi lastnosti slovenščine temelji na kombinaciji obeh vrst [1].

ClauseIE

Sistem ClauseIE za ekstrakcijo uporablja metodo, ki temelji na podlagi pravil in razčlembe odvisnosti [2].

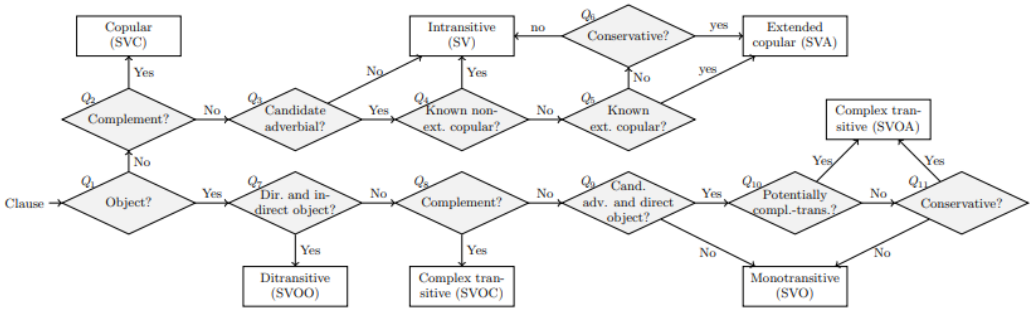
V prvem koraku sistem izvede razčlembo odvisnosti nad povedjo. To stori z uporabo Stanfordovega razčlenjevalnika odvisnosti [14]. S tem sistem

odkrije sintaktično strukturo vhodne povedi. Razčlenjene odvisnosti so sestavljene iz množice neposrednih sintaktičnih relacij med besedami v povedi. Koren odvisnosti je nepovezani glagol ali predmetni dodatek kopularnega glagola.

V naslednjem koraku se identificira stavke v vhodni povedi. S tem želimo pridobiti glavno besedo iz vseh posameznih stavkov. Najprej sestavimo stavke za vsak osebek v odvisnostni strukturi (primer nsubj). Odvisni člen je sestavljen iz osebka (S) in vodilnega glagola (V). Vse ostale zloženke v stavku so odvisne od glagola: predmeti (O), komplementi (C) in prislovi (A).

Za izboljšanje deleža pridobljenih in informativnih ekstrakcij ClauseIE ustvari še nekaj "sintetičnih stavkov". To so stavki, ki se sami ne pojavijo v povedi, a so obravnavani identično kot stavki v povedi. Sintetični stavki so v določeni meri uporabljeni pri obravnavi ekstrakcij, ki ne vsebujejo vodilnega glagola (primer apozicija ali posesiv).

V tretjem koraku iz pridobljenih stavkov sistem poskuša identificirati tip vsakega posameznega stavka. To stori z uporabo znanja o lastnosti glagola v stavku in znanja o strukturi vhodnega stavka. Opis, kako poteka prepoznavanje tipa stavka, je razvidn na sliki 2.4.



Slika 2.4: Odločitveno drevo za prepoznavanje tipa stavka.

Posledica ločitve stavkov in ločitev zaznavanja tipa stavkov od generiranja predloga je v tem, da je slednje fleksibilno in je mogoče urejati za posamezne vrste uporabe. Generiranje predloga je sestavljeno iz dveh korakov. Prvi

korak je odločitev, katere zloženke tvorijo predlog, drugi korak pa generiranje predloga iz izbranih zloženk.

Po identifikaciji tipa stavka sistem uporabi temu ustrezna pravila. Ta pravila so vidna na sliki 2.5.

	Pattern	Clause type	Example	Derived clauses
			Basic patterns	
S_1:	SV_i	SV	AE died.	(AE, died)
S_2:	SV_eA	SVA	AE remained in Princeton.	(AE, remained, in Princeton)
S_3:	SV_cC	SVC	AE is smart.	(AE, is, smart)
S_4:	$SV_{mt}O$	SVO	AE has won the Nobel Prize.	(AE, has won, the Nobel Prize)
S_5:	$SV_{dt}O_iO$	SVOO	RSAS gave AE the Nobel Prize.	(RSAS, gave, AE, the Nobel Prize)
S_6:	$SV_{ct}OA$	SVOA	The doorman showed AE to his office.	(The doorman, showed, AE, to his office)
S_7:	$SV_{ct}OC$	SVOC	AE declared the meeting open.	(AE, declared, the meeting, open)
			Some extended patterns	
S_8:	SV_iAA	SV	AE died in Princeton in 1955.	(AE, died) (AE, died, in Princeton) (AE, died, in 1955) (AE, died, in Princeton, in 1955)
S_9:	SV_eAA	SVA	AE remained in Princeton until his death.	(AE, remained, in Princeton) (AE, remained, in Princeton, until his death)
S_{10}:	SV_cCA	SVC	AE is a scientist of the 20th century.	(AE, is, a scientist) (AE, is, a scientist, of the 20th century)
S_{11}:	$SV_{mt}OA$	SVO	AE has won the Nobel Prize in 1921.	(AE, has won, the Nobel Prize) (AE, has won, the Nobel Prize, in 1921)
S_{12}:	$ASV_{mt}O$	SVO	In 1921, AE has won the Nobel Prize.	(AE, has won, the Nobel Prize) (AE, has won, the Nobel Prize, in 1921)

S: Subject, V: Verb, C: Complement, O: Direct object, O_i: Indirect object, A: Adverbial, V_i: Intransitive verb, V_c: Copular verb, V_e: Extended-copular verb, V_{mt}: Monotransitive verb, V_{dt}: Ditransitive verb, V_{ct}: Complex-transitive verb

Slika 2.5: Pravila za posamezen tip stavka v sistemu ClauseIE.

ClauseIE generira en predlog za vsako izbrano podmnožico zloženk. Da generira predlog, mora sistem določiti, kateri del vsake zloženke postavi v subjekt, katerega v relacijo in katerega v objekt. Subjekt se preslika v subjekt, glagol pa se preslika v relacijo. Ob ekstrakciji se ustvari še objekt iz preostalih zloženk. Za ekstrakcijo trojice se združijo vsi objekti.

Poglavje 3

Implementacija

3.1 Pregled

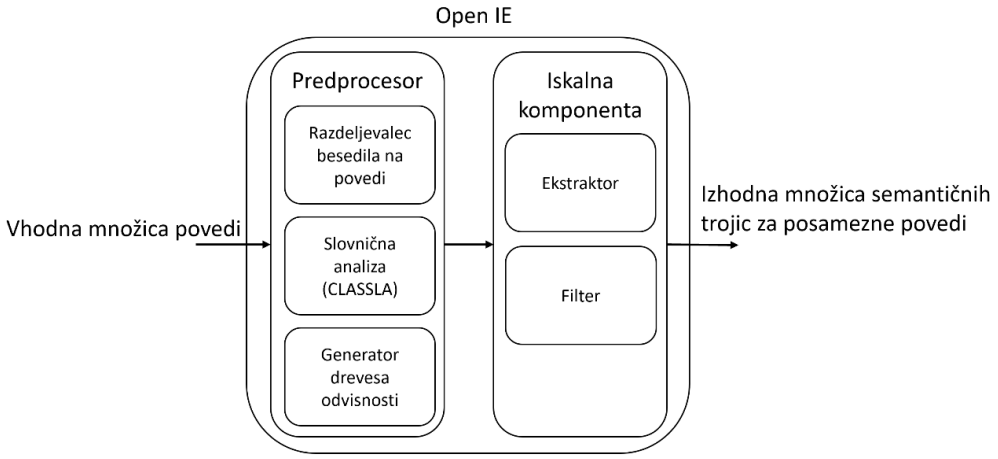
V splošnem je implementacija ekstrakcije sestavljena iz naslednjih komponent:

- predprocesor,
- iskalna komponenta,
- učna komponenta [17].

Ker implementiramo odprto ekstrakcijo informacij z metodo na podlagi pravil (druga generacija), učne komponente ne potrebujemo. Arhitektura razvitega sistema je prikazana na sliki 3.1.

3.2 Predprocesor

Naloga predprocesorja je obdelava vhodnega besedila pred ekstrakcijo informacij. V postopek obdelave spada razdelitev besedila na ločene povedi, označevanje dela govora, lematizacija in generiranje drevesa odvisnosti. Za označevanje dela govora, lematizacijo in iskanje odvisnosti v povedi bo uporabljen sistem CLASSLA z nestandardnim modelom za slovenščino.



Slika 3.1: Arhitektura razvitega sistema. Sistem vsebuje dve glavni komponenti – predprocesor in iskalno komponento.

3.2.1 CLASSLA

Sistem CLASSLA temelji na Stanfordovem sistemu Stanza. To sta sistema za analizo naravnega jezika. Uporabljata se za pretvorbo teksta v človeškem jeziku v seznam besed z njihovimi formalnimi lastnostmi.

Sistema Stanza in CLASSLA temeljita principu nevronskih mrež komponent, ki omogočajo učinkovito učenje in vrednotenje z označenimi povedmi [11, 14].

Pri implementaciji je bil uporabljen sistem CLASSLA z modelom za nestandardno slovenščino, ker v primeru slogovno zaznamovane povedi standardni model ne vrne pravilnih rezultatov. Uporabljeni korpus vsebuje besedila različnih virov, zato lahko vsebuje tudi besedila z nestandardno slovenščino. Zato je bil izbrani model nestandarden. CLASSLA je sestavljena iz več enot, imenovanih procesor: tokenizer, označevalec dela govora, lema, razčlenjevalec odvisnosti relacij, večbesedni tokenizer, identifikator imenskih entitet, identifikator čustev.

Za potrebe ekstrakcije so uporabljeni procesorji tokanizer, označevalec dela govora, lema in razčlenjevalec odvisnosti relacij.

Kot rezultat nam sistem CLASSLA vrne JSON dokument.

Primer

Na dani povedi zaženemo opisani postopek: "To velja posebno za območje severovzhodne Severne Amerike, kjer se je intenzivno zbiranje začelo dolgo pred prihodom prvih poklicnih etnografov." Rezultat postopka je naslednji:

```
[
    {
        "id":1,
        "text":"Bilo",
        "lemma":"biti",
        "upos":"AUX",
        "xpos":"Va-p-sn",
        "feats":"Gender=Neut|Number=Sing|...",
        "head":3,
        "deprel":"cop"
    }, {
        "id":2,
        "text":"je",
        "lemma":"biti",
        "upos":"AUX",
        "xpos":"Va-r3s-n",
        "feats":"Mood=Ind|Number=Sing|Person=3|...",
        "head":3,
        "deprel":"aux"
    }, ...
]
```

Rezultat postopka je datoteka tipa JSON, ki vsebuje tabelo besed z opisom slovničnih lastnosti.

3.2.2 Generiranje drevesa odvisnosti

Drevo odvisnosti je usmerjeno drevo iz listov drevesa proti korenu, kjer ima vsak element poljubno število sinov. Formalno je definiran kot:

$$G = (V, E)$$

V je definiran kot množica vozlišč (vertex).
E je množica povezav, ki je v splošnem definirana naslednje:

$$E \subseteq \{(x, y) | (x, y) \in V^2 \wedge x \neq y \wedge y \, je \, starš \, od \, x\}$$

Po pridobitvi slovničnih lastnosti za posamezne besede v dani povedi, je mogoče generirati drevo odvisnosti. Izgradnja drevesa je sestavljena iz naslednjih postopkov:

- povezovanje vozlišč,
- minimizacija drevesa in združevanje vozlišč.

Povezovanje vozlišč

V drevesu odvisnosti vsaka beseda predstavlja svoje vozlišče, ki vsebuje rezultat sistema CLASSLA za dano besedo.

Povezavo v drevesu odvisnosti lahko definiramo tako:

$$e = \{e \in E \wedge (x, y) \in V^2 \wedge head \in x \wedge id \in y | e \in E \wedge head = id\}$$

Pri povezovanju vozlišč sta ključna parametra "id" in "head". Vrednost parametra "head" za neko besedo kaže na drugo besedo, ki ima parameter "id" z isto vrednostjo. Za vsa vozlišča ponovimo ta proces, da dobimo rezultat, ki je usmerjeno drevo v smeri od listov proti korenu.[17]

Minimizacija drevesa in združevanje vozlišč

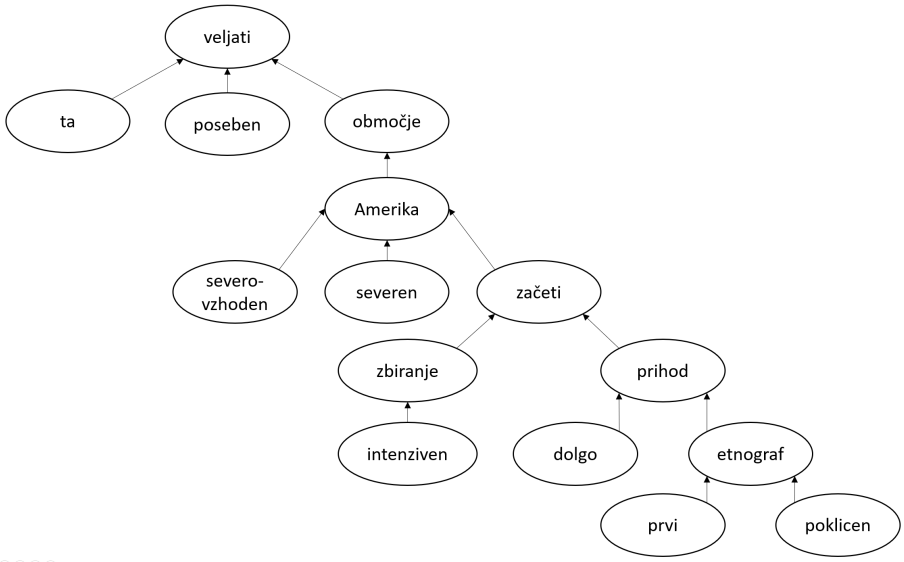
Cilj minimizacije in združevanja vozlišč je zmanjševanje kompleksnosti drevesa in posledično tudi rezultata.

V procesu minimizacije se odstrani vozlišča z besedami, ki so prisotne v povedih zaradi sintaktičnih razlogov ali pa ne nosijo nobene informacije. To so predvsem mašila ali prislovi.

V procesu združevanja vozlišč se združijo vozlišča, ki samostojno ne izražajo ničesar, skupaj s staršem pa izražajo neko mnenje. To so besedne vrste, kot so členek in pomožni glagoli.

Primer

Za poved: "To velja posebno za območje severovzhodne Severne Amerike, kjer se je intenzivno zbiranje začelo dolgo pred prihodom prvih poklicnih etnografov," se generira semantično drevo, kot je vidno na sliki 3.2:



Slika 3.2: Prikaz generiranega drevesa odvisnosti.

V semantičnem drevesu elipse (kot so razvidne na sliki 3.2) predstavljajo posamezno vozlišče, ki vsebuje besedo (in lemo) in vse prepoznane slovnične lastnosti za to besedo. Puščice med vozlišči predstavljajo povezavo v semantičnem drevesu in vedno kažejo v smer proti korenu (najvišje vozlišče). Vsako vozlišče ima lahko 0 ali več sinov in največ enega starša.

3.3 Ekstrakcija informacij

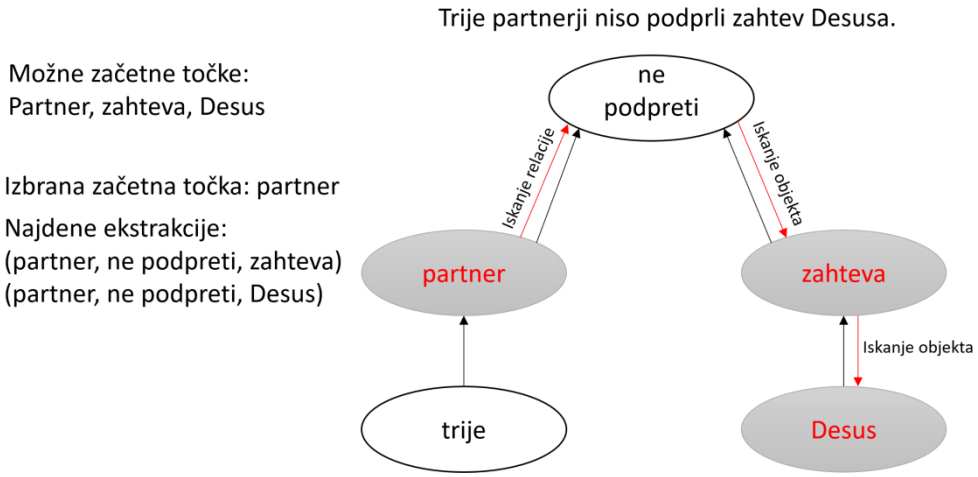
Ekstrakcija informacij je osrednji proces v sistemu. Kot vhod algoritem dobi drevo odvisnosti, ki vsebuje informacije o slovnici in odvisnostih za posamezne besede, vrne pa seznam ustreznih semantičnih trojic. To lahko definiramo kot funkcijo:

$$f : G(V, E) \rightarrow T$$

T je definiran kot množica ustreznih semantičnih trojic za drevo odvisnosti.

Algoritem za to funkcijo (prikazan na sliki 3.3) je naslednji:

1. Iz vozlišč v drevesu poišči ustrezne začetne točke. To so samostalniki in zaimki.
2. Za vsako začetno točno začni ekstrakcijo:
   (a) Iščemo prvi člen v trojici (subjekt). Pomikamo se po drevesu navzgor proti korenu, dokler ne najdemo ustreznega člena. To so samostalniki in zaimki.
   (b) Od najdenega člena se premikamo navzgor proti korenu, dokler ne najdemo ustreznega glagola. To je drugi člen v trojici (relacija).
   (c) Od najdenega člena imamo več možnosti za iskanje tretjega člena (objekt). Iščemo en ali več objektov. Možni objekt je neposredni naslednik trenutnega vozlišča in vsi predniki z izjemo veje, v kateri je subjekt.
3. Za vse dobljene semantične trojice zaženi postopek filtracije:
   (a) Preverimo odvisnosti. Če odvisnost specifično navaja, da je nek člen objekt ali subjekt, elementa ustrezno zamenjamo.
   (b) Preverimo sklone. Skloni so od imenovalnika do orodnika, po vrsti naraščujoče oštevilčeni. Če vrednost sklona subjekta ni manjša ali enaka vrednosti sklona objekta, se trojica odstrani.
   (c) Odstranimo ponavljajoče se trojice.



Slika 3.3: Prikaz poteka algoritma na semantičnem drevesu za začetno točko "partner". Rdeče puščice predstavljajo potek algoritma, črne puščice odvisnosti v drevesu, sivo ozadje možne začetne točke in rdeč tekst izbrana vozlišča za rezultat.

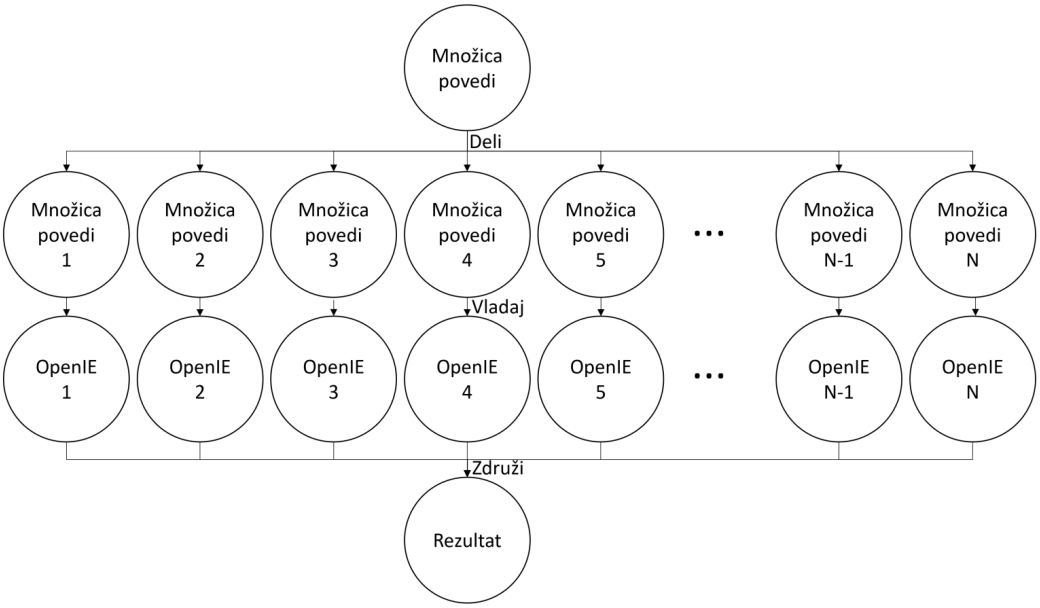
3.4 Paralelizacija rešitve

Odprta ekstrakcija informacij je časovno potraten postopek. V razvitem sistemu je najbolj potratni del analize naravnega jezika s sistemom CLASSLA. Za eno poved ta postopek lahko traja tudi več kot sekundo.

Algoritem za odprto ekstrakcijo informacij iz nekega korpusa je enostavno paralelizirati, ker so posamezne povedi med seboj neodvisne. Ideja za paralelizacijo algoritma temelji na metodi "Deli in vladaj" [10].

Metoda deli in vladaj predstavlja strategijo delitve problema na več manjših problemov enake vrste. Metoda je sestavljena iz treh korakov:

1. Deli.
2. Vladaj.
3. Združi.



Slika 3.4: Shema metode deli in vladaj na problemu odprte ekstrakcije informacij.

V koraku "deli" razdelimo množico povedi med poljubno število niti. Vse niti nato zaženemo, da tečejo paralelno.

Korak "vladaj" je odrednji korak metode. V tem koraku se izvaja algoritem za posamezen reduciran problem. Pri odprti ekstrakciji informacij je to glavni del, ki ga sestavljajo analiza naravnega jezika, sestavljanje semantičnega drevesa in ekstrakcija informacij.

Združevanje v tem sistemu poteka na nivoju podatkovne baze. Ko neka nit pridobi rezultat, ga takoj shrani v podatkovno bazo. S tem delno podaljša čas trajanje izvedbe z dodatno vhodno-izhodno operacijo, a hkrati tudi zmanjša potrebo po pomnilniku.

3.5 Hramba izvlečenih semantičnih trojic

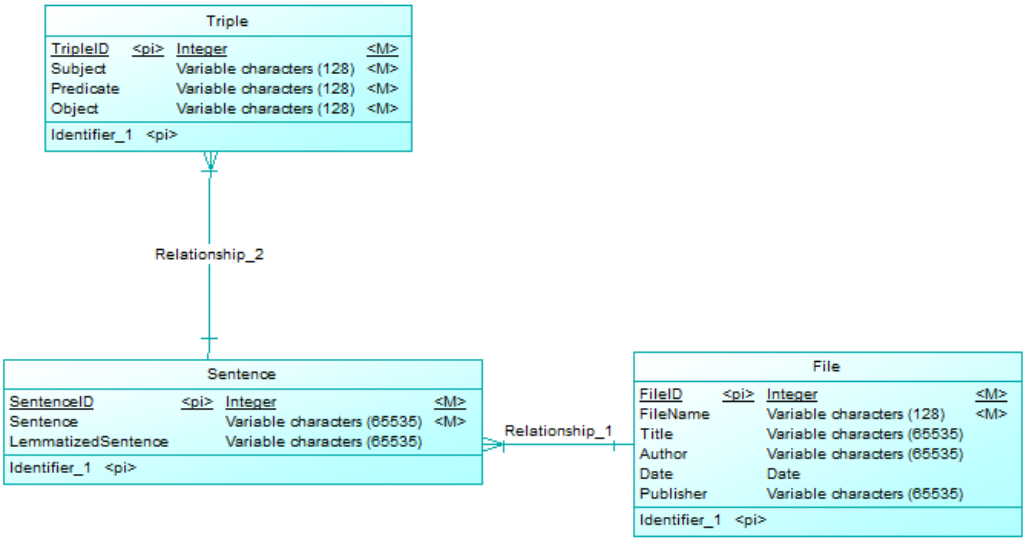
Rezultat sistema ekstrakcije informacij je seznam semantičnih trojic za posamezno poved in dokument. Zato si moramo v podatkovni bazi shraniti pet

različnih vrednosti – semantično trojico, poved in dokument, kot je vidno v konceptualnem modelu na sliki 3.5. Podatkovna baza v tretji normalni obliki zato vsebuje tri entitete, ki so med seboj povezane s tujimi ključi. To nam omogoča enostavno poizvedovanje z minimalnim podvajanjem podatkov znotraj podatkovne baze.

Uporabljen sistem za podatkovne baze je PostgreSQL, saj nam omogoča deklarativno poizvedovanje po vsebini baze, kar poenostavi iskalnik.

3.6 Iskanje

Rezultat ekstrakcije za posamezno poved (množica semantičnih trojic) se shrani v enostavno podatkovno tabelo.



Slika 3.5: Konceptualni model podatkovne baze.

Iskalnik za iskanje rezultatov v množici uporablja parametre semantičnih trojic. Ta postopek lahko poteka na dva različna načina.

Iskanje povedi

Postopek iskanja povedi zahteva izpolnjene vse tri parametre semantične trojice. V podatkovni bazi nato poišče vse povedi, ki imajo kot enega od rezultatov podano semantično trojico.

Dopolnjevanje parametrov

Postopek dopolnjevanja parametrov zahteva dva izpolnjena pogoja, od katerih je relacija vedno obvezna. Iskalnik sprejme dva parametra in na podlagi njiju poišče v bazi vse ustrezne semantične trojice. Nato združi pridobljene rezultate po povedih. Za manjkajoči člen se nato sestavo besedna zveza iz - v iskalniku - manjkajočega člena v vseh semantičnih trojicah za posamezno poved.

Poglavje 4

Primerjava odprtega sistema IE za slovenščino s sistemi za odprto IE za angleščino

4.1 Primerjava slovenščine in angleščine

Slovenščina je pogosto pisana v nestandardni slovnični obliki in z nestandardnim besediščem. Vzroki za to so lahko narečje, sleng ali pa slovnične napake. V primerjavi z angleščino, kjer je besedni red natančno določen s slovnico, je v slovenščini ta bolj prost. Vrstni red besed v slovenščini izraža zaznamovanost stavka.

4.1.1 Sklanjatve

Sklanjatve so kategorije samostalnikov in samostalniških modifikatorjev, ki ustrezajo različnim slovničnim funkcijam. Angleščina je večinoma izgubila sistem sklanjatev. Preostanek sklanjatev obstaja le še za osebne zaimke, kjer jezik pozna tri sklone (imenovalnik, casus obliquus, posesiv) [8]. Slovenščina pozna šest sklonov (imenovalnik, rodilnik, dajalnik, tožilnik, mestnik, orodnik). Ker nam skloni opisujejo vrsto konteksta (na primer, mestnik govori o nečem), lahko uporabimo to znanje za filtriranje rezultatov ekstrakcije.

Pravila za filtracijo

Filtracija ekstrakcij sledi naslednjim pravilom:

- Če je objekt v imenovalniku, mora biti v imenovalniku tudi subjekt. To velja, ker imenovalnik v objektu pomeni, da imamo informacijo kdo ali kaj nekaj ali nekdo je.
- Tožilnik je lahko samo v objektu [7], ker v slovenščini tožilnik lahko izraža le predmet.

4.1.2 Odvisnost relacij

Odvisnost relacij opisuje, kako neka beseda vpliva na besedo, na katero kaže v semantičnem drevesu. Možnih vrst relacij je več kot 63 [3]. Te odvisnosti se lahko uporabljajo pri odločitvi, če je neka beseda lahko subjekt ali objekt.

Me pomembnejšimi odvisnostmi so vezniki in različni modifikatorji, saj le-ti samostojno ne nosijo informacijo (na primer negacija z besedo ne, pridevniki...).

Z uporabo te lastnosti besede lahko minimiziramo semantično drevo (na primer mašila) in odločamo, ali je nek člen primeren za subjekt ali objekt.

4.2 ReVerb

ReVerb [6] je sistem odprte ekstrakcie informacij druge generacije. Sistem temelji na ročno določenih pravilih in plitki sintaksi [16]. Cilj sistema ReVerb je bil razrešitev treh pogostih napak:

1. Rezultati so pogosto vsebovali neinformativne rezultate.
2. Neskladne ekstrakcije.
3. Prekomerno specifične relacije, ki vsebujejo preveč informacij, da bi bile uporabne pri semantičnih nalogah [12].

ReVerb je izboljšan na področju teh napak zaradi uporabe semantičnih omejitev. Te semantične omejitve so izražene kot enostavni regularni izraz besednih vrst [12].

```
V | VP | VW*P
V = glagolski členek? prislov
W = (samostalnik |pridevnik | prislov | zaimek | determiner)
P = (predlog | členek | infinitivna oznaka)
```

Slika 4.1: Regularni izraz semantičnih omejitev sistema ReVerb.

Sistem ReVerb išče ekstrakcije tako, da najprej identificira relacijske besede ali besedne zveze[5]. To naredi tako, da poišče vse fraze, ki ustrezajo določenim pravilom. Ta pravila so naslednja:

1. Fraza se začne z glagolom.

2. Fraza ustreza sintaktičnim omejitvam, kar pomeni, da preveri, ali fraza ustreza regularnemu izrazu za relacijo iz slike 4.1.

3. Fraza ustreza leksikalnim omejitvam, kar stori tako, da uporabi večji slovar relacijskih fraz, ki lahko sprejmejo različne argumente. V naslednjem koraku izgradi slovar s postopkom iskanja vseh besed ustreznih besednih vrst. Nato se množica normalizira z odstranitvijo slovnične zaznamovanosti, pomožnih glagolov, pridevnikov in prislovov.

4. Če so pari fraz sosednji ali se prekrivajo, se združijo.

Ekstrakcija argumenta poteka v naslednjih korakih:

1. Za subjekt poišče najbližjo samostalniško frazo levo od relacije, ki ni relativni zaimek, “kdo” prislov ali eksistencialni “tukaj”.

2. Za objekt poišče najbližjo samostalniško frazo desno od relacije.

4.3 TextRunner

TextRunner [16] je sistem odprte ekstrakcije informacij prve generacije (samonadzorovana učna metoda [12]). Ta sistem deluje po principu iskanja vzorcev v besedilu.

Učna komponenta sistema deluje tako, da označi lastne učne primere kot pravilne ali nepravilne [4]. Z označenimi učnimi primeri nato nauči naivni Bayesov klasifikator, ki je nato uporabljen v ekstraktorju. Za generacijo učnih primerov so uporabljeni sistemi druge generacije (primer ReVerb [5]).

Ekstraktor z enim obhodom čez korpus izvleče n-terice za vse možne relacije. To analizo izvede z uporabo modelov z maksimalno entropijo za označevanje POS in iskanje samostalniških fraz. Za to izvedbo uporablja orodje "OpenNLP toolkit"[4].

4.4 Odprta ekstrakcija informacij za slovenščino

4.4.1 Splošna pravila

Pravila za slovenščino so v primerjavi z angleščino precej podobna, vendar pa zaradi poljubnega vrstnega reda besed v slovenščini ne združujemo več besed v en člen semantične trojice (razen členka v relaciji – primer negacija). Ta semantična pravila lahko izrazimo kot naslednji izraz:

$$S|R|O$$

$$S = (samostalnik|zaimek|deprel_{subject})$$

$$R = ((členek^* + glagol)|(pomožniGlagol + (samostalnik|prislov|zaimek)))$$

$$O = (deprel_{object}|samostalnik|determiner|prislov|zaimek|število)$$

4.4.2 Subjekt

Za razliko od sistema ReVerb v odprti ekstrakciji informacij za slovenščino ne začnemo z iskanjem relacij, ampak najprej poiščemo vse možne kandidate za

subjekt. To naredimo tako, da iz množice vseh besed poiščemo vse kandidate za subjekt, ki ustrezajo zgornjemu regularnemu izrazu. Ni potrebno, da so ti kandidati listi v semantičnemu drevesu, ampak so lahko na poljubnemu mestu. S subjektom začnemo, ker je v slovenščini besedni red v stavku precej prost.

4.4.3 Relacija

Po najdenem subjektu iščemo relacijo. Pri tem postopku je pravilo, da iščemo samo najbližji glagol. Če glagol (v smeri navzgor proti korenu) ni najbližji, ta glagol pripada nadrednemu stavku v povedi, zato tak glagol ni ustrezen. Glagol prav tako ni ustrezen, če je iskan v smeri od korena do lista, saj tak glagol pripada sorednemu ali podrednemu stavku v povedi.

Za razliko od sistema ReVerba je izjemoma lahko relacija samostalniška ali pridevniška beseda, ki vsebuje pomožni glagol, vendar pa odvisnost te relacije ne sme biti modifikator ali pa subjekt. Modifikator ne sme biti, ker taka samostalniška beseda ne izraža popolne informacije; subjekt ne sme biti, saj je taka beseda že v členu subjekt v semantični trojici.

4.4.4 Objekt

Po najdeni relaciji iščemo množico možnih objektov. Objekt iščemo v členu, ki je za en korak bližje korena od relacije ali v vejah sinov relacije (razen v veji, v kateri je subjekt). Znotraj veje se omejimo na iskanje objektov samo v istem stavku neke povedi. To ločujemo tako, da ustavimo iskanje v veji, če najdemo besedo besedne vrste glagol ali odvisnostno relacilo marker (ki, če, pa...).

Marker je vrsta odvisnosti, ki razločuje med stavkom in njem podrednim stavkom. Taka beseda je v slovenščini vedno na prvem mestu v podrednem stavku.

Beseda je primerna za objekt, če je odvisnost relacije object ali pa je samostalnik, determiner, prislov ali zaimek [13].

Poglavje 5

Evolucija in diskusija

5.1 Število ekstrakcij

Za milijon povedi sistem vrne 9.179.345 semantičnih trojic, kar pomeni približno 4,71 semantičnih trojic na poved. Število ekstrakcij v posamezni povedi je odvisno od vsebine, in ni navzgor omejeno. Robni primer povedi, ki vrne večje število ekstrakcij (1692), je naslednji:

"SZF so z donacijami podprli naslednji posamezniki : Lidija Andolšek Jeras, Zoran Marij Arnež, Alenka Aškerc Mikeln, Darja Barlič Magajna, Andrej Bauer, Ana Benedetič, Anton Beovič, Anton Bergant, Mitja Bernik, Borut Božič, Aleš Bulc, Violeta Bulc..."

Zgornja poved vrne večjo množico semantičnih trojic v obliki

$$(ime/priimek, podpreti, SZF)$$

$$(ime/priimek, podpreti, donacija)$$

$$(ime/priimek, podpreti, Jeras)$$

$$(posameznik, podpreti, ime/priimek)$$

$$(SZF, podpreti, donacija)$$

Vendar večje število semantičnih trojic pride zaradi napačnega prepoznavanja imen ali priimkov (Arnež, Jeras, Marij) kot tuje besede pri sistemu

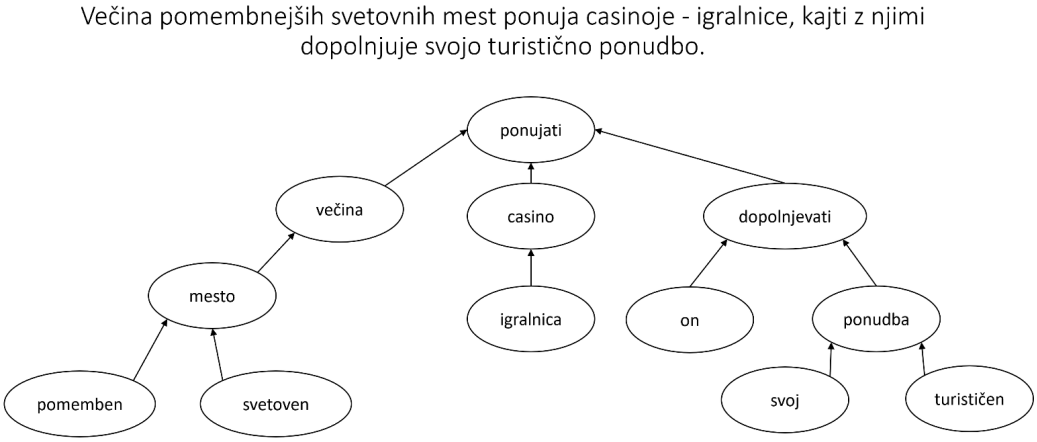
CLASSLA. To onemogoči generiranje ustreznega semantičnega drevesa. Taka poved kljub temu vrne več sto pravilnih semantičnih trojic, kar je rezultat naštevanja imen, kar povzroči večje število subjektov.

5.2 Analiza ekstrakcij

Primer 1

Kot primer vzamemo naslednjo poved: "Večina pomembnejših svetovnih mest ponuja casinoje - igralnice, kajti z njimi dopolnjuje svojo turistično ponudbo."

Ta poved je izbrana, ker vsebuje več stavkov, med katerimi je podredni odnos; je slovnično pravilna in ima bolj standardno obliko. Ta poved ne vsebuje posebnosti. Dana poved generira naslednje semantično drevo kot je razvidno na sliki 5.1.



Slika 5.1: Semantično drevo za dano poved.

Ker je poved slovnično pravilna in brez posebnosti, dobimo pravilno semantično drevo. Podredje je jasno vidno pri povezavi *dopolnjevati* → *ponujati*.

Z ekstrakcijo informacij dobimo naslednji rezultat:

$$1.(\textit{večina}, \textit{ponujati}, \textit{casino})$$

$$2.(\textit{mesto}, \textit{ponujati}, \textit{casino})$$

$$3.(\textit{on}, \textit{dopolnjevati}, \textit{svoj})$$

$$4.(\textit{on}, \textit{dopolnjevati}, \textit{ponudba})$$

Pridobljene rezultate lahko združimo po relaciji, s čimer pridobimo končne informacije:

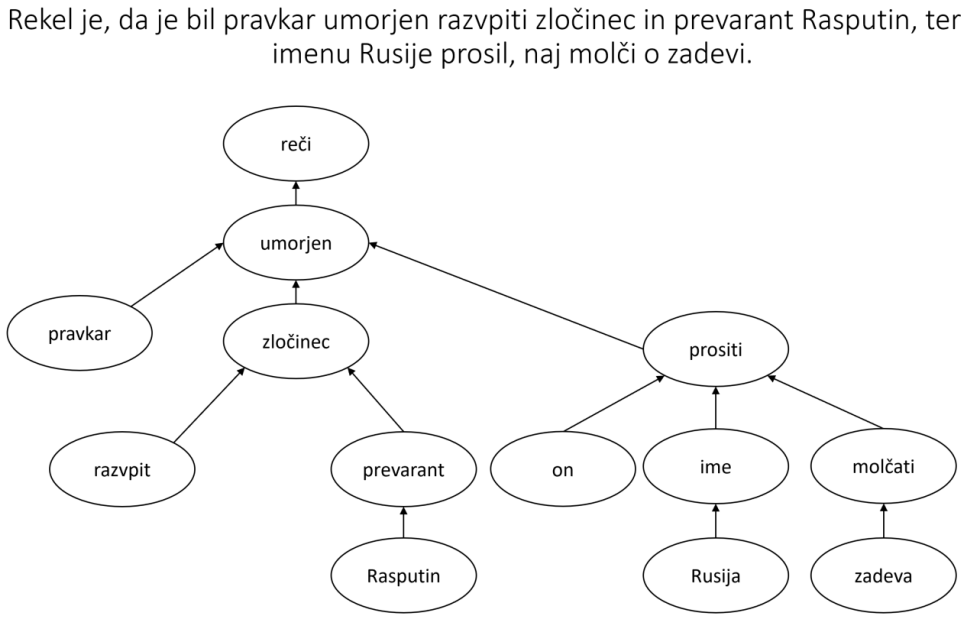
$$1.(\textit{večina mesto}, \textit{ponujati}, \textit{casino})$$

$$2.(\textit{on}, \textit{dopolnjevati}, \textit{svoj ponudba})$$

Primer 2

Kot primer vzamemo naslednjo poved: "Rekel je, da je bil pravkar umorjen razvpiti zločinec in prevarant Rasputin, ter ga v imenu Rusije prosil, naj molči o zadevi."

Ta poved je bila izbrana, ker vsebuje tako priredja kot podredja. Primer priredne odvisnosti stavkov v povedi je "rekel je" in "naj molči o zadevi". Primer podredne odvisnosti je stavek "da je bil pravkar umorjen razvpiti zločinec in prevarant Rasputin," ki je podredno odvisen od stavka "rekel je". Dana poved generira naslednje semantično drevo, kot je razvidno na sliki 5.2.



Slika 5.2: Semantično drevo za dano poved.

Z ekstrakcijo informacij dobimo naslednji rezultat:

$$1.(ime, prositi, on)$$

$$2.(Rusija, prositi, on)$$

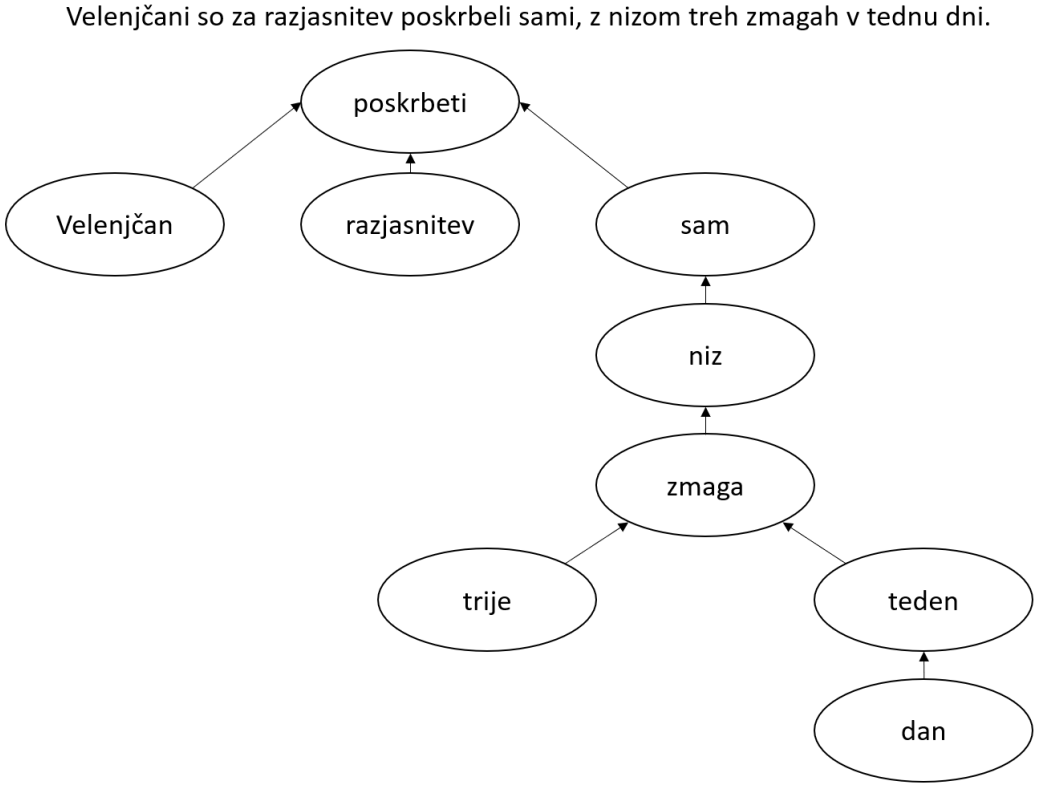
Stavka "rekel je" in "naj molči o zadevi" ne vsebujeta kandidatov za ekstrakcijo, zato taka stavka ne vrneta rezultatov. Vzrok za to je v tem, da

imajo po minimizaciji semantičnega drevesa, taki stavki kot rezultat manj kot tri vozlišča. Če imamo manj kot tri vozlišča, je nemogoče izbrati vsaj en subjekt, relacijo in objekt, saj morajo biti elementi med seboj različni.

Primer 3

Kot primer vzamemo naslednjo poved: "Velenjčani so za razjasnitev poskrbeli sami, z nizom treh zmagah v tednu dni."

Ta poved je bila izbrana, ker vsebuje slovnično napako. Slovnična napaka v povedi je napačno postavljena vejica. Dana poved generira naslednje semantično drevo, kot je razvidno na sliki 5.3.



Slika 5.3: Semantično drevo za dano poved.

V povedi vejica tipično označuje ločilo med stavki ali naštevanje. V tej povedi ni naštevanja ali več stavkov. Kljub napačni postavitvi vejice je sistem pravilno prepoznal stavke in izgradil ustrezno semantično drevo.

Z ekstrakcijo informacij dobimo naslednji rezultat:

$$1.(Velenjčan, poskrbeti, razjasnitev)$$

$$2.(Velenjčan, poskrbeti, niz)$$

$$3.(Velenjčan, poskrbeti, zmaga)$$

$$4.(Velenjčan, poskrbeti, trije)$$

$$5.(Velenjčan, poskrbeti, teden)$$

$$6.(Velenjčan, poskrbeti, dan)$$

$$7.(niz, poskrbeti, razjasnitev)$$

$$8.(teden, poskrbeti, razjasnitev)$$

$$9.(zmaga, poskrbeti, razjasnitev)$$

$$10.(dan, poskrbeti, razjasnitev)$$

Beseda razjasnitev se v rezultatih večkrat pojavi kot objekt. Beseda razjasnitev je v povedi v tožilniku, zato je vedno objekt. Besede, kot so niz, zmaga, teden, dan, se lahko pojavijo kot subjekt ali objekt, vendar v različnem kontekstu (primer: za kaj so poskrbeli Velenjčani in z nizom so poskrbeli za kaj?).

5.3 Analiza poizvedb

Za analizo rezultatov uporabimo iskalnik, ki je bil razvit v okviru sistema. Rezultati so možni na dva načina:

1. Katerim povedim ustreza neka semantična trojica?

2. S čim lahko dopolnemo semantično trojico?

5.3.1 Katerim povedim ustreza semantična trojica

Ko iščemo, katerim povedim ustreza neka semantična trojica, nas zanima, v katerem kontekstu se neka informacija nahaja. V tem primeru zato vrnemo vse ustrezne povedi, ki jim ustreza iskana semantična trojica.

Kot primer vzamemo semantično trojico (škoda, nastati, požar). Torej nas zanima vse glede nastale škode v požaru. Za rezultat želimo dobiti množico povedi, ki vsebujejo podatek o škodi, ki nastane pri požaru. Kot rezultat za tako poizvedbo dobimo naslednje povedi:
14 rezultatov

- Zaradi požara je nastalo za okrog 100 tisočakov škode.
- Do požara, v katerem je nastalo za okoli deset tisoč evrov materialne škode, naj bi prišlo zaradi preobremenitve transformatorske postaje.
- S hitrim posredovanjem so preprečili, da bi se ogenj razširil, vseeno pa je v požaru nastalo za 15.000 evrov škode.
- Pri ugotavljanju vzrokov požara, v katerem je nastalo za okoli milijon tolarjev škode, so ugotovili, da gre skoraj zagotovo za požig.
- V zadnjem požaru je nastalo za 600 tisočakov škode.
- Primož pri Ljubnem - Poldrugi milijon škode je nastalo v požaru, ki je v ponedeljek malo po osmi uri zjutraj zajel starejšo zapuščeno stanovanjsko hišo.
- Pri požaru, ki je domnevno nastal zaradi napake na električni napeljavi, je nastalo za okoli 2 milijona tolarjev škode.
- Zaradi hitrega posredovanja gasilcev se ogenj ni razširil, tako da je ob požaru nastala le manjša škoda.
- Po nestrokovni oceni je v požaru nastalo za 75 tisoč evrov škode.

- Koper - V avtoličarski delavnici v Spodnjih Škofijah je v ponedeljek okoli 15.30 izbruhnil požar, v katerem je nastala večja škoda.
- Do požara, v katerem je nastalo za okoli 15.000 evrov škode, je prišlo zaradi pregretja dimnika, na katerega sta bila prislonjena avtomobilska pnevmatika in sedež, ki sta se vnela.
- ILIRSKA BISTRICA Za poldrugi milijon tolarjev škode je med požarom nastalo na počitniški hiši Carinarnice Sežana na Sviščakih.
- Škocjan – Za okrog 50 tisoč evrov škode je nastalo v požaru na Jelendolu, potem ko je v petek popoldne strela udarila v kozolec.
- Duplek - V požaru, ki je izbruhnil v kuhinji nadstropne stanovanjske hiše v Zgornji Koreni, je po prvih ocenah nastalo kar za pet milijonov tolarjev škode.

Za iskano semantično trojico dobimo 14 rezultatov. Vsaka izmed najdenih povedi govori o škodi, nastali pri požaru, kar je ustrezno za iskalne parametre. Na podlagi tega ugotovimo, da najdene povedi ustrezno predstavljajo semantično trojico.

Za vsako najdeno poved lahko v bazi poiščemo podatke o avtorju in o objavi. Na primer za poved “Do požara, v katerem je nastalo za okoli 15.000 evrov škode, je prišlo zaradi pregretja dimnika, na katerega sta bila prislonjena avtomobilska pnevmatika in sedež, ki sta se vnela,” ugotovimo, da je bila objavljena leta 2007 pod naslovom Celjan v Novicah. Objavil jo je neznani novinar.

5.3.2 Dopolnitev semantične trojice

Za dopolnjevanje semantične trojice moramo vedno poznati vsaj dva člena trojice, od katerih je relacija obvezna. To pomeni, da lahko iščemo samo subjekte ali objekte. Manjkajoči parameter je v semantični trojici označen s simbolom "*", kar predstavlja nadomestni znak (to je simbol, ki je nedoločen in je lahko nadomeščen s poljubno vsebino). V naslednjih primerih so pravilni rezultati podčrtani.

Iskanje objekta

Kot primer iščemo objekte za semantično trojico (Maribor, izgubiti, *). Ta poizvedba je bila izbrana, ker vemo, da lahko pričakujemo športne rezultate. Rezultati, ki jih dobimo, so naslednji:
8 rezultatov

- Krka niz pet nesrečno *Žužemberčan liga 1.*
  Odbojkarji Krke so po petih nizih nesrečno izgubili z Mariborom - Žužemberčani prijetno presenečenje 1.B lige - Igralke TPV-ja so z manj tekmami še vedno prve

- delo
  V Mariboru bo izgubilo delo najmanj sto ljudi in bo, po besedah Alenke Iskra, direktorice podjetja Terme (ki je lastnik največjih tovrstnih trgovin na tukajšnjem delu severne meje), samo to podjetje utrpelo zmanjšanje prihodka za 3,5 milijona mark, dodatne štiri milijone mark pa bo stalo odstranjevanje teh trgovin z meje.

- *kar* njegov ukinitev ministrstvo gospodarstvo
  Kangler namreč neprestano opozarja, da bi lahko celo preselili sedež Pošte v Ljubljano, s čimer bi Maribor po njegovem ponovno veliko izgubil (kot že zaradi ukinitve ministrstva za malo gospodarstvo in turizem).

- življenje nesreča *kateri*
  Lenart – Neprevidnost ali prevelika hitrost je, kot kaže, botrovala včerajšni prometni nesreči, v kateri je življenje izgubil 46-letni voznik osebnega avtomobila iz Ruš pri Maribora.

- *Laško* reprezentanca Iran 0:5
  Maribor Pivovarna Laško je izgubil z reprezentanco Irana z 0:5 (0:3)

- *tekma* reprezentanca Koreja
  Lansko sezono je odlično priložnost izpustila generacija slovenskih hokejistov, ki je v zadnji tekmi, v bistvu finalni tekmi, v Mariboru izgubila proti reprezentanci Južne Koreje in s tem porazom »osemnajstico« zasidrala v diviziji 2.

- *tekma* Sobota Creativ 107:82
  V prijateljski košarkarski tekmi so v Murski Soboti košarkarji soboškega Creativa s 107:82 izgubili proti košarkarjem Pošte Maribora Branika.

- slavija m Optima 2:16
  Maribor je doma izgubil s Slavijo M Optimo z 2:16, MARC Interieri pa so zmagali z Bledom s 4:3.

V rezultatih vidimo, da pet od osmih povedi govori o športni tekmi, kot je bilo pričakovano. Preostale tri povedi pa govorijo o različnih področjih, kot je izguba dela ali življenja.

Pri tem primeru najbolj izstopa primer “Laško reprezentanca Iran”. Vzrok za to leži v obliki izvorne povedi “Maribor Pivovarna Laško je izgubil z reprezentanco Irana z 0:5 (0:3)”. Ker se poved začne slovnično nepravilno z besedo “Maribor”, sistem CLASSLA ne zna dobro obravnavati povedi, zato neodvisno loči besedo “Maribor” od zaporedja “Pivovarna Laško”. To povzroči napačne ekstrakcije in posledično rezultate poizvedbe.

Kot primer iščemo objekte za semantično trojico (Olimpija, premagati, *). Ta primer je bil izbran, ker prav tako kot pri prejšnjem primeru tudi tukaj vemo, da lahko pričakujemo športne rezultate. Rezultati, ki jih dobimo, so naslednji:
17 rezultatov

- Zalog slavija Jata
  Ekipa Acroniksa Jesenic je v torek v Mariboru slavila z minimalno prednostjo 1:2 (0:1, 1:1, 0:0), Olimpija Hertz pa je v Zalogu premagala Slavijo Jato z 1:16 (1:6, 0:2, 0:8).

- Jadran Lamo
  Obakrat je SCT Olimpija premagala Jadran Lamo z 12:0 in sicer v sezonah 1991/92 in 1995/96.

- tekma Jadran
  Olimpija je v drugi tekmi premagal Jadran s 35:12.

- on
  Najprej jih je doma premagala Olimpija, nato pa Triglav.

- ponedeljek Cibalia *ta*
  V ponedeljek je Olimpija namreč premagala Cibalio, toda kljub temu športni direktor nad prikazanim ni bil navdušen.

- tekma dva Medveščak *volan 4:1 Slavija*
  V drugih dveh tekmah je ZM Olimpija z 10:1 premagala Medveščak, Alba Volan pa je bila s 4:1 boljša od VTZ Slavije.

- *finale zmaga*
  V finalu so kar s 4:0 v zmagah premagali večne tekmece hokejiste ZM Olimpije.

- Joventut točka 17
  Pred tekmo so nekateri upali, da bo Olimpija premagala Joventut za

več kot 17 točk, za kolikor je izgubila v Badaloni, ob polčasu pa bi mnogi takoj stavili, da bo razlika tolikšna – a za goste.

- primer Cibona *partizan*
  V primeru, da Olimpija premaga Cibono, Partizan pa AEK, mora Pau Orthez premagati Romo.

- poskus liga
  ŠIROKI BRIJEG - Košarkarji Širokega tudi v šestem poskusu v ligi Goodyear niso uspeli premagati Uniona Olimpije.

- *izid Orlando 97:84 Antonio 79:93* krog 20. liga lija Primorje
  Izidi: Charlotte Orlando 97:84, Portland San Antonio 79:93, Houston Los Angeles 99:87V 20. krogu lige slovenske nogometne lije je Olimpija premagala Primorje z 1:0

- *turnir liga bazen Triglav*
  Na finalnem turnirju lige Alpe Adria so v domačem bazenu vaterpolisti Triglava najprej premagali ekipo Slovana Olimpije, v polfinalu častno izgubili s kasnejšim prvakom Uniqu UTE in v borbi za tretje mesto potopili Dunajčane.

- *Kmetec minuta 83.*
  Gostje so prek Kmetca, ki je zadel z glavo po podaji Gerenčerja, v 83. minuti vendarle uspeli premagati komaj 16-letnega vratarja Olimpije Oblaka.

- finale Sportino
  V finalu je Olimpija premagala Sportino s 6:3.

- sezona CMC publikum prvenstvo
  Nepomembna tudi ni okoliščina, da je Olimpija v novi sezoni CMC Publikum premagala s 3:1 v prvenstvu in z 1:0 v tekmovanju za pokal NZS.

- hala Tivoli Alleghe 5:4
  Ljubljanska Olimpija pa je sinoči v hali Tivoli premagala Alleghe s 5:4 (2:3, 1:1, 2:0).

- skupina petek gostovanje Graz 99 Tivoli Innsbruck 5
  V kvalifikacijski skupini je ekipa ZM Olimpije najprej v petek na gostovanju premagala Graz 99ers z 1 : 6, nato pa je bila v nedeljo doma v Tivoliju boljša od Innsbrucka s 5 : 2 in osvojila prvo mesto v tej skupini.

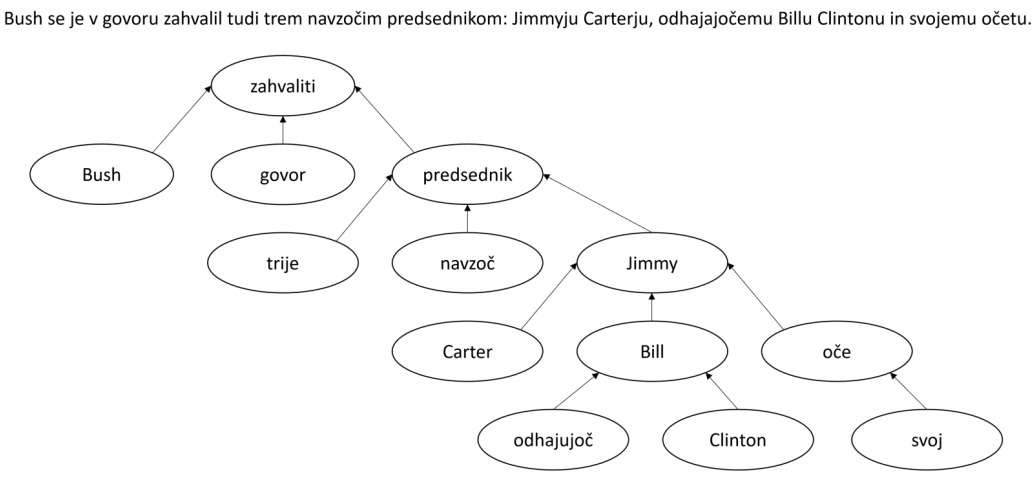
Pričakovan rezultat v temu primeru je množica objektov iz povedi, ki govorijo o zmagi kluba Olimpije na neki tekmi. Izmed sedemnajstih najdenih povedi le tri ne govorijo o zmagi kluba z imenom Olimpija, vseeno pa govorijo o porazu.

Za razliko od prejšnjega primera pri tem primeru dobimo povedi, ki govorijo le o športnih rezultatih. Vzrok za to je v tem, da Maribor ni samo klub, ampak je tudi kraj, medtem ko Olimpija pomeni skoraj samo klub.

Tukaj ima sistem približno 30 % napačnih rezultatov, od tega je približno 13 % zaradi napak v Classli ali povedi (kot primer napačen sklon ali manjkajoč presledek med dvema povedima).

Kot primer iščemo objekte za semantično trojico (Bush, zahvaliti, *). V tem primeru pričakujemo rezultate, ki govorijo o ameriškem predsedniku. Rezultati, ki jih dobimo so nasledni:
3 rezultati

- govor predsednik trije Jimmy Carter
  Bush se je v govoru zahvalil tudi trem navzočim predsednikom: Jimmyju Carterju, odhajajočemu Billu Clintonu in svojemu očetu.
- gostitelj pomoč oskrbovanje postaja katastrofa Columbia
  Državnika sta se tudi dogovorila o sodelovanju v vesoljskem programu, Bush pa se je gostitelju zahvalil za pomoč pri oskrbovanju mednarodne vesoljske postaje po katastrofi Columbie.
- premier Mahmud Abas udeležba vrh
  Bush se je še zahvalil palestinskemu premieru Mahmudu Abasu za udeležbo na vrhu in ga pozval k ustavitvi terorističnih napadov na izraelske cilje.



Slika 5.4: Semantično drevo za dano poved.

Pri tem primeru vidimo, da so vsi rezultati pravilni, a če pogledamo poved "Bush se je v govoru zahvalil tudi trem navzočim predsednikom: Jimmyju

Carterju, odhajajočemu Billu Clintonu in svojemu očetu," vidimo, da nam pri rezultatu "govor predsednik Jimmy Carter" manjkajo rezultati Bill Clinton in oče. Vzrok za tak rezultat je v napačno najdenih odvisnostih s sistemom CLASSLA.

V semantičnem drevesu je razvidno, da sta ostali dve osebi odvisni od prve, čeprav so osebe enakovredno naštete. Besedi "Bill" in "oče" v tem drevesu nastopata kot veznik (CONJUNCT). Glede na pravila pri ekstrakciji za slovenščino, se na tej točki ustavi iskanje objekta. To pomeni, da algoritem ne najde osebe "Bill Clinton" ali "oče" kot primernega objekta.

Iskanje subjekta

Kot primer iščemo subjekte za semantično trojico (*, pripraviti, ocena). Ta primer je bil izbran, ker na podlagi navedenih parametrov vemo, da lahko pričakujemo rezultate o tem, kdo je pripravil oceno. Rezultati, ki jih dobimo, so naslednji: 13 rezultatov

- sklad republika Slovenija
  Po ocenah, ki so jih pripravili v Stanovanjskem skladu Republike Slovenije, bo v novi varčevalni shemi 12 do 15 tisoč varčevalcev

- skupina vrnitev
  Ocenjevalna skupina bo po vrnitvi v Bruselj pripravila oceno, pri kateri bo lahko s svojimi pripombami in predlogi aktivno sodelovala tudi Slovenija, saj bodo naši predstavniki (na ravni državnih sekretarjev) sodelovali na vojaško-političnem odboru zavezništva 11. marca v Bruslju.

- *rebalans* konec junij
  Vsaj osnove rebalansa bi morali po moji oceni pripraviti do konca junija in ga sprejeti jeseni, kar sicer pade v silno neprijetno predvolilno obdobje.

- torek
  Breščak je za projekt ureditve CPB obljubil, da bo do torka, 30.5.2006 pripravil predračun za dodatno arheološko izkopavanje z oceno trajanja in stroškov teh del.

- Kostanjevica Krka društvo vinogradnik
  KOSTANJEVICA NA KRKI - Društvo vinogradnikov Kostanjevica je pripravilo oceno vinskih vzorcev.

- *sojenje oktober*
  Ker se strokovni mnenji o stopnji opitosti tako razlikujeta, je sodišče odredilo, da bo komisija medicinske fakultete za izvedenska mnenja oboje

ocenila in pripravila neodvisno oceno, tako da se bo sojenje nadaljevalo 12. oktobra.

- urad osnova metodologija
  Realna rast bruto domačega proizvoda v letu 2002 je po oceni, ki jo je statistični urad pripravil na osnovi nove metodologije in v cenah novega baznega leta 2000, znašala 3,2 odstotka, kar je povsem enako napovedi UMAR iz jesenskega poročila.

- ZPO
  Ker tudi s prenovo in rekonstrukcijo zdajšnje športne dvorane na Golovcu ni mogoče izpeljati tekmovanja, je ZPO pripravil za mestni svet urbanistično, arhitektonsko in projektno dokumentacijo z investicijsko oceno ter investicijskim programom za gradnjo novega športnega objekta na lokaciji severno od novega nogometnega stadiona ter hipermarketa Mercator.

- delavec OVS
  Oceno ogroženosti objekta pripravi delavec OVS, elaborat straže pa izdela pristojni častnik Slovenske vojske.

- vlada
  Vlada je rebalans pripravila na podlagi ocen iz jesenske napovedi gospodarskih gibanj Urada RS za makroekonomske analize in razvoj, od tedaj pa so se makroekonomske razmere pogoji spremenili, poleg tega so bila proračunska gibanja v letu 2007 drugačna od napovedanih, saj je proračun prav tako ustvaril presežek.

- *naloga program člen zakon* september
  Za izvajanje nalog po programih iz 3. člena tega zakona se pripravi vsako leto do 1. septembra operativni plan aktivnosti za izvedbo programov v naslednjem letu z oceno potrebnih sredstev, do 31. marca pa poročilo o izvajanju programa za preteklo leto.

- odvetnica Petra Starič Bohinj
  Odvetnica Petra Starič je za občino Bohinj pripravila oceno, ki popolnoma zavrača zahtevek sedmih ljubljanskih lastnikov.

- podjetje Elite
  Podjetje Elite je pripravilo oceno investicije.

Med rezultati res prevladujejo osebe, ki so pripravile neko oceno, vendar pa obstajajo tudi primeri, ki govorijo o tem, kdaj ali kako je bila ta ocena pripravljena.

Če pogledamo primer "urad osnova metodologija" in ga primerjamo z izvorno povedjo "Realna rast bruto domačega proizvoda v letu 2002 je po oceni, ki jo je statistični urad pripravil na osnovi nove metodologije in v cenah novega baznega leta 2000, znašala 3,2 odstotka, kar je povsem enako napovedi UMAR iz jesenskega poročila," ugotovimo, da subjekt pokriva dve vprašanji:

- Kdo pripravi oceno? Urad.

- Kako pripravi oceno? Na osnovi metodologije.

V relaciji z nekim objektom je lahko več različnih subjektov. Taki subjekti so medsebojno odvisni. Informacija "na osnovi metodologije je pripravil oceno" nam ne pove, kdo je tako oceno pripravil. Vsi subjekti v relaciji z nekim objektom skupaj opisujejo celotno informacijo.

Kot primer iščemo subjekte za semantično trojico (*, predstaviti, ponudba). Ta primer je bil izbran, ker glede na vnešene parametre vemo, da lahko pričakujemo osebo, ki je predstavljala neko ponudbo. Rezultati, ki jih dobimo so naslednji:
28 rezultatov

- čas predstavnik Bizeljsko društvo
  V popoldanskem času pa so svoj del ponudbe na Bizeljsko sremiški vinsko turistični cesti predstavili predstavniki Turističnega društva Bizeljsko.

- konferenca vodstvo banka Celje
  Na tiskovni konferenci je vodstvo Banke Celje predstavilo tudi novosti v ponudbi.

- družba HIT
  Družba Hit je namreč takoj zatem predstavila svojo ponudbo partnerstva pri ustanavljanju zasebno-javnega podjetja, v katero naj bi vključili tudi tukajšnje občine, a v občinah Miren - Kostanjevica in Renče - Vogrsko so za zdaj zelo zadržani, saj uradne ponudbe še nimajo.

- bančnik
  Bančniki vam bodo predstavili NLB Leasing ponudbo, poslovnim subjektom pa predstavili načine za razpršitev tveganj, ki jih ponujajo faktorinške in podobne storitve.

- recaro Slovenija zastopnik poligon
  Pri Recaro Slovenija, ki je v naši državi zastopnik ali prodajalec za znamke Recaro, Schroth in Stilo, so na logaškem poligonu predstavili novosti v svoji prodajni ponudbi.

- radio Jesenice
  Lastnikom radia je v soboto na Jesenicah predstavil svojo ponudbo za

odkup njihovih deležev oziroma dokapitalizacijo, na njihovo odločitev pa še čaka.

- iskra sistem Ljubljana sejem elektronika
  ISKRA SISTEMI, d.d. iz Ljubljane se bo tudi letos s svojimi novostmi v ponudbi predstavila na sejmu Sodobna elektronika 2002 v Ljubljani.

- predavanje *Gal*
  Ob premiernem predvajanju bodo pestro ponudbo domače scene predstavili Gal in Galeristi s svežim prvencem Fetiš ter ta trenutek ena najbolj aktualnih zasedb Leaf-fat.

- podjetje Kompas februar
  V največjem slovenskem turističnem podjetju Kompas so 2. februarja predstavili najnovejši katalog iz letošnje ponudbe - Poletje 2000.

- podjetje TÜV Bayern Sava *komisija*
  Znano je le, da je podjetje TÜV Bayern Sava svoje poročilo o vrednotenju ponudb treh ponudnikov, Impakte, Salbatringa in Gorenja, predstavilo članom razpisne komisije in nadzornega odbora, ti pa naj bi zahtevali dodatna pojasnila in sicer do jutri.

- Real Madrid
  Ljubljana - Dan pred zadnjim rokom registracije novih okrepitev na španski in italijanski nogometni sceni se je spet pogrela »stara« juha: Real Madrid je Interju včeraj predstavil še zadnjo ponudbo za Ronalda.

- veselje podjetje Domos
  Z veseljem vam bodo v podjetju Domos predstavili svojo ponudbo in vam približali za vas najprimernejše izdelke- ležišča.

- podjetje Kočevje restavracija hotel Valentin
  PREDSTAVILI PONUDBO "KORENINE" - Podjetje za zaposlovanje in rehabilitacijo invalidov Recinko, d.o.o., iz Kočevja je v petek popoldan v restavraciji hotela Valentin predstavilo pestro ponudbo izdelkov,

ki jih je moč dobiti v njihovi pred nedavnim odprti specializirani trgovini Korenina v prostorih Valentina.

- salon Frankfurt BMW
  Na septembrskem avtomobilskem salonu v Frankfurtu bo BMW predstavil popolno novost v ponudbi, in sicer X3.

- *oglas* oglaševalec teleteks
  Ker so televizijski oglasi vedno krajši in sporočajo le osnovno informacijo, lahko oglaševalec svojo ponudbo predstavi na teletekstu, nanjo pa opozori na televizijskem oglasu.

- Canon *razred*
  ZA: Canon je predstavil osvežitev ponudbe optičnih bralnikov tudi v srednjem cenovnem razredu, kjer novi model CanoScan 5600F pokriva luknjo med starejšima modeloma 4400 in 8800.

- *ponudba letovanje* prireditev občina *podeželje*
  Poleg ponudbe turističnih letovanj doma in v tujini, rekvizitov za šport, avdio in video opreme, pripomočkov za lov, ribolov in prosti čas ter plovil so se na prireditvi predstavile tudi slovenske občine z bogato turistično ponudbo našega podeželja.

- podjetje Maremico sejm Ljubljana
  Podjetje Maremico bo na pohištvenem sejmu v Ljubljani predstavilo novosti iz svoje ponudbe, o katerih govori tudi nov katalog Lectus 2008.

- desetletnica podjetje Soča delovanje rafting Bovec
  Ob desetletnici delovanja se je podjetje Soča rafting iz Bovca predstavilo z novo celostno podobo svoje bogate ponudbe športnega turizma, ki je ena najperspektivnejših vej.

- številka Val
  V nekaj prejšnjih številkah Vala smo vam predstavili gumenjake iz sre-

dine ponudbe napihljivih čolnov različnih italijanskih izdelovalcev iz velikostnega razreda šestih metrov.

- ta tržnica
  Ti so na adventni tržnici in na turistični delavnici predstavili "vročo" slovensko ponudbo, "ki je zares presenečenje", kot je dejal Gerhard Buxbaum, predstavnik avstrijske agencije Ruefa Reisen.

- Magda Krošelj
  Ponudbo na krškem območju GDVC je predstavila Magda Krošelj.

- *kar*
  Poskrbela naj bi za predstavitev različnih obrti in dejavnosti s področja turizma in gostinstva, kulinarike in obrti, kar bi predstavilo pestro in vabljivo slovensko ponudbo.

- Digitel
  Digitel predstavil ponudbo za koncesionarja GSM

- konec leto
  Kot novost so na koncu leta predstavili ponudbo celotne baze na portalu wap (naslov: wap.wlw.si), na katerem je mogoče iskati po storitvah, izdelkih ali imenu podjetja.

- mlad
  Mladi so se obiskovalcem, ki so napolnili Kosovo gostišče, predstavili s svojevrstno ponudbo.

- konferenca sejm vino
  Včeraj so na novinarski konferenci pred sejmom Vino in Kulinarika predstavili tudi rezultate raziskovalne naloge o ponudbi vin v Sloveniji, ki jo je pri agenciji Ninamedia naročila poslovna skupnost, sofinanciralo pa jo je kmetijsko ministrstvo.

- nissan segment vozilo
  Nissan je v segmentu malih vozil predstavil najmanjši avtomobil v svoji ponudbi, namenjen za uporabo v mestnih središčih in urbanih naseljih.

Prav tako kot pri prejšnjem primeru tudi tukaj prevladujejo rezultati, ki govorijo o tem, kdo predstavlja ponudbo, vendar pa je pri tem primeru tudi večje število rezultatov, ki govorijo o tem, kdaj je prišlo do predstavitve in kaj je bilo predstavljeno.

Poglavje 6

Sklepne ugotovitve

V korpusu Gigafida je 7 milijonov povedi, od katerih so nekatere bolj, druge manj primerne za ekstrakcijo informacij. To je razvidno iz tega da določene povedi ne vrnejo nobene ekstrakcije, določene povedi pa lahko tudi več kot tisoč ekstrakcij, čeprav je povprečje okrog devet ekstrakcij na poved. Število ekstrakcij navzgor ni omejeno, saj tudi število stavkov v povedi ni omejeno. Znotraj korpusa tudi niso vse povedi v standardni slovenščini, ampak lahko vsebujejo tudi narečja ali slovnične napake.

Čeprav je uporabljen nestandardni model za slovnično analizo povedi v slovenščini, so primeri, da oblika povedi in izbrane besede povzročijo, da te besede ne morejo biti ustrezno prepoznane. To se najprej prepozna pri napačni izgradnji semantičnega drevesa, kar posledično prinese napačno izbiro in povezovanje subjektov in objektov. Tipična napaka pri slovnični analizi je neprepoznavanje besed kot ime ali kot tujka ali pa poved uporablja napačna ločila (primer nestandardni apostrof namesto standardni apostrof v tujih frazah).

V primerjavi z angleščino ima ekstrakcija v slovenščini zaradi dinamičnega besednega reda in sklanjatev bolj kompleksna pravila. Za natančno ekstrakcijo niso dovolj le pravila z uporabo besednih vrst, ampak so potrebna tudi pravila z odvisnostjo relacij in sklonov. V slovenščini sklanjatve omogočajo odločitev, če je nek člen objekt ali subjekt. Besede v tožilniku na primer

lahko nastopajo samo kot objekt. Sistem za odprto ekstrakcijo informacij za slovenščino temelji na podlagi pravil (druga generacija), vendar zaradi navedenih razlogov zavzema obe vrsti (na podlagi pravil, plitke sintakse in razčlembe odvisnosti) in ne le ene.

Pri dopolnjevanju semantičnih trojic s poizvedovanjem po rezultatih nad korpusom ugotovimo, da določen dopolnjeni člen lahko odgovarja na več vprašanj (kot primer, kdo je nekaj naredil in kako je nekaj naredil).

Članki v revijah

[1] Sally Ali, Hamdy Mousa in M Hussien. "A Review of Open Information Extraction Techniques". V: *IJCI. International Journal of Computers and Information* 6.1 (2019), str. 20–28.

[4] Oren Etzioni in sod. "Open information extraction from the web". V: *Communications of the ACM* 51.12 (2008), str. 68–74.

[8] Stuart James. "The Chambers Dictionary". V: *Reference Reviews* (2009).

[9] Ruth E. Kott. "The origins of writing". V: *The University of Chicago Magazine* (2013). DOI: 10.1080/14626268.2016.1258422.

[12] Christina Niklaus in sod. "A survey on open information extraction". V: *arXiv preprint arXiv:1806.05599* (2018).

[13] Aleksandra Bizjak Primož Jakopin. "Part-of-speech tagging of Slovenian text". V: *Slavistična revija* 45."3-4" (1997), str. 513–532.

Članki v zbornikih

[2] Luciano Del Corro in Rainer Gemulla. “Clausie: clause-based open information extraction”. V: *Proceedings of the 22nd international conference on World Wide Web*. 2013, str. 355–366.

[5] Anthony Fader, Stephen Soderland in Oren Etzioni. “Identifying Relations for Open Information Extraction”. V: *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*. Edinburgh, Scotland, UK, jul. 2011.

[6] Anthony Fader, Stephen Soderland in Oren Etzioni. “Identifying relations for open information extraction”. V: *Proceedings of the 2011 conference on empirical methods in natural language processing*. 2011, str. 1535–1545.

[7] Toporišič J. “Slovenska slovnica”. V: *Obzorja*. Maribor, 2004.

[10] Cong Li in sod. “The Partition Heuristic Information Extraction Algorithm of Unstructured Data”. V: *2013 International Conference on Cloud Computing and Big Data*. IEEE. 2013, str. 570–576.

[11] Nikola Ljubešić in Kaja Dobrovoljc. “”What does Neural Bring? Analysing Improvements in Morphosyntactic Annotation and Lemmatisation of Slovenian, Croatian and Serbian””. V: *”Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing”*. ”Florence, Italy”: ”Association for Computational Linguistics”, avg. 2019, ”29–34”. DOI: "10.18653/v1/W19-3704". URL: https://www.aclweb.org/anthology/W19-3704.

[14] Peng Qi in sod. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". V: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2020.

[16] Michael Schmitz in sod. "Open language learning for information extraction". V: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 2012, str. 523–534.

[17] Fei Wu in Daniel S Weld. "Open information extraction using wikipedia". V: *Proceedings of the 48th annual meeting of the association for computational linguistics*. 2010, str. 118–127.

Celotna literatura

[1] Sally Ali, Hamdy Mousa in M Hussien. "A Review of Open Information Extraction Techniques". V: *IJCI. International Journal of Computers and Information* 6.1 (2019), str. 20–28.

[2] Luciano Del Corro in Rainer Gemulla. "Clausie: clause-based open information extraction". V: *Proceedings of the 22nd international conference on World Wide Web*. 2013, str. 355–366.

[3] *Universal Dependency Relations*. 2022. URL: `https://universaldependencies.org/u/dep/index.html` (pridobljeno 19. 2. 2022).

[4] Oren Etzioni in sod. "Open information extraction from the web". V: *Communications of the ACM* 51.12 (2008), str. 68–74.

[5] Anthony Fader, Stephen Soderland in Oren Etzioni. "Identifying Relations for Open Information Extraction". V: *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*. Edinburgh, Scotland, UK, jul. 2011.

[6] Anthony Fader, Stephen Soderland in Oren Etzioni. "Identifying relations for open information extraction". V: *Proceedings of the 2011 conference on empirical methods in natural language processing*. 2011, str. 1535–1545.

[7] Toporišič J. "Slovenska slovnica". V: *Obzorja*. Maribor, 2004.

[8] Stuart James. "The Chambers Dictionary". V: *Reference Reviews* (2009).

[9] Ruth E. Kott. "The origins of writing". V: *The University of Chicago Magazine* (2013). DOI: `10.1080/14626268.2016.1258422`.

[10] Cong Li in sod. "The Partition Heuristic Information Extraction Algorithm of Unstructured Data". V: *2013 International Conference on Cloud Computing and Big Data*. IEEE. 2013, str. 570–576.

[11] Nikola Ljubešić in Kaja Dobrovoljc. ""What does Neural Bring? Analysing Improvements in Morphosyntactic Annotation and Lemmatisation of Slovenian, Croatian and Serbian"". V: *"Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing"*. "Florence, Italy": "Association for Computational Linguistics", avg. 2019, "29–34". DOI: "10.18653/v1/W19-3704". URL: https://www.aclweb.org/anthology/W19-3704.

[12] Christina Niklaus in sod. "A survey on open information extraction". V: *arXiv preprint arXiv:1806.05599* (2018).

[13] Aleksandra Bizjak Primož Jakopin. "Part-of-speech tagging of Slovenian text". V: *Slavistična revija* 45."3-4" (1997), str. 513–532.

[14] Peng Qi in sod. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". V: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2020.

[15] *RDF 1.1 Primer*. 2022. URL: https://www.w3.org/TR/rdf11-primer/#section-triple (pridobljeno 7. 3. 2022).

[16] Michael Schmitz in sod. "Open language learning for information extraction". V: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 2012, str. 523–534.

[17] Fei Wu in Daniel S Weld. "Open information extraction using wikipedia". V: *Proceedings of the 48th annual meeting of the association for computational linguistics*. 2010, str. 118–127.