Sveučilište u Zagrebu
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Goran Glavaš

CRPLJENJE I PRETRAŽIVANJE TEKSTNIH INFORMACIJA TEMELJEM GRAFOVA DOGAĐAJA

DOKTORSKI RAD

Zagreb, 2014.

Sveučilište u Zagrebu
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Goran Glavaš

CRPLJENJE I PRETRAŽIVANJE TEKSTNIH INFORMACIJA TEMELJEM GRAFOVA DOGAĐAJA

DOKTORSKI RAD

Mentor: Doc. dr. sc. Jan Šnajder

Zagreb, 2014.

University of Zagreb
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Goran Glavaš

TEXT INFORMATION EXTRACTION AND RETRIEVAL BASED ON EVENT GRAPHS

DOCTORAL THESIS

Supervisor: Assistant Professor Jan Šnajder, PhD

Zagreb, 2014

Doktorski rad izrađen je na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva, na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave, u Laboratoriju za analizu teksta i inženjerstvo znanja (TakeLab).

Mentor: Doc. dr. sc. Jan Šnajder

Doktorski rad ima: 240 stranica

Doktorski rad br.: _____

O mentoru

Jan Šnajder diplomirao je, magistrirao i doktorirao u polju računarstva na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva (FER), 2002., 2006. odnosno 2010. godine. Od rujna 2002. godine radi kao znanstveni novak, a od 2011. godine kao docent na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave FER- a. Od 2012. do 2013. godine bio je gostujući istraživač na Institutu za računalnu lingvistiku Sveučilištu u Heidelbergu.

Doc. Jan Šnajder sudjelovao je na četiri znanstvena projekta Ministarstva znanosti, obrazovanja i sporta Republike Hrvatske, četiri međunarodna projekta i dva domaća projekata. Bio je voditeljem poslijedoktorskog projekta "Uporaba računalnih modela tvorbene semantike u pretraživanju informacija", koji je financirala Hrvatska zaklada za znanost. Autor je ili suautor više od 50 znanstvenih radova u časopisima i zbornicima konferencija u području obrade prirodnog jezika i pretraživanja informacija. Bio je mentorom ili sumentorom studentima na 60 preddiplomskih i diplomskih radova.

Doc. Jan Šnajder član je stručnih udruga IEEE, ACM, ACL i Hrvatskog društva za jezične tehnologije. Sudjelovao je u više međunarodnih programskih odbora znanstvenih konferencija u području računalne lingvistike te je bio recenzentom u većem broju inozemnih časopisa i na međunarodnim konferencijama. Godine 2010. primio je srebrnu plaketu "Josip Lončar" FER-a za posebno istaknutu doktorsku disertaciju.

About the Supervisor

Jan Šnajder has received his BSc, MSc, and PhD degrees in Computer Science from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia, in 2002, 2006, and 2010, respectively. From September 2002 he is working as a research assistant, and from 2011 as an Assistant Professor at the Department of Electronics, Microelectronics, Computer and Intelligent Systems at FER. From 2012 to 2013 he was a visiting researcher at the Institute for Computational Linguistics at the University of Heidelberg.

Doc. Jan Šnajder participated in four research projects funded by the Ministry of Science, Education and Sports of the Republic of Croatia, four international projects, and two domestic projects. He was the PI of the postdoc project "Derivational Semantic Models for Information Retrieval", funded by the Croatian Science Foundation. He has (co-) authored more than 50 papers in journals and conferences in natural language processing and information retrieval. He has supervised and co-supervised 60 BA and MA theses.

Doc. Šnajder is a member of IEEE, ACM, ACL, and the Croatian Language Technologies Society. He participated in several program committees of computational linguistics conference and reviewed for a number of international journals and conferences. In 2010 he was awarded the Silver plaque "Josip Lončar" from FER for an outstanding PhD thesis.

Zahvala

Janu, koji je za ovaj doktorat zaslužan koliko i ja. Janu, uz kojeg sam rastao (i još uvijek rastem) kao istraživač i kao čovjek. Janu, koji ne prestaje učiti (tako je navodno naučio i popuštati) i podučavati druge te jedinoj osobi koja me u odrasloj dobi rasplakala u javnosti.

Mami, koja je za trajanja doktorata oprala nekoliko tona prljavog rublja, skuhala nebrojeno ručaka te učinila Jana ovisnim o svojim kolačima. Mami kojoj nije ništa teško (pa ni privremeno preseliti u Zagreb) i mami koja nikad ne zaboravi nacrtati smajlića na pjenu Nescaffea od čokolade.

Tati, koji je vjerovao u ovaj doktorat i kad ja nisam i koji me redovito podsjećao koliko znam i vrijedim. Tati koji me nagovorio da odem u znanost i bio moj granitni oslonac cijelim putem. Konačno, tati koji je (vlastitim izborom!) naučio puno o grafovima događaja i važnosti NLP konferencija.

Seki, koja jedina zna u potpunosti kako sam se u nekim trenucima osjećao. Seki, s kojom sve dijelim i pred kojom nemam tajni. Seki, koja s ovom disertacijom više nije jedini doktor u obitelji (ali i dalje jedina smije prepisivati lijekove).

Mladenu, vjernome drugu i suborcu u toj napornoj bitci zvanoj doktorat. Mladenu, koji je požrtvovno preuzimao na sebe brojne zadatke kako bih ja imao više vremena za istraživanje. Mladenu, duši od čovjeka i umješnom troleru.

Profesorici Dalbelo, koja mi je omogućila da postanem član TakeLaba i znanstveni novak, koja drži labos na okupu i koja mi je, unatoč vlastitim životnim izazovima, bila stalna potpora tijekom doktorata.

Papchetu, Čikitu i Mećiju, koji su najzaslužniji što se moj socijalni život tijekom doktorata nije ugasio (ok, barem ne potpuno :).

Svima ostalima koji su bili dio mog života u ovom intenzivnom razdoblju, svima koji su trpili moja ćudljiva raspoloženja i povremene frustracije, svima koji su se iskreno veselili mojim uspjesima i bili potpora u manje uspješnim momentima.

Sažetak

Tekstni izvori koji opisuju događaje iz stvarnoga svijeta (npr. novinski članci) sve su brojniji, a informacijske potrebe korisnika koje se tiču događaja sve su izraženije. Stoga su postupci za automatizirano crpljenje i pretraživanje informacija o događajima sve potrebniji. U okviru disertacije predstavljen je model grafa događaja kao strukture koja sadrži sve bitne informacijske aspekte događaja iz stvarnog svijeta. Vrhovi grafa događaja predstavljaju pojedinačna spominjanja događaja u tekstu, a bridovi vremenske odnose među njima. Ostvaren je potpuno automatizirani postupak izgradnje grafova događaja iz teksta koji kombinira modele za crpljenje informacija temeljene na nadziranom strojnom učenju s modelima temeljenim na pravilima. Provedeno je iscrpno intrinzično eksperimentalno vrednovanje svih modela koji sudjeluju u izgradnji grafova događaja, a predstavljene su i dvije nove mjere za vrednovanje ukupne kakvoće automatski izgrađenih grafova događaja. Predstavljen je model za usporedbu dokumenata usporedbom grafova događaja pomoću jezgrenih funkcija nad grafovima. Učinkovitost predstavljanja dokumenata grafovima događaja i njihove usporedbe jezgrenim funkcijama nad grafovima utvrđena je ekstrinzičnim vrednovanjem na različitim zadatcima pretraživanja informacija. Korisnost crpljenja i strukturiranja informacija o događajima iz teksta dodatno je potvrđena vrednovanjem na zadatcima sažimanja grupa dokumenata te pojednostavljivanja novinskih članaka. Pristupi crpljenju i pretraživanju informacija opisani u ovoj disertaciji usredotočeni su na engleski jezik, ali ih je, uz pretpostavku postojanja određenih jezičnih resursa i alata, moguće prilagoditi na način da budu primjenjivi i za druge jezike.

Ključne riječi: crpljenje tekstnih informacija, graf događaja, jezgrene funkcije nad grafovima, pretraživanje tekstnih informacija, obrada prirodnog jezika.

Summary

Text Information Extraction and Retrieval Based on Event Graphs

As our society becomes increasingly digital, there are a growing number of textual information sources (e.g., breaking news, investigative stories, police reports, tweets, historical texts, electronic health records) that are filled with descriptions of events. The ability to automatically extract and analyse events from text is now more important than ever, with applications that range from security and intelligence to journalism, media analysis, and historical research. Efficiently satisfying event-oriented user information needs requires precise extraction of event-related information, which is a very demanding task considering the complexity, vagueness, and ambiguity of natural language.

In text, real-world events are represented by the so-called linguistic events, or event mentions. Due to ambiguity and vagueness of natural language, the mapping of real-world events and their relations (temporal, causal, etc.) to their linguistic counterparts introduces a loss of information. Event mentions are structured – they consist of event anchors, being words bearing the core meaning of events, and event arguments, being the phrases that denote protagonists and circumstances (e.g., time and location) of events. Documents describing real-world events, thus, give rise to a structure in which there are relations between different event mentions as well as relations between anchors and arguments within individual event mentions.

In this dissertation I have proposed an *event graph*, a structured representation of event-oriented documents containing all informationally-relevant aspects of real-world events. Vertices of event graphs denote individual event mentions extracted from text, whereas edges may denote various semantic relations that hold between event mentions. Although, model-wise, event graphs allow for any semantic relation between events, temporal relations between events have been considered in particular due to inherent temporal aspect of events. Based on the model of event graph, a fully automated procedure for constructing event graphs has been developed. Automated construction of event graphs includes four different information extraction models: (1) a supervised model for extraction of event anchors, (2) a rule-based model for extraction of event arguments, (3) a supervised model for extracting temporal relations between events, and (4) a supervised model for resolving coreference of event mentions. Models for extracting event anchors and temporal relations between event mentions are linear regression models based on rich set of lexical, syntactic, and semantic features. The argument extraction model is based on a set of syntactic extraction patterns and semantic disambiguation rules. The event coreference resolution model is a support vector machines model with the set of numeric features indicating the similarities between anchors and arguments with matching roles between two event mentions. Each of the four models was thoroughly intrinsically evaluated using standard evaluation metrics – precision, recall, and F-score. Two novel metrics for evaluating the

overall quality of the automated construction process have been proposed and empirically validated and the overall quality of automatically constructed event graphs has been measured using these metrics.

In order to develop and evaluate information extraction models included in construction of event graphs, a large corpus, named EVEXTRA, manually annotated with factual event mentions has been compiled. The EVEXTRA is currently the largest corpus manually annotated with event-oriented information. It is approximately three times larger than the TimeBank corpus, which has typically been used in event extraction tasks.

Comparison of documents describing real-world events is performed by comparing their corresponding event graphs. An innovative method for efficient comparison of event graphs, based on semantic extensions of graph kernels has been designed and implemented. Two different graph kernels – product graph kernel and weighted decomposition kernel – have been semantically extended to account for event-specific semantics.

Efficient information retrieval models based on construction and comparison of event graphs have been proposed and evaluated on several information retrieval tasks. Experimental results show that the retrieval models based on event graph and graph kernels outperform traditional retrieval models, which represent documents in an unstructured fashion (i.e., as bags of words) such as vector space models, language models, and probabilistic models.

The usefulness of structured event-centered document representation has been additionally verified on two different natural language processing tasks: multi-document summarization and text simplification. A novel algorithm for multi-document summarization which exploits event-oriented information and temporal structure contained in event graphs has been developed. The novel event-based multi-document summarization algorithm outperforms competitive methods on standard summarization datasets. The algorithm for automated simplification of news stories eliminates all content not relating to event mentions and transforms individual event mentions into separate sentences in the simplified text. Human evaluation shows that text produced with this simplification method are highly grammatical and contain only the most relevant information from the original text.

The research covered in the dissertation focused on texts written in English. Although the event graph formalism itself is language independent, some parts of the models used for automated construction of event graphs are language dependent. The adjustment of the graph construction pipeline for another language is possible, although not an easy task. One of the main directions in future work will tackle adjustment of the automated graph construction pipeline for Croatian.

This dissertation lays the foundation for structured event-based document analysis and uncovers many interesting directions for future research. Event graphs can be extended conceptually by considering relations between event mentions other than temporal relations (e.g., causality,

subordination). Event graphs could also be applied in other natural language processing tasks (e.g., question answering) and other text domains (e.g., biographies). Finally, I envisage a formal framework, based on event graphs, which would enable the modeling of events in a continuous event space that spans from linguistic events at the lowest level to topics at the highest level. Such an event graph-based framework would enable a uniform and elegant treatment of both events and topics for the purpose of event-based document analysis.

Keywords: information extraction, event graph, graph kernels, information retrieval, natural language processing.

Sadržaj

Popis slika

Popis tablica

Poglavlje 1

Uvod

Događajima se, u širem smislu, označava sve što se *ostvaruje* ili *odvija* u vremenu (Pustejovsky i dr., 2003b), pri čemu pojedini događaj osim same akcije određuju i svi sudionici odnosno okolnosti (npr. lokacija, vrijeme) odvijanja te akcije. Informacije o događajima, koji se u stvarnom svijetu odvijaju i izmjenjuju neprestano, zapisuju se i pohranjuju većinom u tekstnom obliku (npr. novinski članci, policijski zapisnici, biografije i dr.). U današnjem digitalnom dobu, potreba za iskorištavanjem informacija o događajima koje se nalaze u tekstnom obliku sve je izraženija, s potencijalnim primjenama od sigurnosnih i obavještajnih aktivnosti do istraživačkog novinarstva, od medijske analitike do povijesnih istraživanja.

Sposobnost ljudi da ručno analiziraju informacije o događajima, njihovim akterima i okolnostima na temelju velikih količina teksta je, nažalost, vrlo ograničena. Stoga je ogromne količine teksta koje nastaju svakodnevno moguće sustavno obraditi jedino automatiziranim postupcima. Nadalje, potrebno je ostvariti učinkovite automatske postupke pretraživanja događaja koristeći prethodno strukturirane informacije o događajima. Postupci automatiziranog crpljenja informacija o događajima pripadaju području crpljenja informacija (engl. *information extraction*, IE) (Pazienza, 1997; Grishman, 2003), dok je pretraživanje događaja dio područja pretraživanja informacija (Moens, 2006; Chowdhury, 2010). Osim za pretraživanje informacija o događajima u velikim tekstnim zbirkama, strukturiranje znanja o događajima crpljenjem informacija iz teksta može biti korisno za mnoge druge zadatke u području obrade prirodnog jezika (engl. *natural language processing*, NLP) poput odgovaranja na pitanja (engl. *question answering*, QA), sažimanja tekstnih informacija (engl. *text summarization*) ili pak automatiziranog pojednostavljivanja teksta (engl. *text simplification*).

Temu ove disertacije čine automatizirani postupci strukturiranja informacija o događajima iz tekstova pisanih prirodnim jezikom te primjene strukturirane reprezentacije događaja u zadatcima pretraživanja informacija i obrade prirodnog jezika. Cilj istraživanja provedenog u ovoj disertaciji bili su razvoj i vrednovanje robusnih postupaka za strukturiranje informacija o događajima iz tekstnih izvora te razvoj postupaka za usporedbu tako strukturiranih informacija

o događajima u cilju učinkovitog pretraživanja informacija, da bi se zadovoljile informacijske potrebe korisnika koje su usmjerene na događaje. Disertacija daje odgovore na sljedeća važna istraživačka pitanja u područjima crpljenja i pretraživanja informacija:

- Je li, i do koje mjere, moguće metodama obrade prirodnog jezika dokumente koji opisuju događaje iz stvarnog svijeta točno i potpuno predstaviti strukturom koja zadržava i ističe ključne informacijske aspekte izvanjezičnih događaja kao što su sudionici događaja i odnosi među događajima?
- Jesu li informacije o događajima koje je moguće strukturirati iz teksta postupcima obrade prirodnog jezika dostatne za utvrđivanje mjere sličnosti i podudarnosti jezičnih događaja koja odgovara ljudskom poimanju sličnosti događaja?
- Postoje li metode za usporedbu strukturiranih informacija o događajima koje se mogu upotrijebiti za učinkovito pretraživanje informacija o događajima?
- Je li strukturu koja sadrži informacije o događajima opisanima u tekstu moguće uspješno primijeniti na druge zadatke analize teksta?

Postupci za crpljenje i pretraživanje informacija o događajima predstavljeni u ovoj disertaciji namijenjeni su tekstovima pisanima engleskim jezikom. Zbog toga su primjeri, koji služe boljem razumijevanju koncepata vezanih za događaje kao i ostvarenih postupaka crpljenja informacija, također pisani engleskim jezikom. Problemi kojima se bavi ova doktorska disertacija nalaze se na presjecištu računarske znanosti, informacijskih znanosti i lingvistike, što disertaciju čini interdisciplinarnom.

1.1 Crpljenje informacija o događajima iz teksta

Prepoznavanje spominjanja događaja, njihovih sudionika i okolnosti odvijanja ključno je za razumijevanje tekstova koji opisuju događaje iz stvarnog svijeta. Prepoznavanje događaja te vremenskih i drugih odnosa među njima u tekstu omogućava predstavljanje dokumenata u strukturiranom obliku (nasuprot teksta, koji je nestrukturiran). Tako uvedena struktura omogućava usporedbu dokumenata prema podudarnosti događaja koje opisuju što, nadalje, omogućava korištenje događaja kao koncepata po kojima se obavlja pretraživanje informacija.

Automatizirano crpljenje informacija o događajima iz teksta, međutim, nije nimalo lagan zadatak. Događaji stvarnog svijeta u tekstu su predstavljeni svojim jezičnim inačicama koje se nazivaju *spominjanjima događaja* (engl. *event mentions, linguistic events*). I dok događaji u stvarnom svijetu imaju jedinstveno prostorno-vremensko prostiranje (tj. ne postoje dva različita događaja u stvarnom svijetu koji se odvijaju u identičnom vremenskom intervalu i na identičnoj lokaciji (Quine, 1985)), pripadna su im spominjanja u tekstu nerijetko višeznačna i neprecizna uslijed inherentne višeznačnosti i nepreciznosti prirodnog jezika. Drugim riječima, rijetko je moguće na temelju spominjanja događaja u tekstu precizno odrediti lokaciju i vremenski in-

terval odvijanja događaja. Posljedično, za mnoga spominjanja događaja nije moguće potpuno pouzdano reći na koji se od više mogućih događaja iz stvarnog svijeta odnose. Razmotrimo sljedeći primjer spominjanja događaja u tekstu:

(1) *Vettel won the race on Sunday.*

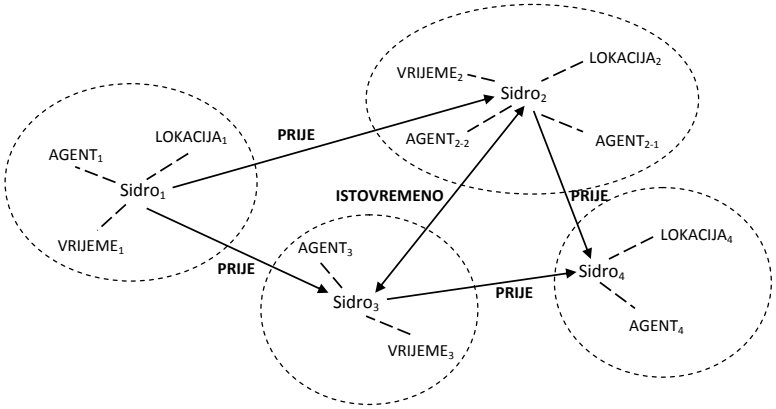
Iz spominjanja događaja nije jasno o kojoj se utrci radi niti koje je točno nedjelje Vettel pobijedio na utrci. Nadalje, lokacija utrke (odnosno Vettelove pobjede) uopće nije poznata. Ovo spominjanje ne određuje jednoznačno događaj stvarnog svijeta budući da se može odnositi na bilo koju Vettelovu pobjedu koja se dogodila u bilo koju nedjelju (a takvih događaja je mnogo, budući da je Vettel svjetski prvak, a utrke Formule 1 održavaju se nedjeljom).

Događaji se u stvarnom svijetu ne odvijaju neovisno jedni o drugima, već se nalaze u različitim međusobnim odnosima. Svaka dva događaja u izvjesnom su vremenskom odnosu (npr. prvi događaj je završio prije početka drugoga; prvi događaj je počeo i završio za vrijeme trajanja drugoga; oba događaja odvijaju se istovremeno i sl.). Osim vremenskog odnosa koji se za dva događaja stvarnog svijeta uvijek može uspostaviti jer je događaj vremenski određen koncept, dva događaja mogu se odvijati na istoj lokaciji, imati zajedničke sudionike, jedan može uzrokovati drugoga i dr. Međutim, kao što je na temelju spominjanja događaja u tekstu često teško točno odrediti o kojem se događaju stvarnog svijeta radi, tako je često teško samo na temelju spominjanja događaja u tekstu odrediti odnose među događajima stvarnog svijeta (npr. vremenski odnos između događaja). Razmotrimo sljedeći primjer s dva spominjanja događaja:

(2) *A car bomb exploded in Baiyaa and another bomb blasted the suburbs of Abu Ghraib.*

Na temelju spominjanja događaja u tekstu nije moguće sa sigurnošću utvrditi vremenski slijed između eksplozije autobombe u distriktu Baiyaa i eksplozije bombe u predgrađu Abu Ghraib. Moguće je, primjerice, da se prva eksplozija dogodila prije druge (ili obrnuto), kao što je moguće i da su se dogodile istovremeno. Ovi primjeri pokazuju kako su uslijed nepreciznosti i višeznačnosti prirodnog jezika zadatci crpljenja informacija o događajima iznimno zahtjevni.

U području crpljenja informacija, srž pojedinačnog spominjanja događaja čini *sidro događaja*. Sidro događaja riječ je ili fraza koja je nositelj temeljnog značenja događaja (npr. "*Vettel won the race*" ili "*The bomb exploded in Baghdad*"). Uz sidro, spominjanje događaja čine i spominjanja sudionika i okolnosti događaja koje nazivamo *argumentima događaja*. Uobičajeni argumenti događaja su oni koji, primjerice, otkrivaju izvršitelja radnje (agenta) ili pak vrijeme odnosno lokaciju odvijanja događaja. Slika 1.1 ilustrira jedan mogući način strukturiranja informacija o događajima iz teksta koji sadrži spominjanja događaja (sidra i argumente) te vremenske odnose među njima. Za domene tekstova u kojima je događaj središnji koncept (npr. novinski tekstovi), strukturiranje informacija o događajima temelj je infrastrukture potrebne za rješavanje zadataka na višoj razini (npr. pretraživanje informacija). Važno je istaknuti



Slika 1.1: Primjer strukture koja sadržava informacije o događajima ekstrahirane iz teksta.

kako je informacije o događajima moguće strukturirati na mnoštvo različitih načina i to (1) odabirom vrsta spominjanja događaja od interesa (npr. samo podskup činjeničnih spominjanja događaja, tj. spominjanja onih događaja za koje smo sigurni da su se doista i ostvarili), (2) odabirom vrsta argumenata događaja (npr. veliki broj semantički specifičnih vrsta argumenata ili pak mali broj semantički općenitih vrsta argumenata) te (3) odabirom vrsta odnosa među događajima (npr. vremenski odnosi, odnos uzroka i učinka, odnos sadržavanja gdje je jedan događaj dio drugoga i dr.). Određena struktura s odabranim vrstama informacija o događajima koje će se ekstrahirati iz teksta može biti više ili manje pogodna za konkretne zadatke u područjima pretraživanja informacija i obrade prirodnog jezika. Tako je crpljenje vremenskih odnosa među događajima, primjerice, ključno za zaključivanje u primjenama koje se bave izgradnjom kronologija ili narativnih lanaca (Moldovan i dr., 2005; Chambers i Jurafsky, 2008). Ispravno ekstrahirani argumenti događaja, s druge strane, vrlo su važni za razrješavanje koreferencije između spominjanja događaja (engl. *event coreference resolution*), tj. pronalaženje spominjanja koja se odnose na isti događaj stvarnog svijeta (Bejan i Harabagiu, 2010; Glavaš i Šnajder, 2013b). Razmotrimo upit postavljen sustavu za pretraživanje informacija koji opisuje konkretan događaj:

(3) *People protested against Bush in South America*

Dokumenti relevantni za ovaj upit su oni dokumenti u kojima se nalazi spominjanje onog događaja iz stvarnog svijeta na kojeg se odnosi spominjanje događaja iz upita. Usporedba argumenata događaja u ovom je scenariju vrlo bitna – želimo dohvatiti samo one dokumente koji opisuju prosvjede protiv američkog predsjednika Busha (ali ne i prosvjede protiv, primjerice, predsjednika Clintona) i samo one prosvjede koji su se odvijali u Južnoj Americi (npr. u Brazilu ili Argentini, ali ne u, primjerice, Kini ili Sjevernoj Koreji.)

U ovoj uvodnoj raspravi razlikovali smo događaje iz stvarnog svijeta (tzv. izvanjezični događaji) od spominjanja događaja kao jezičnih instanci događaja stvarnog svijeta. Međutim, kako

će se ostatak rada baviti isključivo opisima događaja u tekstu, radi jednostavnosti, spominjanja događaja često će se nazivati samo *događajima*.

1.2 Znanstveni doprinosi

Znanstveni doprinosi ove disertacije su: (1) model grafa događaja kao strukture koja zadržava bitne informacijske aspekte izvanjezičnih događaja, (2) potpuno automatizirani postupak robusnog crpljenja informacija o događajima iz teksta, tj. automatizirani postupak izgradnje grafova događaja iz teksta, (3) algoritmi za mjerenje sličnosti i pronalaženje preklapanja među događajima temeljeni na umnošku grafova događaja i računanju jezgrenih funkcija nad grafovima događaja, (4) primjena i vrednovanje postupaka izgradnje i usporedbe grafova događaja u pretraživanju tekstnih informacija te (5) primjena i vrednovanje postupaka izgradnje grafova događaja na zadatcima sažimanja i pojednostavljivanja tekstova.

Graf događaja je graf koji sadrži informacije o pojedinačnim izvanjezičnim događajima kao i informacije o njihovim međusobnim vremenskim odnosima. Vrhovi grafa događaja predstavljaju spominjanja događaja (sidra događaja i argumente događaja), dok bridovi predstavljaju vremenske odnose između događaja. Jedan graf događaja u pravilu predstavlja jedan tekstni dokument koji opisuje događaje stvarnog svijeta (npr. jedan novinski članak). Grafovi događaja omogućuju da se događaji na razini dokumenata predstave pomoću spominjanja događaja na rečeničnoj razini.

Grafovi događaja grade se iz čistog teksta potpuno automatiziranim postupkom koji uključuje crpljenje spominjanja događaja pomoću modela nadziranog strojnog učenja, crpljenje argumenata događaja korištenjem skupa sintaktički i semantički motiviranih pravila te određivanje vremenskih odnosa među događajima korištenjem modela nadziranog strojnog učenja. Za modele nadziranog strojnog učenja koji služe crpljenju spominjanja događaja i određivanju vremenskih odnosa među njima koristi se opsežan skup značajki koje opisuju događaje na leksičkoj, sintaktičkoj i semantičkoj razini. Osim automatiziranog postupka izgradnje grafova događaja, doprinos predstavljaju i predložene mjere za vrednovanje ukupne kakvoće automatski izgrađenih grafova događaja.

Kako bi se otkrili dokumenti koji raspravljaju o istim ili povezanim događajima potrebno je imati mehanizme za pronalaženje preklapanja i mjerenje sličnosti između grafova događaja izgrađenih nad dokumentima. Pronalaženje preklapanja među grafovima događaja obavlja se metodom temeljenom na semantičkom proširenju operacije umnoška grafova, dok se mjerenje sličnosti među grafovima događaja ostvaruje korištenjem jezgrenih funkcija nad grafovima (engl. *graph kernels*), koje predstavljaju računalno učinkovitu alternativu tradicionalnim metodama za pronalaženje sličnosti među grafovima. Računanje semantički proširenog umnoška grafova te računanje jezgrenih funkcija nad grafovima oslanjaju se na uparivanje semantički po-

dudarnih vrhova grafova. Budući da vrhovi grafova događaja predstavljaju spominjanja događaja, podudarni vrhovi su oni koji određuju koreferentna spominjanja događaja, tj. spominjanja koja opisuju isti događaj stvarnog svijeta. Stoga je razvijen model nadziranog strojnog učenja za razrješavanje koreferencije događaja koji identificira parove spominjanja događaja koja opisuju isti događaj stvarnog svijeta. Model za razrješavanje koreferencije događaja temelji se na semantičkoj usporedbi sidara i podudarnih argumenata spominjanja događaja.

Kakvoća postupka izgradnje grafova događaja vrednuje se intrinzično i ekstrinzično. Intrinzično vrednovanje provodi se na dva načina: (1)vrednovanjem svih pojedinačnih modela koji čine postupak izgradnje grafova događaja i (2) vrednovanjem kakvoće cjelokupnih grafova događaja usporedbom s referentnim, ručno izgrađenim grafovima događaja. Ekstrinzično vrednovanje učinkovitosti postupaka izgradnje grafova događaja te algoritama za njihovu usporedbu procjenjujemo na zadatcima pretraživanja informacija i obrade prirodnog jezika. U području pretraživanja informacija grafove događaja koristimo na dvama zadatcima – na zadatku pronalaženja dokumenata koji opisuju iste ili slične događaje te na zadatku pretraživanju dokumenata za upite koji sadrže opise događaja. Ekstrinzično vrednovanje na navedenim zadatcima, gdje postupci pretraživanja informacija temeljeni na grafovima događaja ostvaruju značajno bolje rezultate od tradicionalnih modela pretraživanja informacija, pokazuje opravdanost postupaka izgradnje i usporedbe grafova događaja. U području obrade prirodnog jezika, grafove događaja odnosno strukturirane informacije o događajima koristimo kao temelj algoritama za zadatke sažimanja grupa tematski povezanih dokumenata (engl. *multi-document summarization*) te za pojednostavljivanje novinskih članaka (engl. *text simplification*) za osobe s poteškoćama u čitanju. I na ovim zadatcima rezultati vrednovanja algoritama koji se oslanjanju na grafove događaja opravdavaju automatizirano strukturiranje informacija o događajima iz teksta.

Za potrebe izgradnje modela nadziranog strojnog učenja potrebno je imati ručno označene tekstne podatke. Budući da je ručno označavanje tekstnih zbirki u pravilu vrlo naporan i skup postupak, uz prethodno navedene doprinose, kao poseban doprinos se ističe velika zbirka novinskih tekstova ručno označena spominjanjima događaja. Označena zbirka tri je puta veća od zbirke TimeBank (Pustejovsky i dr., 2003a), koja se standardno koristi za razvoj postupaka za prepoznavanje događaja i vremenskih odnosa u tekstu. Štoviše, za potrebe vrednovanja, ručno su izgrađeni cjelokupni grafovi događaja za nezanemariv podskup dokumenata novoizgrađene zbirke.

1.3 Struktura rada

Disertacija je podijeljena u tri dijela. Uvodni dio (poglavlja dva i tri) analizira manifestaciju događaja i njihova vremenskog aspekta u prirodnom jeziku te povezana istraživanja u području crpljenja informacija o događajima. Drugi je dio (poglavlja četiri, pet, šest i sedam) meto-

dološke prirode i pokriva glavne doprinose disertacije – model grafa događaja, automatizirani postupak izgradnje grafova događaja, algoritme za usporedbu grafova događaja, te intrinzično vrednovanje kakvoće automatski izgrađenih grafova događaja. Treći se dio bavi primjenama grafova događaja u pretraživanju informacija (poglavlje osam) i obradi prirodnog jezika (poglavlje devet). Konačno, u poglavlju je deset dana analiza provedenog istraživanja i smjerova za buduća istraživanja koji se otvaraju na temelju ostvarenoga u okviru disertacije.

Drugo poglavlje analizira koncept događaja iz filozofskog kuta, pri čemu je poseban naglasak stavljen na vremenske aspekte događaja. Dan je pregled istraživanja koja analiziraju metafiziku događaja i njihovih vremenskih aspekata, ali i istraživanja koja se bave manifestacijom događaja u prirodnom jeziku. Razumijevanje načina na koji se događaji i njihovi vremenski odnosi ostvaruju u jeziku vrlo je važno za osmišljavanje računalnih modela za crpljenje informacija o događajima iz teksta.

Treće poglavlje daje pregled srodnih istraživanja u području crpljenja događaja i vremenskih informacija, od ranih pristupa temeljenih na predlošcima do modela temeljenih na nadziranom strojnom učenju. Ujedno se razmatraju i modeli za određivanje semantičkih uloga argumenata predikata (engl. *semantic role labeling*) te modeli za razrješavanje koreferencije spominjanja događaja. Analiziraju se i poveznice s istraživanjima u području otkrivanja i praćenja tema (engl. *topic detection and tracking*, TDT), gdje se događaji razmatraju na razini dokumenata, a ne na rečeničnoj razini.

U četvrtom se poglavlju formalizira koncept grafa događaja te se detaljno opisuju sve njegove komponente kao i pretpostavke koje su uključene u postupak izgradnje takvih grafova. U ovom je poglavlju predstavljana i metoda za pronalaženje preklapanja među grafovima događaja temeljena na semantičkom umnošku grafova, ali i metoda za mjerenje strukturne i semantičke sličnosti grafova događaja korištenjem jezgrenih funkcija nad grafovima.

Peto poglavlje opisuje postupak izgradnje ručno označene tekstne zbirke stvorene za potrebe izgradnje modela nadziranog strojnog učenja kao i za vrednovanje kakvoće automatski izgrađenih grafova događaja. Analizira se međusobno slaganje označivača (engl. *inter-annotator agreement*, IAA) na različitim zadatcima označavanja u cilju procjene gornje granice uspješnosti automatiziranih modela na istim zadatcima.

U šestom je poglavlju detaljno objašnjen postupak izgradnje grafa događaja. Opisani su i vrednovani svi pojedinačni modeli koji su uključeni u izgradnju grafova događaja: (1) modeli nadziranog strojnog učenja za crpljenje i klasifikaciju sidara događaja, (2) model za crpljenje argumenata događaja temeljen na skupu sintaktičkih uzoraka i semantičkih pravila, (3) model za crpljenje i klasifikaciju vremenskih odnosa među događajima te (4) model za prepoznavanje parova koreferentnih spominjanja događaja (tj. spominjanja događaja koja se odnose na isti događaj u stvarnome svijetu).

Sedmo poglavlje pruža uvid u intrinzično vrednovanje kakvoće postupka izgradnje grafova

događaja. Automatski izgrađeni grafovi događaja uspoređuju se s referentnim ručno izgrađenim grafovima, a kakvoća pojedinačnog automatski izgrađenog grafa događaja mjeri se dvjema novim mjerama na temelju njegova tenzorskog umnoška s pripadnim mu referentnim grafom događaja.

Osmo poglavlje opisuje modele za pretraživanje tekstnih informacija koji koriste postupak izgradnje grafova događaja i metode njihove usporedbe iz četvrtog i šestog poglavlja. Metode za pretraživanje informacija temeljene na grafovima događaja vrednuju se na dvama zadatcima. U prvom zadatku, koji pripada području otkrivanja i praćenja tema, cilj je prepoznati parove dokumenata koji opisuju iste događaje kao i rangirati parove dokumenata na temelju sličnosti događaja koje opisuju. Drugi zadatak odnosi se na pretraživanje informacija nad općenitim odnosno tematski ograničenim tekstnim zbirkama, gdje su upiti usmjereni na događaje, tj. cilj je dohvatiti dokumente koji opisuju događaje određene upitom.

Deveto poglavlje pokazuje kako se grafovi događaja mogu uspješno primjeniti i na zadatke u području obrade prirodnog jezika. U ovom poglavlju predstavljen je model za sažimanje grupa tematski povezanih dokumenata koji koristi grafove događaja za razlikovanje značajnih od manje značajnih informacija. Opisan je i model koji koristi pojedinačne događaje (sidra i argumente) da bi se pojednostavnili dijelovi novinskih članaka koji su sintaktički i semantički zahtjevni i tako omogućilo lakše razumijevanje teksta osobama s poteškoćama u čitanju.

Konačno, posljednje, deseto poglavlje zaključuje disertaciju dajući sažet pregled istraživanja predstavljenog u disertaciji i analizirajući značaj ostvarenih znanstvenih doprinosa. Dan je i poseban osvrt na prostor za daljnja istraživanja vezana za crpljenje informacija o događajima, koji se otvara na temelju rezultata ostvarenih u okviru ove disertacije.

Dio I

Događaji

Poglavlje 2

Koncept događaja

Ontološki, *događaj* spada u krovne pojmove (engl. *umbrella notion*) zajedno s ostalim apstraktnim pojmovima koji imaju široku paletu značenja kao što su *entitet*, *objekt*, *odnos*, *svojstvo* i dr. (Casati i Varzi, 2008). Širina značenja krovnih pojmova uzrok je njihovoj obilatoj uporabi u znanosti jer, iako instance krovnih pojmova u različitim područjima znanosti nemaju potpuno podudarno značenje, postoji zajednička značenjska osnova određena upravo krovnim pojmom (Casati i Varzi, 2008). Cilj ovog poglavlja jest dati kratak pregled tretmana pojma događaja u literaturi, odnosno odrediti što čini presjek pojmova događaja u različitim znanstvenim disciplinama. U prvom se dijelu koncept događaja anslizira na metafizičkoj razini, s naglaskom na one aspekte događaja koji su važni za ovu disertaciju (odnos događaja i objekata, vremenski i prostorni aspekt događaja, identitet događaja). U drugom se dijelu podrobnije analizira manifestacija događaja u jeziku kroz istaknute lingvističke teorije vezane uz događaje. U trećem se dijelu detaljnije razmatraju vremenski odnosi među događajima.

2.1 Metafizika događaja

Pojam *događaja* igra vrlo važnu ulogu u mnogim filozofskim i znanstvenim granama: od metafizike do psihologije, od teorije vjerojatnosti i umjetne inteligencije do teorije književnosti i povijesti. Prirodno se postavlja pitanje – ima li što zajedničko u pojmovima događaja u svim tim disciplinama? Nadalje, ako postoji nepromjenjivi dio pojma događaja koji nadilazi granice znanstvenih područja (tzv. kanonski pojam događaja), odgovara li on apriornom shvaćanju pojma događaja u ljudi, tj. konceptu događaja koji postoji neovisno o teorijama koje ga pokušavaju objasniti? Ako postoji kanonski pojam događaja, kakav je njegov odnos s drugim krovnim pojmovima kao što su *objekt* ili *svojstvo*? Imaju li događaji koji odgovaraju kanonskom pojmu događaja svojstva i kojim su to svojstvima takvi događaji određeni? Je li moguće događaju koji odgovara kanonskom pojmu dodijeliti identitet, odnosno postoje li svojstva po kojima je moguće jedan događaj razlikovati od drugih događaja koji su u skladu s istim kanonskim pojmom?

Vrlo je bitno utvrditi stupanj preklapanja između shvaćanja pojma u različitim disciplinama, budući da je pojmove koji imaju različita teorijska utemeljenja potrebno i različito nazivati (Chomsky, 2000).

2.1.1 Razine shvaćanja događaja

U kontekstu ove disertacije prikladnim se čini razlikovanje četiriju vrsta shvaćanja pojma događaja koje su nedavno predložili Casati i Varzi (2008) (podjela nije specifična za pojam događaja, već je prikladna i za druge krovne pojmove):

- Predteorijsko ili intuitivno shvaćanje događaja (engl. *pre-theoretical/common-sense notion of event*) shvaćanje je pojma događaja koje ljudi imaju neovisno o teorijama koje pokušavaju definirati pojam unutar neke discipline (pa makar ta disciplina bila široka poput metafizike);
- Filozofsko shvaćanje događaja (engl. *philosophically refined notion of event*) shvaćanje je pojma događaja koje proizlazi iz promišljanja o nekonzistentnostima intuitivnog (tj. predteorijskog) shvaćanja pojma događaja;
- Znanstveno shvaćanje događaja (engl. *scientifically refined notion of event*) shvaćanje je pojma događaja koje proizlazi iz opažanja u stvarnom svijetu odnosno shvaćanje po kojem pojam ima dodanu vrijednost u vidu pojašnjavanja nekog drugog pojma (npr. dodana vrijednost shvaćanja događaja kao jedinstvene prostorno-vremenske lokacije jest u tome da u izvjesnoj mjeri određuje pojmove *prostor* i *vrijeme*);
- Psihološko ili interno shvaćanje događaja (engl. *psychological/internal notion of event*) shvaćanje je pojma događaja kojim pojedinac sebi objašnjava zašto intuitivno shvaćanje pojma događaja ima upravo onakvu strukturu i onakva svojstva kakva ima.

Istraživanje obuhvaćeno ovom disertacijom dijelom se oslanja na jezično shvaćanje događaja (tj. znanstveno shvaćanje u području lingvistike), a dijelom na interno shvaćanje događaja (npr. neprihvaćanje stanja kao događaja i prihvaćanje promjena stanja kao vrste događaja).

2.1.2 Odnos događaja i objekata

Razmotrimo sada odnos pojma događaja s drugim, u najmanju ruku jednako važnim krovnim pojmom *objekta*. Neki filozofi (Goodman, 1966; Quine, 1986) skloni su izjednačiti događaje i objekte, smatrajući kako na metafizičkoj razini nema konceptualnih razlika između tih pojmova, odnosno da razlike postoje samo na razini percepcije. Goodman (1966) tako, primjerice, smatra da se događaji mogu promatrati kao "nestabilni" objekti, a objekti kao "monotoni" događaji. Ipak, većina filozofskih shvaćanja događaja promatra objekte kao kategoriju različitu od događaja (Brand, 1977; Bennett, 1988; Parsons, 1991). Međutim, i filozofi koji se slažu oko toga da su objekti i događaji različite metafizičke kategorije, ne slažu se u tvrdnji da uvijek

postoji jasna veza između te dvije kategorije, tj. da svaki događaj ima *sudionike* koji su instance kategorije objekt. I dok neki navode događaje poput *kiše* ili *bljeska munje* kao događaje koji nemaju sudionika (Brand, 1977), drugi smatraju da i u takvim slučajevima postoje objekti koji sudjeluju u događaju (npr. kapljice vode u kiši ili fotoni u bljesku munje) (Bennett, 1988). U sklopu ove disertacije prihvaćeno je stajalište po kojem svaki događaj ima sudionike. Posve je drugo pitanje, međutim, koliko je često te sudionike moguće odrediti iz informacija dostupnih u tekstu.

2.1.3 Događaji u prostoru i vremenu

Odnos prema prostoru i vremenu glavni je kriterij po kojem se u filozofskim istraživanjima događaji razlikuju u odnosu na objekte (Quinton, 1979; Hacker, 1982). Prevladava tumačenje po kojem objekti traju tijekom vrijemena, dok događaji imaju ograničeno vremensko trajanje, tj. prostiru se preko ograničenog vremenskog intervala. Vremenska ograničenost toliko je snažno svojstvo događaja da postoje i tumačenja kako nije vrijeme svojstvo događaja, već je događaj svojstvo vremena (Montague, 1969). Odnos događaja i prostora s filozofske je strane još složeniji od odnosa događaja i vremena, jer događaji imaju svojstvo promjenjivosti, dok je prostor (za razliku od vremena) statičan, zbog čega događaj tijekom svog trajanja može mijenjati svoje prostorne koordinate i dimenzije. Problem zrnatosti prostora i vremena odvijanja događaja posebno je izražen u jeziku. Razmotrimo sljedeći primjer:

(4) *Several explosions accompanied demonstrations in Sarajevo on Monday.*

Znamo li u prethodnom primjeru u kojem su se kvartu ili ulici dogodile eksplozije? Znamo li jesu li se eksplozije dogodile ujutro, popodne ili navečer? Nepreciznosti ovog tipa i višeznačnosti koje iz njih proizlaze pripisuju se ipak nedovoljnoj značenjskoj određenosti riječi koje se koriste za opis događaja, a ne tome da događaji apriorno imaju nejasne prostorne granice (Quine, 1985; Lewis, 1986). U ovoj disertaciji prihvaćen je stav po kojem svaki događaj ima precizno prostorno-vremensko prostiranje (Quine, 1985), imajući na umu, međutim, kako je točno prostorno-vremensko prostiranje događaja u pravilu teško (a često i nemoguće) utvrditi iz njegova jezičnog opisa.

2.1.4 Identitet događaja

Konačno, razmotrimo ukratko što događaj čini jedinstvenim, odnosno što određuje identitet događaja. Quine (1985), primjerice, smatra kako je događaj jedinstveno određen svojim prostorno-vremenskim prostiranjem. Drugim riječima, dva se događaja ne mogu odvijati u potpuno isto vrijeme na potpuno istome mjestu. Ova je tvrdnja povezana s Quineovim shvaćanjem zrnatosti prostornih granica događaja koje su vrlo detaljne. Promatramo li prostor (ali i vrijeme)

na detaljnijoj razini zrnatosti, tada lako možemo zamisliti dva događaja koji se odvijaju istovremeno na istome mjestu. Zamislimo, primjerice, kotače automobila koji se *vrtnjom* ujedno i *zagrijavaju*; *vrtnja* i *zagrijavanje* kotača čine se ipak različitim događajima iako se odvijaju istovremeno i na istome mjestu. Kriteriji koji određuju identitet događaja usko su povezani i s razinom zrnatosti na kojoj se sami događaji promatraju. Tako će se jedinstvenost događaja drugačije definirati za događaje koji imaju grubu zrnatost poput objekata (Quine, 1950) u odnosu na događaje koji imaju finu zrnatost poput činjenica (Kim, 1966). U okviru ovog istraživanja smatra se da je događaj jedinstveno određen sudionicima događaja, vrstom interakcije između sudionika događaja (npr. *vrtnja* i *zagrijavanje* kao različite vrste interakcije između tla i kotača) te prostorno-vremenskim prostiranjem događaja. Već je razmotren primjer (*vrtnja* i *zagrijavanje* kotača automobila) koji pokazuje kako događaj nije potpuno određen svojim prostorno-vremenskim prostiranjem. Događaj također nije određen ni samo sudionicima i vrstom interakcije među njima budući da isti sudionici mogu na isti način interagirati na nekom drugom mjestu te ranije ili kasnije u vremenu. Iako su u stvarnom svijetu jedinstveno određeni kombinacijom svih navedenih aspekata (vrsta radnje, sudionici, mjesto, vrijeme), iz teksta je u pravilu vrlo teško odrediti točan identitet događaja iz stvarnog svijeta kojeg spominjanje referencira, ponajprije zbog nedovoljne preciznosti u opisu vremena i mjesta odvijanja događaja (primjer 4). Uslijed nemogućnosti utrđivanja točnog identiteta događaja na temelju njegova spominjanja u tekstu, utvrđivanje koreferencije između dvaju spominjanja događaja neminovno se izvodi s izvjesnom nepouzdanošću (Glavaš i Šnajder, 2013b).

2.2 Događaji u jeziku

Oslanjanjem na koncept događaja, mogu se objasniti mnoge pojave u jeziku (npr. nominalizacije, priložne oznake, aspekti i vremena glagola i dr.) (Davidson, 1967). Kao posljedica ove spoznaje, događaji su u posljednjih pola stoljeća u lingvističkim teorijama dobili mjesto ravnopravno entitetima. Korištenje događaja za objašnjavanje jezičnih fenomena potaknulo je niz radova koji su pokušali izvanjezično (tj. metafizički) utemeljiti koncept događaja (kao što smo vidjeli u prethodnom potpoglavlju). Srećom (s obzirom da na metafizičkoj razini postoje mnoga neslaganja), lingvističke teorije temeljene na događajima nisu strogo povezane s metafizičkim teorijama. Drugim riječima, moguće je analizirati jezične pojavnosti temeljem događaja koristeći predteorijsko ili pak znanstveno shvaćanje događaja umjesto filozofskog shvaćanja događaja (Pianesi i Varzi, 2000; Casati i Varzi, 2008).

2.2.1 Veza između jezika i događaja

U svom poticajnom radu, Davidson (1967) eksplicitno sugerira da mnoge rečenice prirodnog jezika uporabom glagola referenciraju događaje, odnosno instanciraju neki događaj iz ontologije događaja. Logički gledano, rečenice koje sadrže akcijske glagole (npr. *swimmed*, *killed*) zapravo podrazumijevaju egzistencijalnu kvantifikaciju nad skupom događaja. Razmotrimo sljedeću rečenicu:

(5) *Ian Thorpe won a gold medal.*

Davidson smatra kako ova rečenica ne odgovara binarnom predikatu između entiteta "*Ian Thorpe*" i "*gold medal*" (izraz 2.1), već ternarnom predikatu čiji je latentni argument događaj osvajanja zlatne medalje (izraz 2.3).

$$Won(Thorpe, medal) \tag{2.1}$$

$$\exists e(Won(Thorpe, medal, e)) \tag{2.2}$$

Nadalje, budući da nominalizacijom akcijskih glagola u principu ne dolazi do promjene značenja rečenice ili fraze (Ramsey i Moore, 1927), nominalizacije akcijskih glagola također referenciraju događaje (npr. "*Thorpe won*" $\rightarrow$ "*Thorpe's victory*"). Ovim pristupom, kojim akcijske glagole povezuje s događajima, Davidson rješava i problem nestalnog broja argumenata (najčešće priložnih oznaka) koji postoji kada se akcijski glagoli tumače izravno kao logički predikati. Kada bismo rečenicu

(6) *Ian Thorpe won a gold medal in Sydney.*

htjeli predstaviti predikatom, taj bi predikat morao imati 3 argumenta:

$$Won(Thorpe, medal, Syndey), \tag{2.3}$$

što je u suprotnosti s primjerom 5 (i izrazom 2.1) gdje je isti akcijski glagol predstavljen predikatom koji ima dva argumenta. Ako pak akcijski glagol referencira događaj tada se sve priložne oznake akcijskog glagola mogu predstaviti kao logički predikati kojima je događaj kojeg akcijski glagol označava argument (izraz 2.4)

$$\exists e(Won(Thorpe, medal, e) \wedge In(e, Sydney)) \tag{2.4}$$

Proširenje Davidsonove teorije (poznato pod nazivom *neodavidsonovska teorija*) argumente povezuje s događajima putem tzv. tematskih uloga (engl. *thematic roles*), koje su predstavljene logičkim predikatima (Castañeda, 1967; Dowty, 1989). Rečenica iz primjera 6 prema ovoj bi teoriji logički bila prikazana na sljedeći način:

$$\exists e(Won(e) \wedge Agent(Thorpe, e) \wedge Target(medal, e) \wedge Location(Sydney, e)). \tag{2.5}$$

U dijelu disertacije koji se bavi crpljenjem argumenata događaja uvažava se upravo ovakvo tumačenje po kojem akcijske rečenice predstavljaju događaje (a ne predikate), a argumenti su povezani s događajima putem logičkih predikata koji određuju tematske (tj. semantičke) uloge argumenata.

2.2.2 Jezično razlikovanje događaja i stanja

U mnogim se lingvističkim radovima uvodi jasna razlika između događaja i *stanja* (Vendler, 1957; Mourelatos, 1978; Parsons, 1987, 1989; Smith, 1999), odnosno između akcijskih rečenica (npr. primjer 5) i rečenica stanja (npr. primjer 7).

(7) *Thorpe was very happy and proud.*

Kriterij po kojemu se u većini literature događaje razlikuje od stanja jest svojstvo promjenjivosti odnosno dinamičnosti. Dok *stanja* označavaju perzistentnost odnosno nepromjenjivost kroz dug vremenski period, događaji su progresivni i označavaju promjenu kroz vrijeme. U svom poticajnom radu Vendler (1957) dijeli glagole u četiri kategorije prema kriterijima kontinuiranosti (glagoli koji mogu imati kontinurani oblik nasuprot onih koji ne mogu) i teličnosti (engl. *telicity*, pojam koji odgovara glagolskome vidu u hrvatskome jeziku). Prema toj podjeli, stanja su oni glagoli koji ne mogu imati kontinurani oblik, a također su i atelični, odnosno nesvršeni (npr. glagol *knew* u primjeru 8).

(8) *Thorpe knew whole Australia was watching.*

Definiciju po kojoj će se u ovoj disertaciji događaji razlikovati od stanja formulirao je Smith (1999), prema kojemu su stanja situacije koje su identične u svim trenucima vremenskog perioda u kojemu vrijede. S druge strane, događaji su situacije koje su drugačije između svaka dva trenutka vremenskog intervala unutar kojeg se odvijaju.

2.2.3 Činjeničnost spominjanja događaja

Svojstvo istinosti (engl. *veracity*) ili činjeničnosti (engl. *factuality*) događaja spomenutih u tekstu koje govori je li se događaj spomenut u tekstu uistinu i ostvario u stvarnom svijetu ključno je za izvođenje znanja o događajima na temelju teksta. Prirodni jezik obiluje mnoštvom mehanizama kojima možemo izraziti nesigurnost u svoje znanje o pojavama i događajima iz stvarnog svijeta. Takvi mehanizmi zajednički se nazivaju spekulativnim jezikom (engl. *speculative language*) (Wiebe i dr., 2001; Saurí i Pustejovsky, 2012). Izvođenje zaključaka nad spominjanjima događaja za koje znamo da se nisu ostvarili ili ne znamo pouzdano da su se ostvarili u stvarnom svijetu (a koje ćemo zajednički nazivati *nečinjeničnim* događajima) bitno je drugačije u odnosu na izvođenje zaključaka na temelju spominjanja događaja za koje je jasno da su se doista i

ostvarili (Saurí i Pustejovsky, 2012). Činjeničnost događaja igra važnu ulogu i u određivanju vremenskih odnosa među događajima, budući da je nečinjenične događaje teško, a u nekim slučajevima i nemoguće precizno smjestiti na vremensku os (Karttunen i Zaenen, 2005). Po pitanju činjeničnosti, razlikujemo spominjanja događaja koja označavaju da se nešto doista dogodilo u stvarnom svijetu (primjer 6) od spominjanja koja (1) označavaju mogućnost (ili nesigurnost) da se nešto dogodilo (primjer 9) ili (2) označavaju da se događaj nije ostvario (primjer 10).

(9) *Thorpe may have won more than one medal in Sydney.*

(10) *Thorpe didn't participate in 2012 Olympics in London.*

Činjeničnost spominjanja događaja ostvaruje se interakcijom raznih elemenata na mnogim jezičnim razinama (leksičkoj, sintaktičkoj, semantičkoj i diskursnoj). Da neki događaj nije činjeničan često je moguće zaključiti na temelju riječi koje to eksplicitno impliciraju (npr. modalni glagoli ili negacije, kao u primjerima 9 i 10). Ponekad, međutim, nečinjeničnost nije toliko eksplicitna, odnosno proizlazi iz složenih interakcija na razini diskursa (kao u primjeru 11 gdje je tek na temelju druge rečenice jasno da događaj spomenut u prvoj rečenici nije činjeničan).

(11) *Thorpe had won the gold medal. Or at least he thought so until he was disqualified.*

Činjeničnost događaja kombinacija je *polariteta* i *pouzdanosti*. Saurí i Pustejovsky (2012) spominjanja događaja u jeziku po polaritetu dijele na *pozitivna* (označavaju da se nešto dogodilo) i *negativna* (označavaju da se nešto nije dogodilo), a po pouzdanosti na *moguća* (engl. *possible*), *vjerojatna* (engl. *probable*) i *sigurna* (engl. *certain*). U okviru ove disertacije zanimaju nas samo činjenična spominjanja događaja, tj. spominjanja događaja koja su prema prethodnoj podjeli *pozitivna* i *sigurna*.

2.2.4 Teorija semantičkih okvira

Teorija semantičkih okvira (engl. *frame semantics theory*) (Fillmore, 1976) pripada grupi teorija koje jezik pokušavaju objasniti kroz komunikacijske procese radije nego kroz formalizme kakvi su pravopis i gramatika. Takav pristup podrazumijeva da karakterizacija jezika, osim formalnih opisa u vidu rječnika, pravopisa i gramatike, sadrži i opis kognitivnih ili interakcijskih okvira po kojima govornik jezika interpretira svoje okruženje, oblikuje svoje poruke i razumije poruke drugih. Osnovna ideja na kojoj se temelji teorija semantičkih okvira jest da ljudi imaju pohranjen skup shema po kojima strukturiraju, klasificiraju i pohranjuju svoja iskustva i razmišljanja. Prirodno je onda očekivati da se te sheme, nazvane *semantičkim okvirima*, manifestiraju i u jeziku kao primarnom sredstvu putem kojeg ljudi izražavaju svoja iskustva i razmišljanja.

Semantički su okviri u jeziku formalizirani u vidu *gramatike slučajeva* (engl. *case grammar*) (Fillmore, 1967) prema kojoj su semantičke veze među riječima tijekom izgradnje rečenica prepostavljene onim sintaktičkima. Fillmore (1967) uočava kako riječi koje su u istom sintaktičkom odnosu s predikatnim izrazom mogu imati različit semantički odnos prema predikatu. Razmotrimo sljedeće dvije rečenice:

(12) *Ian painted his house.*

(13) *Ian built a house.*

Iako imamo isti sintaktički odnos *predikat-objekt* između "*painted*" i "*house*" u prvoj rečenici te između "*built*" i "*house*" u drugoj, semantičke uloge koje ima objekt "*house*" u tim dvama primjerima očigledno se razlikuju – dok je u prvoj rečenici "*house*" u ulozi trpitelja radnje (tj. "ono što biva oličeno"), u drugoj "*house*" predstavlja rezultat radnje (tj. "ono što biva stvoreno").

(14) [Ian$_{AGENT}$] **painted** [his house$_{PATIENT}$].

(15) [Ian$_{AGENT}$] **built** [a house$_{RESULT}$].

Prema gramatici slučajeva, dakle, jezik se tumači kao skup predikata i njihovih sintaktičkih argumenata koji, ovisno o semantičkom okviru koji instanciraju, imaju određene semantičke uloge. Stoga je semantički okvir u jeziku određen predikatom i skupom semantičkih uloga koje su pridijeljene sintaktičkim argumentima predikata.

(16) [Sardar Patel$_{HELPER}$] **assisted** [Gandhi$_{BENEFITED_PARTY}$] [in the Salt Satyagraha$_{GOAL}$] [in March 1930$_{TIME}$].

U primjeru 16 glagol "*assisted*" instancira semantički okvir ASSISTANCE, kojim su predviđene (a u konkretnom primjeru i ostvarene) semantičke uloge HELPER, BENEFITED_PARTY, GOAL i TIME. U skladu s teorijom semantičkih okvira, razvijeno je i nekoliko jezičnih resursa u kojima su pobrojani mnogi semantički okviri s primjerima njihovih jezičnih instanci, kao što su FrameNet (Baker i dr., 1998) ili NomBank (Meyers i dr., 2004). Iako su predikati najčešće glagoli, važno je primijetiti kako imenice mogu jednako tako instancirati semantičke okvire (Meyers i dr., 2004), kao što je prikazano u primjeru 17.

(17) ...[12%$_{QUANTITY}$] **growth** [in dividends$_{TARGET}$] [next year$_{TIME}$].

Važno je isto tako napomenuti da ne odgovaraju svi predikati koji instanciraju semantičke okvire činjeničnim spominjanjima događaja. U okviru ove disertacije, za predikate (kako glagolske tako i imeničke) koji odgovaraju činjeničnim spominjanjima događaja, argumentima događaja dodjeljivat će se semantičke uloge. Ipak, u cilju robusnog crpljenja informacija o argumentima događajima, usredotočit ćemo se na semantičke uloge koje su zrnatije od onih definiranih postojećim resursima.

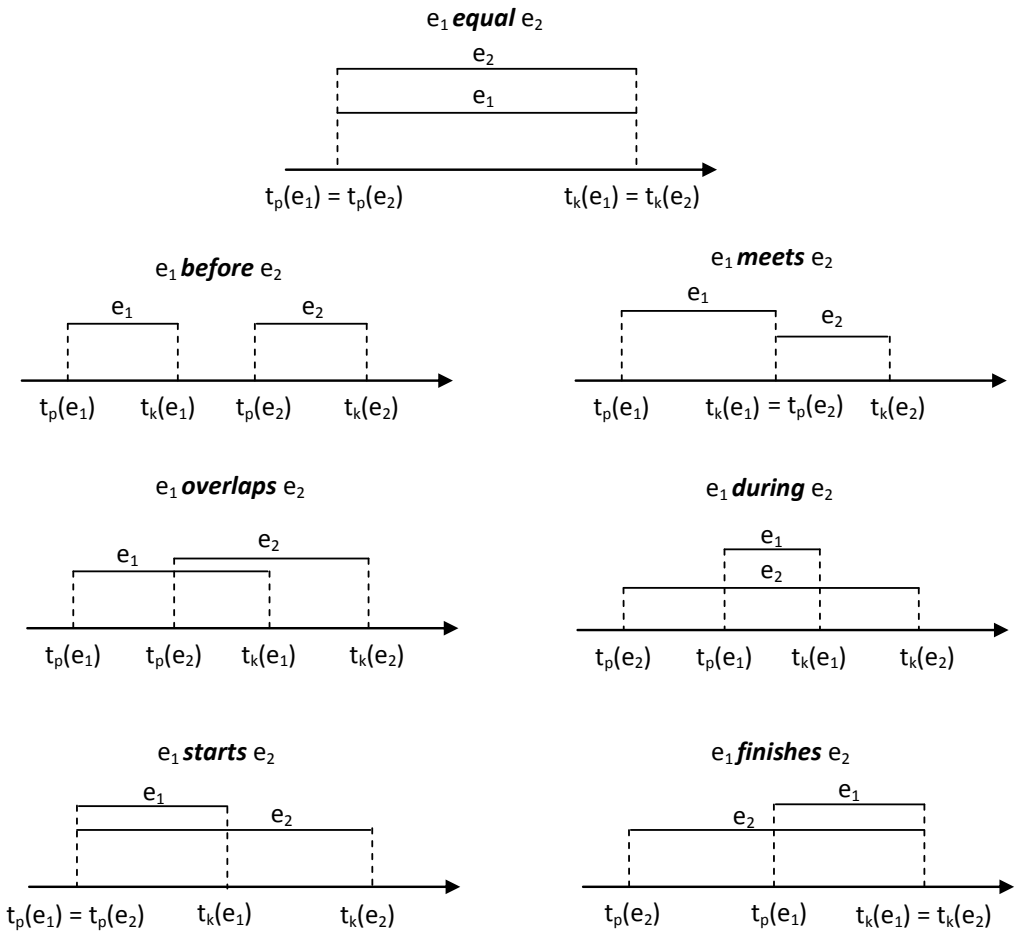
2.3 Predstavljanje vremenskih odnosa među događajima

Suprotno pokušajima izjednačavanja događaja i vremena, ili čak esktremnim tumačenjima po kojima događaji nisu ništa drugo do vremenski intervali s opisom (Van Benthem, 1983), u većini literature događaji se od vremena razlikuju po svojstvu da bivaju opaženi (Gibson, 1975). Vremenska je određenost događaja ipak nepobitna. Događaji se, određeni trenucima početka i kraja, dobro uklapaju u detaljno razrađenu *teoriju vremenskih intervala* (Allen, 1983).

Učinkovita reprezentacija vremena i vremenskih odnosa između događaja, koja bi omogućila automatizirano vremensko zaključivanje, desetljećima je predmet istraživanja u informacijskim znanostima (Bruce, 1972; Allen, 1983; Ladkin, 1987; Ligozat, 1990). Prema Allenu, autoru *de facto* standardno korištene teorije intervalne algebre (Allen, 1983), dobar model za predstavljanje vremena i vremenskih odnosa treba posjedovati sljedeće značajke:

- Modelom treba dozvoliti određenu nepreciznost budući da je većina vremenskog znanja sadržana u relativnim vremenskim odnosima (npr. događaj A *prije* događaja B);
- U model treba uključiti i nepouzdanost informacije, budući da točan vremenski odnos između dva događaja često nije razlučiv (što pogotovo vrijedi za spominjanja događaja u prirodnom jeziku), ali su poznata ograničenja koja sugeriraju kakav bi taj odnos mogao biti;
- Model bi trebao dozvoljavati izvođenje vremenskih zaključaka na različitim razinama zrnatosti (npr. na razini mjeseci ili godina u kontekstu analize povijesti čovječanstva ili pak na razini mikrosekundi ili milisekundi u kontekstu kemijskih reakcija);
- Model treba podržavati svojstvo perzistentnosti događaja. Drugim riječima, za događaj koji je započet pretpostavlja se da traje sve dok ne dobijemo jasnu indikaciju da je uistinu i završio.

Allen predstavlja *teoriju vremenskih intervala* kao teoriju koja zadovoljava prethodne kriterije, a temeljena je na opažanju da svaki događaj možemo rastaviti na događaje "sitnije" zrnatosti. Tako je događaj osvajanja zlatne medalje iz primjera 6 moguće razložiti, primjerice, na (1) Thorpeovo plivanje, (2) Thorpeov ulazak u ciljnu ravninu i (3) dodjelu zlatne medalje Thorpeu. Dodjelu zlatne medalje Thorpeu, potom je, primjerice, moguće razložiti na (1) donošenje zlatne medalje od strane predstavnika plivačke federacije i (2) stavljanje zlatne medalje oko Thorpeovog vrata, itd. Budući da je svaki događaj moguće rastaviti na manje događaje, predstavljanje događaja točkama u vremenu nema smisla. Nasuprot tome, svaki događaj predstavlja se vremenskim intervalom pri čemu su granice intervala definirane kao vremenske točke. Neka $t(e)$ označava vremenski interval događaja e te neka su $t_p(e)$ i $t_k(e)$ točke početka odnosno kraja tog intervala. Allen definira ukupno 13 vremenskih odnosa (šest parova simetričnih odnosa plus odnos vremenske ekvivalencije) koji mogu vrijediti između dva događaja (grafički prikaz dan



Slika 2.1: Grafički prikaz Allenovih intervalnih odnosa

je na slici 2.1):

$$
\begin{aligned}
e_1 \ \ before \ \ e_2 \ &\Leftrightarrow \ t_k(e_1) < t_p(e_2) \\
e_1 \ \ equal \ \ e_2 \ &\Leftrightarrow \ t_p(e_1) = t_p(e_2) \wedge t_k(e_1) = t_k(e_2) \\
e_1 \ \ meets \ \ e_2 \ &\Leftrightarrow \ t_k(e_1) = t_p(e_2) \\
e_1 \ \ overlaps \ \ e_2 \ &\Leftrightarrow \ t_p(e_1) < t_p(e_2) \wedge t_p(e_2) < t_k(e_1) \wedge t_k(e_1) < t_k(e_2) \\
e_1 \ \ during \ \ e_2 \ &\Leftrightarrow \ t_p(e_2) < t_p(e_1) \wedge t_k(e_1) < t_k(e_2) \\
e_1 \ \ starts \ \ e_2 \ &\Leftrightarrow \ t_p(e_2) = t_p(e_1) \wedge t_k(e_1) < t_k(e_2) \\
e_1 \ \ finishes \ \ e_2 \ &\Leftrightarrow \ t_p(e_2) < t_p(e_1) \wedge t_k(e_1) = t_k(e_2)
\end{aligned}
$$

pri čemu $a < b$ označava da točka a vremenski prethodi točki b. Određivanje vremenskih odnosa između događaja na temelju njihovih spominjanja u tekstu u okviru ove disertacije temeljit će se na Allenovoj intervalnoj teoriji, uvažavajući pritom činjenicu da je neke Allenove odnose (npr. *starts* ili *finishes*) iznimno teško prepoznati u tekstu.

Poglavlje 3

Događaji u obradi prirodnog jezika i pretraživanju informacija

Kao što smo vidjeli u prethodnom poglavlju, oko koncepta događaja u filozofiji i lingvistici vode se brojne rasprave i ne postoji jedna, univerzalno prihvaćena teorija koja objašnjava taj koncept. Slično je i u dubinskoj analizi teksta i pretraživanju informacija, gdje također postoje različita poimanja događaja. Ugrubo bi se moglo reći da po tretmanu događaja postoje dva glavna smjera istraživanja: istraživanja u području obrade prirodnog jezika, usmjerena na spominjanja događaja na rečeničnoj razini (Pustejovsky i dr., 2003b; Verhagen i dr., 2010; UzZaman i dr., 2013) te istraživanja u području oktrivanja i praćenja tema, usmjerena na događaje na razini dokumenata (Yang i dr., 1999; Allan, 2002). Ta dva glavna smjera istraživanja razvijani su, međutim, izolirani jedan od drugoga. Središnji doprinos ovog doktorskog rada – graf događaja – povezuje ova dva pogleda na događaje, omogućavajući da se događaji na razini dokumenata predstave koristeći spominjanja događaja na rečeničnoj razini. U skladu s time, u ovom poglavlju razmatramo rezultate dosadašnjih istraživanja, kako postupaka za crpljenje događaja na rečeničnoj razini (odjeljci 3.1 – 3.4), tako i pristupa za analizu događaja na razini dokumenata, odnosno grupa tematski povezanih dokumenata (odjeljak 3.5).

3.1 Crpljenje događaja temeljem predložaka

Počeci crpljenja informacija o događajima iz dokumenata pisanih prirodnim jezikom vežu se uz slijed konferencija pod zajedničkim nazivom Message Understanding Conference (MUC) koje su se održavale krajem 80-ih i početkom 90-ih godina prošlog stoljeća (Grishman i Sundheim, 1996). Konferencije MUC definirale su zadatke crpljenja informacija o događajima prema unaprijed zadanim predlošcima (engl. *templates*). Zadatci na konferencijama MUC uobičajeno su bili vezani za tematski vrlo uske domene (npr. MUC-2 – mornaričke poruke, MUC-3 i MUC-4 – teroristički napadi, MUC-5 – ulaganja u dionice i proizvodnja elektroničkih sklopova) za

koje su unaprijed bili definirani predlošci s unaprijed definiranim brojem informacijskih utora (engl. *slots*) koje je trebalo popuniti specifičnim informacijama iz teksta (npr. za domenu terorističkih napada – napadač, lokacija, vrijeme, meta, sredstvo).

Reprezentativan predstavnik sustava za crpljenje informacija o događajima temeljem predložaka jest sustav LaSIE (Large Scale Information Extraction), razvijen na Sveučilištu u Sheffieldu (Gaizauskas i dr., 1995; Humphreys i dr., 1998), koji je sudjelovao na konferencijama MUC-6 I MUC-7. Interpretacija dokumenta u sustavu LaSIE temelji se na jednostavnoj gramatici izgrađenoj izravno na temelju predložaka događaja konferencije MUC-7 (npr. predložak *launch_event* kojim se opisuju događaji lansiranja svemirskih letjelica):

```
entity(X) ==> object(X) v event(X) v property(X).
object(X) ==> artifact(X) v ...
event(X) ==> launch_event(X) v ...

artifact(X) ==> vehicle(X) v payload(X).
vehicle(X) ==> spacecraft(X) v aircraft(X) v ground_vehicle(X)
               v water_vehicle(X).
spacecraft(X) ==> shuttle(X) v rocket(X).
payload(X) ==> satellite(X) v missile(X) v space_probe(X)
               v material(X) v personnel(X).
```

Semantičke uloge koje su predviđene gramatikom dodatno je moguće preciznije definirati ontologijama koje su zadužene za pojedine vrste argumenata (Humphreys i dr., 1998). Primjerice, uloga "*person*" može se razraditi domenski specifičnom ontologijom kojom se definira da osoba može biti "*pilot*", "*astronaut*" i sl. Cjelokupan sustav temelji se na ručno definiranim pravilima, uključujući i pravila za prepoznavanje imenovanih entiteta u tekstu (Gaizauskas i dr., 1995). Iako su neki dijelovi sustava oblikovani domenski neovisno (npr. osnovne produkcije gramatike za intepretaciju diskursa), za svaku je usku domenu potrebno ručno konfigurirati sustav, što je prilično zahtjevno, kako vremenski tako i što se tiče ljudskog rada.

Sustavi razvijani neposredno nakon konferencija MUC bili su usmjereni uglavnom na proširivanje broja predložaka kako bi mogli crpiti informacije o događajima za veći broj različitih tekstnih domena. Sustav REES (Aone i Ramos-Santacruz, 2000), primjerice, implementira pravila za crpljenje preko 100 tipova događaja i odnosa među entitetima. Primjer predloška za crpljenje informacija o događaju tipa ATTACK dan je u nastavku:

```
<ATTACK TARGET-AP8804160078-12>:=
TYPE: CONFLICT
SUBTYPE: ATTACK TARGET
ATTACKER: [TE for "an Iraqi warplane"]
TARGET: [TE for "the frigate Stark"]
WEAPON: [TE for "missiles"]
```

```
TIME: "May 17, 1987"
PLACE: [TE for "the gulf']
COMMENT: "attacked"
```

U sustavu REES definirana su i pravila koja imaju za cilj prepoznati različita spominjanja istog događaja. Pravila za razrješavanje koreferencije vrlo su jednostavna, a temelje se na izravnoj usporedbi sidara događaja: dva su događaja koreferentna ukoliko su im sidra događaja podudarna ili je jedno nominalizacija drugoga. Cilj razrješavanja koreferencije u sustavu REES jest sjedinjavanje informacija o događaju prepoznatih korištenjem više različitih predložaka. Iako sadrži bitno više predložaka i pokriva veći skup domena, REES je u smislu proširivosti ograničen na potpuno isti način kao i LaSIE – za nove domene potrebno je izraditi nove predloške i pravila za crpljenje informacija. Navedeni je nedostatak ujedno i glavni uzrok napuštanja pristupa temeljenih na predloščima i usmjerenja na statističke pristupe za crpljenje događaja i vremenskih informacija temeljene na modelima nadziranog strojnog učenja.

3.2 Crpljenje događaja na rečeničnoj razini

Razvoj općenitijih pristupa crpljenju informacija o događajima, temeljenih na statističkim modelima strojnog učenja, usko je vezan za pojavu standarda TimeML (Pustejovsky i dr., 2003b), koji je definirao način označavanja događaja i vremenskih odnosa među događajima na rečeničnoj razini, kao i uz izgradnju tekstne zbirke TimeBank (Pustejovsky i dr., 2003a) označene sukladno standardu TimeML. Uskoro su se pojavili javni zadatci za vrednovanje pristupa za crpljenje događaja i vremenskih odnosa iz teksta u skladu s TimeML definicijom događaja (Verhagen i dr., 2007, 2010; UzZaman i dr., 2013). Uz istraživanja temeljena na standardu TimeML, posebno se izdvojio i smjer istraživanja usmjeren na učenje narativnih lanaca iz teksta (Chambers i Jurafsky, 2008; Jans i dr., 2012).

3.2.1 Standard TimeML i zbirka TimeBank

TimeML (Pustejovsky i dr., 2003b) je standard za označavanje događaja i vremenskih informacija u prirodnom jeziku početno osmišljen za potrebe sustava za pretraživanje informacija i automatsko odgovaranje na pitanja (engl. *question answering*, QA) u okviru istraživačkog programa AQUIANT.[1] Standard TimeML predstavlja prvu formalnu specifikaciju za označavanje spominjanja događaja i vremenskih informacija u prirodnom jeziku. U ovom dijelu bit će predstavljeni oni elementi standarda TimeML koji su relevantni za ovu doktorsku disertaciju.

TimeML standardom jasno se definiraju tri temeljna koncepta:

1. Sustavno se definira širok spektar jezičnih izraza kojima se *događaji* ukorjenjuju u tekst;

[1] http://www-nlpir.nist.gov/projects/aquaint/index.html

2. Sustavno se definiraju *vremenski izrazi*, uključujući i relativne vremenske izraze koji nepotpuno određuju trenutak ili interval stvarnog vremena (npr. "*three weeks before*" nasuprot "*2013-31-12*");
3. Određuje se vremenski redoslijed događaja na temelju *vremenskih odnosa* koji se označavaju između izraza koji označavaju događaje.

U standardu TimeML događajima se smatraju svi izrazi koji opisuju da se nešto odvilo ili ostvarilo (engl. *happened or occurred*), bez obzira na dužinu trajanja radnje. Događajima se također smatraju i svi predikati koji opisuju *stanja* (engl. *states*) ili *okolnosti* (engl. *circumstances*), a koji vrijede u nekom vremenskom intervalu. Događajima se ne smatraju jedino predikati koji označavaju radnje koje imaju generički karakter, tj. izrazi koji se odnose na cijele razrede situacija, a ne na konkretne situacije (npr. "*Living is hard with a small salary*"). TimeML ne uvodi ograničenja na događaje u vidu vrsta riječi kojima su događaji predstavljeni. Izrazi kojima su predstavljeni događaji se, po standardu TimeML, u načelu mogu sastojati od jedne ili više riječi.[2] Standard TimeML izraze događaja dijeli u sedam lingvistički motiviranih semantičkih razreda:

- OCCURRENCE – najširi razred događaja koji obuhvaća sve što se odvilo i ostvarilo, a da nije obuhvaćeno definicijom nekog od ostalih razreda (npr. *play*, *murder*, *buy*);
- STATE – razred koji sadrži situacije koje nepromijenjeno vrijede u nekom promatranom vremenskom intervalu (npr. *kidnapped*, *stolen*, *fear*);
- REPORTING – razred u koji spadaju događaji koji eksplicitno označavaju nečije izražavanje, najčešće u usmenom obliku (npr. *say*, *tell*, *voice*, *report*);
- INTENTIONAL ACTION (I-ACTION) – razred događaja koji gramatički izravno otvaraju mjesto drugom događaju (npr. *try*, *offer*, *attempt*);
- INTENTIONAL STATE (I-STATE) – razred događaja koji obuhvaća unutrašnja stanja pojedinca (engl. *desire, believe, think*);
- ASPECTUAL – u ovom razredu nalaze se događaji koji označavaju vremenski aspekt (npr. početak ili kraj) drugog događaja (npr. *start*, *end*, *continue*);
- PERCEPTION – razred koji sadrži događaje koji se odnose na ljudske osjete (npr. *see*, *feel*, *taste*).

Standardom TimeML predviđene su tri vrste odnosa između događaja, odnosno između događaja i vremenskih izraza:

- Veze TLINK označavaju vremenski odnos između dva događaja ili između događaja i vremenskog izraza;[3]
- Veze SLINK označavaju odnos nadređenosti/podređenosti između dva događaja (tj. je-

[2] Iako po standardu TimeML događaji mogu biti višerječni izrazi, u zbirci TimeBank (najveća zbirka označena prema standardu TimeML) čak 99.63% označenih događaja jednorječni su izrazi (Boguraev i Ando, 2005).

[3] Tvrdnja nije u potpunosti točna jer veze razreda IDENTICAL označavaju koreferenciju između spominjanja događaja, a ne njihov vremenski odnos (iako koreferencija, naravno, podrazumijeva vremensku podudarnost).

dan događaj sadrži drugi);

- Veze ALINK označavaju aspektualni odnos između događaja tipa ASPECTUAL i događaja čiji vremenski aspekt taj događaj definira.

Vremenski odnosi koje određuju TLINK veze usko su povezani s Allenovim intervalnim odnosima (Allen, 1983) predstavljenima u drugome poglavlju:

- Vrsta veze BEFORE/AFTER označava da je jedan događaj završio *prije* nego je drugi započeo odnosno, ekvivalentno, da je drugi započeo *nakon* što je prvi završio (npr. "*Thorpe had won three golds before losing in the relay race*"). Ove vrste veza odgovaraju istoimenim Allenovim intervalnim odnosima.
- Vrsta veze IMMEDIATELY_BEFORE/IMMEDIATELY_AFTER označava da jedan od događaja završava točno u trenutku u kojemu drugi počinje (npr. "*The rebels surrendered when soldiers entered the house*"). Ova vrsta TLINK veze odgovara Allenovom intervalnom odnosu MEETS/MET_BY.
- Vrsta veze SIMULTANEOUS označava da se dva događaja odvijaju istovremeno, odnosno da su im početak i završetak podudarni (npr. "*The crowd was cheering while Thorpe was swimming.*"). Ova veza odgovara Allenovom intervalnom odnosu EQUAL;
- Vrsta veze IDENTICAL označava da se dva spominjanja događaja u tekstu odnose na isti događaj iz stvarnog svijeta. Drugim riječima, takva veza zapravo ne označava vremenski odnos, već povezuje dva koreferentna spominjanja događaja (iako veza IDENTICAL, naravno, implicira i vezu SIMULTANEOUS budući da spominjanja određuju isti događaj stvarnog svijeta);
- INCLUDES/INCLUDED_BY vrsta veze označava da je jedan od događaja počeo i završio za vrijeme trajanja drugoga ("*Thorpe won the gold during the deep personal crisis*"). Ova vrsta veze odgovara Allenovom intervalnom odnosu DURING/COVERS;
- Vrsta veze BEGINS/BEGUN_BY označava da je jedan događaj početak drugog događaja ili vremenskog intervala ("*Thorpe has been winning medals on a big scene since the World Championship in Barcelona in 2005*"). Ova veza odgovara Allenovom intervalnom odnosu STARTS/STARTED_BY;
- Vrsta veze ENDS/ENDED_BY označava da jedan događaj predstavlja završetak drugog događaja ili vremenskog intervala ("*Thorpe had been winning medals until he severely injured his knee*"). Ova veza odgovara Allenovom intervalnom odnosu FINISHES/ FINISHED_BY.

Kako bi se omogućili statistički pristupi crpljenju događaja i vremenskih odnosa među njima, prema standardu TimeML ručno je označena tekstna zbirka TimeBank koja se sastoji od 300 dokumenata i približno 100 tisuća pojavnica (engl. *tokens*) (Pustejovsky i dr., 2003a). Zbirka sadrži ukupno 7570 označenih spominjanja događaja, 1423 označena vremenska izraza te 5132 označene vremenske veze između parova događaja odnosno između događaja i vremen-

skih izraza. Razdioba označenih događaja po lingvistički motiviranim razredima, međutim, vrlo je nejednolika. Samo na razred OCCURRENCE otpada preko 50% označenih događaja, dok na tri najbrojnija razreda (OCCURRENCE, STATE i REPORTING) zajedno otpada oko 80% ukupno označenih događaja zbirke. S druge strane, za najmanje zastupljen razred (PERCEPTION) zbirka sadrži svega 40 označenih primjera. Slična neuravnotežena raspodjela uočljiva je i kod vremenskih veza gdje najzastupljeniji razred (BEFORE) sadrži 1183 primjera, dok najmanje zastupljeni razredi (BEGINS, ENDS, I-AFTER, I-BEFORE) sadrže svega između 30 i 70 označenih primjera. Skromna veličina zbirke TimeBank te neravnomjerna raspodjela primjera po vrstama događaja i vremenskih odnosa (tj. vrlo maleni broj primjera za neke od razreda događaja i vremenskih odnosa) čine tu zbirku manje prikladnom za izgradnju modela koji služe crpljenju događaja i vremenskih odnosa među događajima metodama nadziranog strojnog učenja (Boguraev i Ando, 2005).

3.2.2 Evaluacijske kampanje

Evaluacijska kampanja naziv je za skup usko usmjerenih i međusobno povezanih javno objavljenih zadataka. Zadatci uobičajeno imaju jasno definirane mjere za vrednovanje po kojima se određuje uspješnost modela razvijenih od strane sudionika kampanje. Važnost evaluacijskih kampanja proizlazi iz kontroliranih uvjeta u kojima se provode eksperimentalna vrednovanja – svi sudionici vrednuju modele na istim podacima i istim mjerama vrednovanja što omogućava izravnu usporedbu uspješnosti različitih pristupa. Na ovom je mjestu dan pregled najvažnijih evaluacijskih kampanja u području crpljenja događaja i vremenskih informacija.

ACE

Program za automatizirano crpljenje sadržaja (engl. *automated content extraction*) ACE imao je za cilj razviti tehnologije koje iz teksta crpe informacije o entitetima, njihovim odnosima te događajima u kojima entiteti sudjeluju (Doddington i dr., 2004). Jedna od glavnih odrednica programa je, dakle, bila usmjerenost na crpljenje informacija o događajima. Poput pristupa crpljenju događaja temeljenih na predlošcima (v. odjeljak 3.1) i ACE zadatci bili su ograničeni specifičnim tekstnim domenama (npr. terorizam). Kampanje ACE05 (ACE, 2005) i ACE07 (ACE, 2007) uključivale su tri zadatka vezana za događaje: (1) prepoznavanje sidara događaja te određivanje njihovih semantičkih razreda (npr. CONFLICT) i podrazreda (npr. ATTACK), (2) određivanje argumenata događaja te prepoznavanje njihovih semantičkih uloga i (3) prepoznavanje koreferentnih spominjanja događaja. Skupovi dopuštenih semantičkih razreda i podrazreda događaja kao i skupovi semantičkih uloga argumenata unaprijed su bili određeni za pojedine tekstne domene. Kandidati za argumente događaja bili su ograničeni na imenovane entitete te vremenske i numeričke izraze.

Jedan od prvih pristupa crpljenju događaja iz teksta temeljen na nadziranom strojnom učenju (Ahn, 2006) bio je usmjeren upravo na zadatke kampanje ACE05. Ahn (2006) koristi klasifikator maksimalne entropije sa skupom jednostavnih leksičkih značajki za prepoznavanje i klasifikaciju sidara događaja.

TempEval

Kampanje TempEval bile su izravno potaknute pojavom standarda TimeML i zbirke TimeBank. Prva TempEval evaluacijska kampanja (Verhagen i dr., 2007) obuhvaćala je tri različita zadatka određivanja vremenskih odnosa: određivanje vremenskih odnosa između događaja i vremenskih izraza, određivanje vremenskih odnosa između događaja i vremena nastanka dokumenta (engl. *document creation time*, DCT) te određivanje vremenskih odnosa između glavnih događaja susjednih rečenica (ovaj će zadatak u kampanji TempEval-2 biti označen kao zadatak E). Događaji i vremenski izrazi ručno su označeni (tj. korištena je izravno zbirka TimeBank).

Uz prethodna tri zadatka, druga evaluacijska kampanja (TempEval-2) je uvela još tri nova zadatka: crpljenje sidara događaja (i određivanje vrste događaja), crpljenje vremenskih izraza te određivanje vremenskih odnosa između događaja iste rečenice gdje jedan od njih sintaktički upravlja drugime (zadatak F). Na zadatku crpljenja sidara događaja i određivanja semantičkog razreda događaja prema standardu TimeML najbolji su rezultat postigla dva međusobno potpuno različita sustava: (1) sustav temeljen na nadziranom strojnom učenju koji koristi algoritam uvjetnih slučajnih polja (engl. *conditional random fields*, CRF) s mnoštvom lingvistički motiviranih značajki (Llorens i dr., 2010) i (2) sustav temeljen na pravilima koji, koristeći atribute leksičko-semantičke ontologije WordNet (Fellbaum, 2010), filtrira glavne glagole i nominalizacije glagola u rečenicama (Grover i dr., 2010). Na zadatcima određivanja vremenskih odnosa među događajima (glavni događaji susjednih rečenica i događaji iste rečenice gdje jedan sintaktički upravlja drugime), sustav koji je ostvario najbolje rezultate temelji se na nadziranom strojnom učenju algoritmom Markovljevih logičkih mreža (UzZaman i Allen, 2010).

Tijekom prve dvije TempEval kampanje sustavi su vrednovani izdvojeno na pojedinačnim zadatcima. Primjerice, vremenski odnosi su se određivali između ručno označenih događaja i vremenskih izraza, a ne između događaja i vremenskih izraza automatski ekstrahiranih u okviru drugih zadataka. Treća evaluacijska kampanja, TempEval-3, uz vrednovanje sustava na pojedinačnim zadatcima (kao u prethodne dvije kampanje), uvodi i ukupno vrednovanje sustava, povezujući rezultate na pojedinim zadatcima. Mjera vrednovanja nazvana *vremenskom svjesnošću sustava* (engl. *temporal awareness*) (UzZaman i Allen, 2011) zajednički ocjenjuje točnost ekstrahiranih događaja, vremenskih izraza i vremenskih odnosa (između prethodno automatski ekstrahiranih događaja i vremenskih izraza). Vremenski "najsvjesniji" sustav, ClearTK (Bethard, 2013), temelji se na slijedu linearnih modela nadziranog strojnog učenja koji koriste jednostavne morfosintaktičke značajke.

3.2.3 Učenje narativnih lanaca

U posljednih nekoliko godina pojavio se novi smjer istraživanja usmjeren na izgradnju tzv. narativnih lanaca iz dokumenata (Chambers i Jurafsky, 2008, 2009; Jans i dr., 2012). Za razliku od evaluacijskih kampanja TempEval gdje su protagonisti i okolnosti događaja potpuno zanemarene (tj. nema zadatka crpljenja argumenata događaja), kod narativnih lanaca protagonisti igraju ključnu ulogu. Narativni lanac definiran je upravo kao slijed događaja povezanih istim protagonistom.

Chambers i Jurafsky (2008) određuju skup događaja koji čine lanac koristeći točkastu procjenu uzajamne informacije (engl. *pointwise mutual information*) koja se računa na temelju vjerojatnosti pojavljivanja sidara događaja i vjerojatnosti njihova supojavljivanja u velikoj tekstnoj zbirci (a vjerojatnosti se računaju korištenjem procjenitelja najveće izglednosti, tj. na temelju frekvencija pojavljivanja i supojavljivanja događaja u velikoj tekstnoj zbirci). Vremenski slijed između parova događaja lanca potom se određuje korištenjem modela strojnog učenja koji je naučen na oznakama zbirke TimeBank. Budući da grade narativne *lance*, Chambers i Jurafsky (2008) uzimaju u obzir isključivo vremenski odnos prethođenja (veze BEFORE i AFTER iz TimeML standarda). Nastavljajući prethodno istraživanje, Chambers i Jurafsky (2009) uvode narativne sheme kao skupove tipiziranih narativnih lanaca. Događaji koji pripadaju istom tipiziranom narativnom lancu moraju, osim uvjeta čestog supojavljivanja i zajedničkog protagonista, zadovoljiti i uvjet podudarnosti u tipu događaja, pri čemu su tipovi određeni skupovima riječi koje imaju izvjesnu leksičko-semantičku povezanost. Narativne sheme skupovi su narativnih lanaca izgrađenih prateći različite protagoniste.

Nedostatak mjera temeljenih na statistici supojavljivanja kakve Chambers i Jurafsky (2008) koriste za izgradnju narativnih lanaca jest u tome da za većinu parova glagola postoji vrlo malo primjera u zbirci dokumenata u kojima se ti glagoli pojavljuju uzastopno. Ovaj problem oskudnosti primjera za parove glagola Jans i dr. (2012) ublažavaju razmatrajući i supojavljivanja između glagola koji nisu nužno uzastopni u narativnom slijedu dokumenta, odnosno parove glagola kod kojih postoji jedan ili dva događaja između njih.

Sa stajališta strukturiranja informacija o svim događajima koji se u dokumentu spominju, pristup izgradnje narativnih lanaca ima nekoliko nedostataka. Kao prvo, u narativnim se lancima razmatraju samo događaji predstavljeni glagolima. Imenička spominjanja događaja (npr. "*elections*", "*match*", "*murder*"), međutim, često mogu biti informacijski ključna za razumijevanje priče i narativne strukture teksta. Nadalje, narativni lanci temelje se na generalizaciji nad velikim tekstnim zbirkama, što ih čini manje prikladnima za detaljno strukturiranje informacija o događajima iz pojedinačnih dokumenata. Konačno, narativnim shemama, koje su definirane kao skupovi narativnih lanaca definiranih protagonistima, nije moguće predstaviti vremenski odnos između dvaju događaja koji nemaju zajedničkih sudionika (npr. "*Obama and Putin <u>met</u> only a week after the <u>meeting</u> between Sarkozy and Merkel in Athens*").

3.2.4 Ostala istraživanja

U okviru evaluacijske kampanje i2b2 (Sun i dr., 2013) razmatrani su zadatci slični onima iz evaluacijske kampanje TempEval-2, no u domeni kliničkih zapisa o pacijentima. Sudionici kampanje iz otpusnih su pisama pacijenata crpili (1) klinički značajne događaje, (2) vremenske izraze (datume, vremena, učestalosti) i (3) vremenske odnose između kliničkih značajnih događaja i vremenskih izraza. Najbolje rezultate u crpljenju događaja ostvarili su pristupi temeljeni na nadziranom strojnom učenju, dok su u prepoznavanju vremenskih izraza najbolji bili sustavi temeljeni na pravilima. Hibridni pristupi koji su kombinirali nadzirano strojno učenje i heuristička pravila najbolje su prepoznavali vremenske odnose između događaja i vremenskih izraza (Sun i dr., 2013).

Prethodno spomenuta istraživanja (uz iznimku kampanje TempEval-3) bila su usmjerena na izolirano određivanje vremenskih relacija između parova događaja. Drugim riječima, pri određivanju vremenskog odnosa za neki par događaja nisu uzimani u obzir vremenski odnosi utvrđeni za druge parove događaja u istom dokumentu. Malo je istraživanja u literaturi usmjereno na izgradnju cjelokupne vremenske strukture dokumenata; iznimka su radovi (Bramsen i dr., 2006) i (Kolomiyets i dr., 2012). Bramsen i dr. (2006) predstavljaju vremensku strukturu dokumenta u obliku usmjerenog acikličkog grafa (engl. *directed acyclic graph*, DAG), pri čemu razmatraju samo odnose vremenskog prethođenja (odnose BEFORE i AFTER). Globalnu strukturu acikličkog usmjerenog grafa dokumenta optimiraju koristeći cjelobrojno linearno programiranje (engl. *integer linear programming*, ILP). Vrhovi u njihovim grafovima, međutim, ne predstavljaju pojedinačna spominjanja događaja, već segmente teksta. Kolomiyets i dr. (2012) grade stabla događaja iz dječjih priča koristeći tehnike ovisnosnog sintaktičkog parsanja (engl. *dependency parsing*). Valja ipak napomenuti kako je u dječjim pričama radnja linearna – događaji se redaju kronološki i u pravilu postoji samo jedan slijed događaja (rijetko postoje grananja u smislu istovremenog odvijanja dvaju ili više događaja). Stoga je vremenska analiza dječjih priča znatno jednostavniji zadatak od istovrsne analize, primjerice, novinskih tekstova.

3.3 Argumenti događaja

U okviru kampanje ACE, crpljenje argumenata događaja predstavljalo je zaseban zadatak. Ahn (2006) koristi klasifikator temeljen na modelu maksimalne entropije kako bi odredio argumente događaja i njihove semantičke uloge. U prvom koraku klasifikator za sve imenovane entitete, sve vremenske izraze i sve numeričke izraze rečenice određuje jesu li argumenti nekog od događaja čija se sidra nalaze u istoj rečenici. U drugom se koraku izrazi prepoznati kao argumenti klasificiraju u jedan od unaprijed određenih razreda semantička uloga (pri tome svaki od (pod)razreda događaja ima vlastiti skup semantičkih uloga). Ovakav pristup crpljenju argumenata događaja ne razmatra argumente događaja predstavljene izrazima koji nisu imenovani

entiteti, vremenski izrazi niti numerički izrazi. Takvi argumenti su, međutim, vrlo česti i nose puno informacija o događajima (npr. "*an old man*", "*in the house*").

Kako su sidra događaja u pravilu predikati rečenica, crpljenje argumenata događaja usko je povezano s prepoznavanjem semantičkih okvira predikata (v. odjeljak 2.2.4) i pridjeljivanjem semantičkih uloga argumentima predikata. Označavanje semantičkih uloga (engl. *semantic role labeling*, SRL) postupak je induciranja semantičke strukture rečenice prepoznavanjem predikata te određivanjem argumenata koji popunjavaju semantičke uloge definirane semantičkim okvirom predikata (Gildea i Jurafsky, 2002; Palmer i dr., 2010). Semantički okviri predikata s označenim semantičkim ulogama argumenata pokazali su se korisnima na zadatcima crpljenja informacija (Surdeanu i dr., 2003; Llorens i dr., 2013) i automatskog odgovaranja na pitanja (Melli i dr., 2006; Moreda i dr., 2011).

Pristupi određivanju semantičkih okvira (odnosno određivanju strukture predikat-argumenti) u pravilu se temelje na jednom od dva resursa – jezičnom resursu FrameNet (Baker i dr., 1998; Ruppenhofer i dr., 2010) ili zbirci sintaktičko-semantičkih stabala PropBank (Kingsbury i Palmer, 2002, 2003). FrameNet je resurs u kojem je potpuno opisano više od 800 različitih semantičkih okvira koji su međusobno semantički povezani u hijerarhiju (Ruppenhofer i dr., 2010). Resurs također sadrži više od 135,000 rečenica u kojima su označeni semantički okviri – predikati i semantičke uloge argumenata. *Meta predikata* (engl. *frame target*) je riječ koja identificira semantički okvir, kojim je pak određen broj argumenata i njihove semantičke uloge. Semantičke uloge argumenata, koje se nazivaju *elementima okvira* (engl. *frame elements*), sitne su zrnatosti (engl. *fine-grained*) i ovise o konkretnom okviru kojem pripadaju. Primjerice, semantički okvir APPLY_HEAT definira semantičke uloge COOK, FOOD i HEATING_INSTRUMENT, a induciran je glagolima poput "*bake*", "*boil*", "*simmer*" ili "*blanch*". Određivanje semantičkih okvira pomoću FrameNeta uobičajeno uključuje sljedeće korake: (1) identifikaciju okvira na temelju mete predikata, (2) identifikaciju izraza koji predstavljaju argumente predikata te (3) dodjeljivanje semantičkih uloga argumentima predikata (Bejan i Hathaway, 2007). Budući da su predikati u velikoj većini slučajeva glagoli, a glagoli su izrazito višeznačne riječi (Miller i dr., 1993), zadatak identifikacije semantičkog okvira koji predikat inducira često je ekvivalentan zadatku razrješavanja višeznačnosti riječi (Stevenson i Wilks, 2003). Kako pristupi razrješavanju višeznačnosti riječi u pravilu bitno lošije razrješavaju višeznačne glagole nego višeznačne imenice i pridjeve (Navigli i Lapata, 2007; Agirre i Soroa, 2009), pristupi označavanju semantičkih uloga temeljeni na teoriji semantičkih okvira razmjerno će često već u prvom koraku (identifikacija semantičkog okvira) odabrati netočan semantički okvir, što automatski povlači pogreške u prepoznavanju argumenata i njihovih semantičkih uloga. Nadalje, iako je FrameNet sa svojih 800 semantičkih okvira prilično velik resurs, postoje mnogi semantički okviri koji njime nisu obuhvaćeni pa su metode za označavanje semantičkih uloga poput (Bejan i Hathaway, 2007) ograničene na prepoznavanje argumenata samo onih okvira obuhvaćenih resursom.

PropBank je u svojoj osnovi zbirka sintaktičkih stabala Penn Treebank (Marcus i dr., 1993), proširena na način da su čvorovi koji predstavljaju argumente predikata dodatno označeni semantičkim ulogama (Kingsbury i Palmer, 2002). Struktura predikata i argumenata na sintaktičkim je stablima označena za količinu teksta od otprilike dva milijuna riječi. Podrazumijeva se da niti jedan predikat nema više od šest argumenata, a argumenti se označavaju oznakama od *Arg0* do *Arg5*. Iako se na prvi pogled može učiniti da to uvjetuje postojanje samo šest različitih semantičkih uloga argumenata, radi se samo o načinu pobrojavanja argumenata. Semantika oznake *ArgX* ovisi o konkretnom predikatu. Predikat *buy*, primjerice, sadrži pet argumenata (koji ne moraju svi biti instancirani u tekstu) koji odgovaraju sljedećim semantičkim ulogama: *Arg0* – BUYER, *Arg1* – THING_BOUGHT, *Arg2* – SELLER, *Arg3* – PRICE_PAID, *Arg4* – BENEFACTIVE. Ipak, zbog ovakvog načina označavanja vrsta argumenata, postupci za određivanje semantičkih uloga argumenata temeljeni na resursu PropBank preskaču korak identifikacije semantičkog okvira koji je nužan kod postupaka temeljenih na resursu FrameNet. Svakom sintaktičkom argumentu predikata automatski je potrebno dodijeliti najviše jednu od oznaka *ArgX*. Pristupi određivanju semantičkih uloga argumenata temeljeni na resursu PropBank počeli su se intenzivno razvijati s pojavom dijeljenog zadatka na konferenciji CoNLL-2005 (Carreras i Màrquez, 2005), gdje je uz zadana sintaktička stabla rečenica i predikate bilo potrebno odrediti koji su izrazi argumenti tih predikata te im dodijeliti jednu od oznaka *Arg0* do *Arg5*. Svih deset sustava koji su sudjelovali u evaluaciji su, bez iznimke, koristili algoritme nadziranog strojnog učenja i mnoštvo sintaktičkih značajki (Surdeanu i Turmo, 2005; Haghighi i dr., 2005; Koomen i dr., 2005). Tipično je korišten pristup koji slijedno povezuje dva klasifikacijska modela: (1) binarni klasifikator koji za svaki pojedini čvor u sintaktičkom stablu označava je li fraza koju čvor predstavlja (a koja je određena listovima podstabla tog čvora) argument danog predikata te (2) višerazredni klasifikator koji svakom čvoru koji je binarni klasifikator proglasio argumentom predikata pridjeljuje jednu od oznaka *ArgX*. Važno je, međutim, primijetiti da je problem identifikacije semantičkog okvira (odnosno problem identifikacije značenja predikata) moguće izbjeći samo u primjenama u kojoj je gruba semantička određenost argumenata putem oznaka *ArgX* dovoljna. U slučaju da je potrebno odrediti točnu semantičku ulogu koju *ArgX* oznaka nosi za konkretan predikat, potrebno je prvo jednoznačno odrediti značenje (engl. *sense*) predikata. Naredna izdanja ovog dijeljenog zadatka na konferenciji CoNLL (Surdeanu i dr., 2008; Hajič i dr., 2009) bila su usmjerena na zajedničko sintaktičko i semantičko parsanje rečenica.

3.4 Razrješavanje koreferencije događaja

Prepoznavanje spominjanja događaja koja se odnose na isti izvanjezični događaj važno je za uspoređivanje dokumenata koji opisuju iste događaje iz stvarnog svijeta, a omogućava i prikupljanje svih informacija o pojedinom događaju budući da različita spominjanja događaja poten-

cijalno sadrže komplementarne informacije o događaju. U primjeru 18, spominjanje "*shelled*" sadrži informaciju o protagonistu događaja ("*Assad's troops*"), dok njegovo koreferentno spominjanje "*attack*" sadrži informaciju o lokaciji odvijanja događaja ("*on Homs*").

(18) *Assad's troops shelled the city. The attack on Homs was the largest bombardment in months.*

Razrješavanje koreferencije događaja prvi su razmatrali Humphreys i dr. (1997), koji spominjanja događaja organiziraju u hijerarhijsku leksičko-semantičku mrežu prema unaprijed zadanoj ontologiji događaja. Koreferenciju spominjanja događaja utvrđuju koristeći mjere temeljene na udaljenosti sidara događaja u prethodno izgrađenoj leksičko-semantičkoj mreži. Bagga i Baldwin (1999) grade tekstne sažetke za svako glagolsko sidro događaja, spajajući rečenice u kojima se pojavljuje to sidro ili neka njegova nominalizacija. Koreferentnima proglašavaju ona spominjanja događaja između čijih sažetaka postoji značajno preklapanje u imenovanim entitetima i vremenskim izrazima. Ahn (2006) predstavlja prvi model za razrješavanje koreferencije događaja temeljen na nadziranom strojnom učenju, pri čemu koristi algoritam maksimizacije entropije i skup značajki koje međusobno uspoređuju prethodno ekstrahirana sidra i argumente događaja.

Bejan i Harabagiu (2010) koreferenciju događaja razrješavaju grupiranjem (nenadzirano strojno učenje) spominjanja događaja na temelju neparametarskih Bayesovih modela (Teh i dr., 2006; Gael i dr., 2008), koji modeliraju podatke apriornim vjerojatnosnim distribucijama s beskonačnim brojem parametara. Za grupiranje se koristi opsežan skup lingvističkih značajki, uključujući i značajke koje uspoređuju argumente istih semantičkih uloga. Određivanje semantičkih okvira odnosno semantičkih uloga argumenata provodi se modelom temeljenim na resursu FrameNet (Bejan i Hathaway, 2007), čime je model za razrješavanje koreferencije u stvari ograničen na ona spominjanja događaja koja odgovaraju predikatima obuhvaćenima tim resursom.

Chen i dr. (2011) predstavljaju radni okvir za razrješavanje koreferencije događaja koji se sastoji od pet različitih modela nadziranog strojnog učenja, gdje je svaki od modela zadužen za parove spominjanja događaja sa specifičnom kombinacijom vrsta riječi (npr. glagol–glagol, glagol–imenica, imenica–imenica). Svih pet modela koriste slične skupove značajki koji uključuju i značajke za usporedbu argumenata događaja. Postupak crpljenja argumenata događaja ograničen je, međutim, samo na modifikatore koji neposredno prethode sidrima događaja te na prijedložne izraze, pa se kod razrješavanja koreferencije zapravo uspoređuje samo podskup stvarnih argumenata događaja.

3.5 Otkrivanje i praćenje tema

Otkrivanje i praćenje tema područje je pretraživanja informacija usmjereno na događaje na razini dokumenata koje ima za cilj organizirati novinske članke što kontinuirano pristižu prema

događajima koje ti članci opisuju (Allan, 2002). Osnovni zadatci sustava za otkrivanje i praćenje tema uključuju kontinuirano praćenje novinskih objava te dojavljivanje pojave članaka koji opisuju *nove događaje*, tj. događaje koji nisu zabilježeni u prethodno pristiglim novinskim člancima. U kontekstu otkrivanja i praćenja tema, događaj se definira kao nešto što se ostvarilo (ili se ostvaruje) na određenom mjestu u određeno vrijeme, zajedno sa svojim nužnim preduvjetima i neminovnim ishodima (Yang i dr., 1999; Fiscus i Doddington, 2002), dok je *tema* (engl. *topic*) definirana kao skup novinskih članaka povezanih zajedničkim izvornim (engl. *seminal*) događajem (Allan, 2002). Primjerice, osnivanje inicijative "U ime obitelji" izvorni je događaj koji pokreće temu referenduma o ustavnoj definiciji braka. Svi novinski članci koji potom raspravljaju o prikupljanju potpisa za referendum, proglašavanju referendumskog pitanja ustavnim, raspisivanju i održavanju referenduma te njegovim rezultatima pripadaju istoj temi.

Istraživanja u području otkrivanja i praćenja tema bila su usmjerena na pet pojedinačnih zadataka čija se rješenja promatraju kao sastavnice jedinstvenog sustava koji bi omogućio organizaciju novinskih članaka na temelju događaja:

- *Segmentacija novosti* (engl. *story segmentation*) odnosi se na raščlambu tekstnih zapisa koji predstavljaju više spojenih novinskih objava na dijelove koji se tiču pojedinačih događaja. Ovaj zadatak važan je isključivo kod transkripcije televizijskih ili radijskih vijesti;
- *Otkrivanje prve objave* (engl. *first story detection*) odnosi se na članke koji započinju nove teme, odnosno na prepoznavanje novinskih članaka koji opisuju nove, prethodno neviđene događaje;
- *Otkrivanje tematskih grupa* (engl. *cluster detection*) jest problem grupiranja novinskih članaka redoslijedom kojim dolaze sukladno temi kojoj pripadaju;
- *Praćenje teme* (engl. *topic tracking*) odnosi se na stalno praćenje trajnih izvora podataka i pronalaženje dodatnih novinskih članaka koji odgovaraju temi prepoznatoj na temelju nekoliko početnih novinskih članaka;
- *Prepoznavanje povezanih članaka* (engl. *story link detection*) jest zadatak u kojem je za dva slučajno odabrana novinska članka potrebno utvrditi pripadaju li istoj temi, tj. jesu li događaji koje ti članci opisuju povezani istim izvornim događajem.

Pristupi grupiranju novinskih objava u pravilu predstavljaju tekst kao vreću riječi (engl. *bag of words*) te koriste tradicionalni model vektorskog prostora (engl. *vector space model*) (Salton i dr., 1975) za mjerenje sličnosti dokumenata. Predstavljajući dokumente kao vreće riječi, zanemaruju se informacije koje proizlaze iz sintaktičkih odnosa ili pak diskursa. Početna istraživanja više su bila usmjerena na modele i algoritme grupiranja nego na način predstavljanja dokumenata. Najuspješniji sustavi u ranoj fazi istraživanja u području koristili su inkrementalni algoritam k-srednjih vrijednosti (engl. *incremental k-means algorithm*) (Walls i dr., 1999), hijerarhijsko grupiranje na temelju prosjeka grupa (engl. *hierarchical group-average clustering*) (Carbonell i dr., 1999), hijerarhijsko aglomerativno grupiranje temeljem jednostruke poveza-

nosti (engl. *single-linkage hierarchical agglomerative clustering*) (Schultz i Liberman, 1999) ili pak složenije statističke modele poput modela beta-binomijalne mješavine (engl. *beta-binomial mixture*) (Lowe, 1999). U kasnijim je radovima ipak uočeno da su neke kategorije riječi informativnije od drugih (Hatzivassiloglou i dr., 2000; Makkonen i dr., 2004; Kumaran i Allan, 2004), pa je veći naglasak stavljen na način predstavljanja dokumenata. Hatzivassiloglou i dr. (2000) grade vektore koji predstavljaju dokumente koristeći samo riječi koje pripadaju imeničkim izrazima (engl. *noun phrases*) ili imenovanim entitetima (engl. *named entities*). Makkonen i dr. (2004) za svaki od četiri vrste izraza – vremenske izraze, lokacijske izraze, imenovane entitete i općenite izraze – grade zasebne vektore riječi, a sličnost dvaju dokumenata određuju kao linearnu kombinaciju četiriju sličnosti između vektora odgovarajućih vrsta izraza. Imenovani entiteti predstavljaju posebno informativan skup riječi za razlikovanje događaja budući da su upravo imenovani entiteti najčešće protagonisti događaja. Metainformacije, poput vremena nastanka dokumenta (engl. *document creation time*, DCT) mogu se iskoristiti za poboljšanje postupaka grupiranja. Atkinson i Van der Goot (2009) kombiniraju DCT s modelom vektorskog prostora za grupiranje, oslanjajući se na pretpostavku da je za vremenski udaljene dokumente vjerojatnost da su povezani istim izvornim događajem manja.

Dio II

Grafovi događaja

Poglavlje 4

Graf događaja

Graf događaja model je kojim se u ovoj disertaciji premošćuje jaz u poimanju i predstavljanju događaja koji postoji između područja crpljenja informacija, gdje se događaji promatraju na razini spominjanja u rečenicama i pretraživanja informacija, gdje se događaji promatraju na razini dokumenata. Preciznije, graf događaja je model koji omogućava da se događaji stvarnog svijeta opisani dokumentima predstave strukturom koja se sastoji od spominjanja događaja na rečeničnoj razini teksta. Predstavljanje događaja stvarnog svijeta u obliku grafova spominjanja događaja u tekstu omogućava nam da događaje stvarnog svijeta analiziramo i uspoređujemo koristeći brojan skup postojećih algoritama za analizu i usporedbu grafova iz područja teorije grafova. Tako se, primjerice, algoritmi za otkrivanje približnog podudaranja podgrafova (engl. *approximate subgraphs matching*) (Cordella i dr., 2004; Tong i dr., 2007) mogu koristiti za otkrivanje dokumenata koji opisuju djelomično podudarne događaje, a jezgrene funkcije nad grafovima (engl. *graph kernels*) (Gärtner i dr., 2003; Borgwardt, 2007) za otkrivanje dokumenata koji opisuju iste događaje (Glavaš i Šnajder, 2013).

4.1 Formalizacija grafa događaja

Graf kao matematički formalizam, ali i kao podatkovna struktura pogodan je za predstavljanje odnosa među konceptima od interesa. U konkretnom slučaju koncepti od interesa su događaji među kojima postoji mnoštvo informativnih odnosa poput vremenskih odnosa, odnosa uzroka i učinka ili odnosa prostorno-vremenskog sadržavanja. Ova se disertacija bavi prvenstveno vremenskim odnosima između događaja s obzirom na inherentnu vremensku određenost događaja (v. odjeljke 2.1.3 i 2.3).

4.1.1 Definicija grafa događaja

Graf događaja označeni je miješani graf (engl. *labeled mixed graph*) u kojem vrhovi predstavljaju spominjanja događaja, a bridovi vremenske odnose i koreferenciju među spominjanjima

događaja. Budući da svaki vrh predstavlja jedno konkretno spominjanje događaja, a svaki brid vremenski odnos ili koreferenciju između dva konkretna spominjanja događaja, svi su vrhovi i bridovi grafa označeni. Drugim riječima, svaki vrh ili brid grafa događaja moguće je razlikovati od svih ostalih vrhova odnosno bridova. Budući da su neki vremenski odnosi među događajima simetrični (npr. SIMULTANEOUS), dok su drugi asimetrični (npr. BEFORE), u grafu događaja možemo imati i usmjerene i neusmjerene bridove, što graf događaja čini miješanim grafom.

Neka je M skup svih spominjanja događaja u nekom dokumentu te neka je R skup svih odnosa od interesa koji su prepoznati među spominjanjima događaja iz skupa M. Graf događaja uređena je petorka:

$$G = (V, E, A, m, r) \tag{4.1}$$

gdje je V skup vrhova grafa, E skup neusmjerenih bridova grafa, A skup usmjerenih bridova (tj. lukova) u grafu, $m : V \rightarrow M$ funkcija označavanja vrhova koja vrhove grafa preslikava u spominjanja događaja, a $r : E \cup A \rightarrow R$ funkcija označavanja bridova koja svakom bridu dodjeljuje vrstu odnosa (npr. vremenski odnos BEFORE) među događajima. Svako spominjanje događaja iz skupa M pripada jednom od semantičkih razreda događaja (npr. OCCURRENCE ili REPORTING), a sastoji se od sidra događaja i skupa argumenata s pridijeljenim semantičkim ulogama. Na slici 4.1 prikazan je graf događaja za sljedeći isječak novinskog teksta:

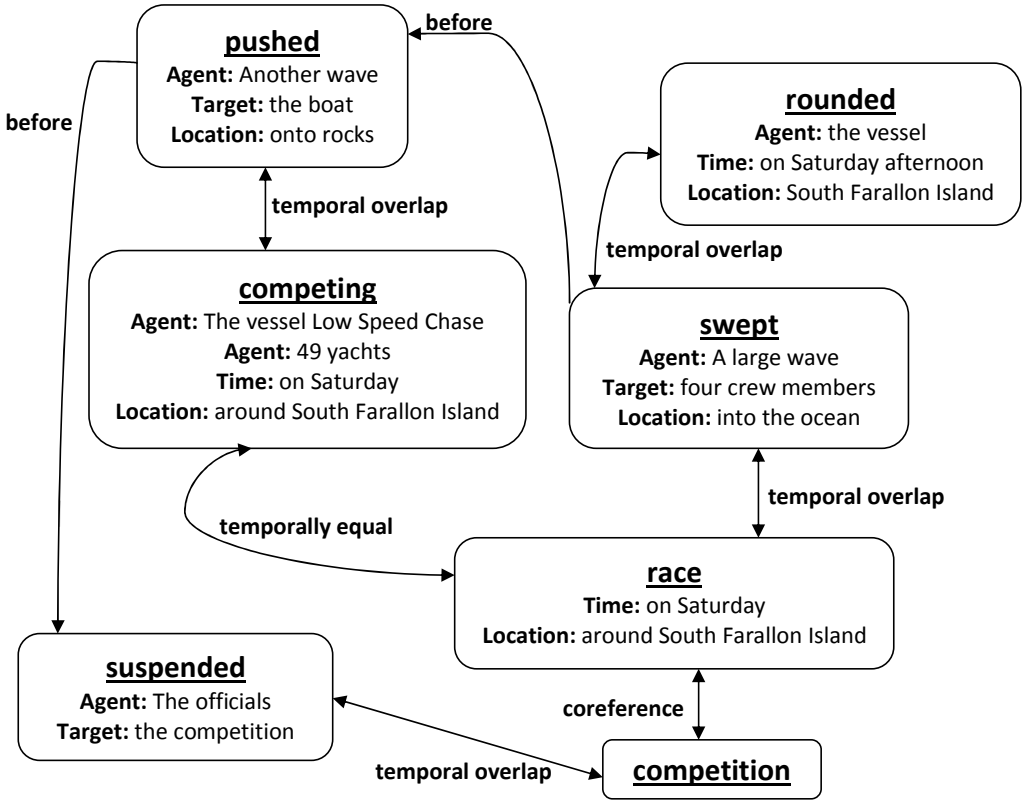
> *The vessel Low Speed Chase was one of 49 yachts competing on Saturday in a race around South Farallon Island. A large wave swept four crew members into the ocean as the vessel rounded South Farallon Island on Saturday afternoon. Another wave pushed the boat onto rocks. The officials immediately suspended the competition.*

Gornja definicija grafa događaja općenita je u smislu da ne postavlja ograničenja na spominjanja događanja niti na odnose među njima, tj. na skupove M i R. Ovakva općenita definicija grafa događaja, međutim, nije izravno upotrebljiva. Za izgradnju grafova događaja koji bi bili izravno upotrebljivi u određenim primjenama potrebno je uvesti dodatne pretpostavke (tj. ograničenja), koje potpuno precizno određuju informacije sadržane u grafu događaja.

4.1.2 Pretpostavke izgradnje grafova događaja

U ovom radu na skup spominjanja događaja M i skup odnosa R među događajima iz skupa M primjenjujemo određene pretpostavke, opisane u nastavku.

Činjeničnost i semantički razredi spominjanja događaja. Informacijske potrebe usmjerene na događaje u pravilu se tiču stvarnih događaja, tj. događaja koji su se doista odvili ili se odvijaju u stvarnom svijetu. Drugim riječima, ljudi puno češće postavljaju upite koji se tiču stvarnih događaja (npr. "*Who killed Osama bin Laden?*") nego hipotetskih (npr. "*When might Obama*



Slika 4.1: Grafički prikaz primjera grafa događaja

resign?") ili negiranih događaja (npr. "*Who did not win a medal in 2012 Olympics?*"). Budući da je krajnji cilj primjena grafova događaja u zadatcima koji uključuju stvarne informacijske potrebe korisnika, vrhovi grafova događaja odgovarat će isključivo činjeničnim spominjanjima događaja. Dakle, nećemo razmatrati spominjanja događaja koja su negirana (npr. "*He did not win*"), hipotetska (npr. "*He may win*"), buduća (npr. "*He will win*") ili su protučinjenične prirode (npr. "*If he had won...*"). Nadalje, slično kao i u standardu TimeML, za uvrštavanje u graf događaja nećemo razmatrati spominjanja događaja generičke prirode koja definiraju vrstu situacije, a ne jedan konkretan događaj (npr. "*It's always hard to lose someone you love*").

Spominjanja događaja svrstavat ćemo u jedan od pet semantičkih razreda: OCCURENCE, I-ACTION, PERCEPTION, REPORTING i STATECHANGE. Prva četiri razreda semantički odgovaraju istoimenim razredima standarda TimeML, dok razredom STATECHANGE razlikujemo, unutar najšireg razreda OCCURRENCE, događaje koji označavaju prijelaz nekog entiteta iz jednog stanja u drugo (npr. "*The CBI index rose 3.2%*"). Preciznije, promjene stanja definiramo kao događaje koji mijenjaju neko intrinzično svojstvo entiteta koji u događaju sudjeluje. Primjerice, spominjanje "*fell*" u rečenici "*The girl fell and started crying*" svrstali bismo u razred OCCURRENCE budući da pad ne mijenja niti jedno intrinzično svojstvo djevojčice (mijenja joj se jedino položaj u prostoru, a to nije njeno intrinzično svojstvo). S druge strane, spominjanje "*fell*" u rečenici "*The P&G stocks fell 5% in early trading*" svrstali bismo u razred STATEC-

HANGE jer označava da se intrinzično svojstvo dionica promijenilo (vrijednost jest intrinzično svojstvo dionice). Mogućnost razlikovanja događaja koji pripadaju razredu STATECHANGE važna je za tekstne domene u kojima se intrinzična svojstva entiteta učestalo mijenjaju (npr. zapisi o pacijentima u bolnicama).

Suprotno standardu TimeML (Pustejovsky i dr., 2003b), događajima ne smatramo predikate koji opisuju stanja i okolnosti pod kojima nešto vrijedi. Stanja su situacije koje se bitno razlikuju od događaja, prije svega po vremenskoj (ne)određenosti (v. odjeljak 2.2.2). U ovoj se disertaciji stanja od događaja razlikuju prema Smithovoj definiciji (1999) po kojoj su, za razliku od događaja (koji se kroz vrijeme mijenjaju), stanja u svakom vremenskom trenutku svoga trajanja jednaka.

Za razliku od standarda TimeML, po kojem sidra događaja načelno mogu biti višerječni izrazi, sidra događaja u spominjanjima koja čine vrhove grafova događaja isključivo su jednorječni izrazi. Iako se u načelu radi o razlici u definiciji spominjanja događaja, stvarne razlike zapravo i nema budući da 99.63% svih spominjanja događaja u zbirci TimeBank otpada na jednorječne izraze (Boguraev i Ando, 2005).

Semantičke uloge argumenata. U cilju ostvarenja robusnog postupka za crpljenje informacija o događajima, ograničit ćemo se na argumente četiriju semantičkih razreda koje otkrivaju ključne informacije o događaju, odnosno daju odgovore na pitanje "*Tko je učinio što kome, kada i gdje?*". Za razliku od pristupa označavanju semantičkih uloga, koji su ograničeni opsegom resursa na kojem se temelje i za koje skup semantičkih uloga ovisi o konkretnom predikatu, naš cilj je prepoznavati argumente četiriju semantičkih uloga grube zrnatosti, a koje imaju svi događaji, odnosno svi predikati – AGENT (odgovara na pitanje "*tko*"), TARGET (odgovara na pitanje "*kome*"), TIME (odgovara na pitanje "*kada*") i LOCATION (odgovara na pitanje "*gdje*").

Odnosi među događajima. Između dva događaja, u načelu, može postojati mnoštvo različitih odnosa. Za svaka dva događaja stvarnog svijeta vrijedi da su u nekom vremenskom odnosu, dok samo za neke parove događaja vrijedi da, primjerice, jedan uzrokuje drugi ili da jedan događaj sadrži drugi (npr. *Summit* zemalja članica G8 sadrži *sastanak* Obame i Putina). Bridove grafova događaja gradit ćemo na temelju dviju vrsta odnosa između događaja: (1) vremenskih odnosa, jer su događaji apriorno vremenski određeni (v. odjeljak 2.3) i (2) odnosa koreferencije spominjanja događaja, jer informacije o jednom te istom događaju u stvarnom svijetu mogu biti raspodijeljene na više koreferentnih spominjanja tog događaja u tekstu. Budući da je već u evaluacijskim kampanjama TempEval (Verhagen i dr., 2007, 2010) prepoznato da je samo na temelju teksta vrlo teško međusobno razlikovati vremenske odnose fine zrnatosti (npr. razlikovati odnos DURING od odnosa BEGINS) te budući da je za izgradnju upotrebljivih grafova događaja potrebno pouzdano crpljenje vremenskih informacija, pri izgradnji grafova događaja

Tablica 4.1: Preslikavanje između vremenskih odnosa u Allenovoj intervalnoj teoriji, standardu TimeML i grafovima događaja

Intervalna teorija	TimeML	Grafovi događaja
MEETS/MET_BY, BEFORE/AFTER	I-BEFORE/I-AFTER, BEFORE/AFTER	BEFORE/AFTER
EQUAL	SIMULTANEOUS, IDENTICAL	EQUAL
COVERS, DURING, OVERLAPS, STARTS, STARTED_BY, FINISHES, FINISHED_BY	INCLUDES, IS_INCLUDED, BEGINS, BEGUN_BY, ENDS, ENDED_BY	OVERLAP

razlikujemo četiri razreda vremenskih odnosa grube zrnatosti: BEFORE, AFTER, OVERLAP i EQUAL. Prva tri razreda preuzeta su iz evaluacijske kampanje TempEval-2 (Verhagen i dr., 2010), dok je vremenska podudarnost događaja (razred EQUAL) dodana jer olakšava zaključivanje nad vremenskom strukturom grafa događaja. Preslikavanje između vremenskih odnosa Allenove intervalne algebre, vremenskih odnosa standarda TimeML i vremenskih odnosa u grafovima događaja dano je u tablici 4.1.

4.2 Usporedba grafova događaja jezgrenim funkcijama

Kako bismo mogli uspoređivati dokumente predstavljene grafovima događaja, odnosno mjeriti koliko su slični po pitanju događaja koje opisuju, potrebna nam je metoda za uspoređivanje grafova, odnosno metoda za mjerenje sličnosti među grafovima. Formalno, problem usporedbe grafova (engl. *graph comparison problem*) svodi se na pronalaženje funkcije

$$s : \mathscr{G} \times \mathscr{G} \to \mathbb{R} \tag{4.2}$$

koja kvantificira sličnost između dvaju ulaznih grafova G i $G' \in \mathscr{G}$. Potrebno je napomenuti kako se, općenito, usporedba grafova može odnositi i na pronalaženje podudarnih dijelova između dva grafa, a ne nužno na mjeru sličnosti. Tradicionalni algoritmi za usporedbu grafova uobičajeno pate od suprapolinomijalne složenosti, što ograničava njihovu primjenjivost pri usporedbi velikih grafova. S druge strane, algoritmi koji imaju polinomijalnu složenost predstavljaju grafove skupom numeričkih značajki, čime zapravo zanemaruju ili u najmanju ruku smanjuju informaciju koja proizlazi iz strukture grafa. Jezgrene funkcije nad grafovima rješavaju oba navedena problema – daju mjeru sličnosti između grafova u polinomijalnom vremenu uvažavajući strukturu i semantiku grafova, tj. bez pojednostavljivanja grafova vektorima značajki.

4.2.1 Tradicionalni pristupi usporedbi grafova

Topološka podudarnost grafova nameće se kao prirodan izbor za mjeru sličnosti – neoznačeni grafovi su podudarni ako i samo ako su izomorfni. Izomorfizam grafova (ili podgrafova zadanih grafova) je, međutim, problem za koji nije poznat algoritam čija je složenost polinomijalna (Garey i Johnson, 1979). Dodatni problem predstavlja i činjenica da potpuna topološka podudarnost grafova u većini stvarnih primjena gotovo da i ne postoji. Stoga su za takve primjene potrebni algoritmi koji mogu prepoznati nepotpunu podudarnost između grafova (engl. *inexact graph matching*), poput algoritama za određivanje *udaljenosti izmjene* (engl. *edit distance*) između grafova. Udaljenost izmjene između dva grafa određena je minimalnim brojem operacija, poput dodavanja, brisanja ili zamjene vrhova i/ili bridova, koje jedan od grafova svode na drugi (Bunke i Allermann, 1983). Ipak, pronalaženje minimalne udaljenosti izmjene grafova je, kao i pronalaženje izomorfnih podgrafova, problem koji nije moguće riješiti u polinomijalnom vremenu.

Problem vremenske učinkovitosti izračuna sličnosti grafova može se riješiti tako da se grafovi predstave vektorima značajki koje opisuju topologiju grafa, a nazivaju se *topološkim deskriptorima* (engl. *topological descriptors*). Topološki deskriptori variraju od vrlo jednostavnih, poput broja vrhova i bridova u grafu ili veličine najveće povezane komponente grafa pa do složenih, poput često korištenog Wienerova indeksa koji odgovara zbroju duljina najkraćih puteva između svih parova vrhova grafa (Wiener, 1947):

$$W(G) = \sum_{v_i \in V_G} \sum_{v_j \in V_G} d(v_i, v_j), \tag{4.3}$$

gdje je V_G skup vrhova grafa G, a $d(v_i, v_j)$ duljina najkraćeg puta ozmeđu vrhova v_i i v_j. Podudarni ili topološki vrlo bliski grafovi imat će i podudarne ili vrlo slične vektore topoloških deskriptora. Nedostatak pristupa temeljenih na vektorima topoloških deskriptora leži u činjenici da ne vrijedi obrat ove tvrdnje – grafovi mogu imati slične vektore topoloških deskriptora, a ne biti podudarni, što je izravna posljedica činjenice da vektori topoloških deskriptora nisu ekspresivni koliko su to izvorni grafovi (tj. bitno različiti grafovi mogu se pretvoriti u iste ili slične vektore značajki).

4.2.2 Jezgrene funkcije

Jezgrene funkcije nad grafovima (engl. *graph kernels*) predstavljaju rješenje za oba prethodno istaknuta problema: (1) računaju sličnost između grafova događaja u polinomijalnom vremenu i pritom (2) koriste svu informaciju sadržanu u grafovima, bez smanjenja ili pojednostavljenja grafova.

Općeniti razvoj jezgrenih funkcija (engl. *kernels*) u vektorskom prostoru započinje s poja-

vom algoritma stroja potpornih vektora (engl. *support vector machines*, SVM) (Cortes i Vapnik, 1995) za binarnu klasifikaciju. U algoritmu SVM, tzv. *jezgreni trik* (engl. *kernel trick*) omogućava da se klasifikacijski problem iz ulaznog prostora u kojem primjeri dvaju razreda nisu linearno odvojivi preseli u prostor značajki veće dimenzionalnosti u kojemu primjeri postaju linearno odvojivi. Cilj je pronaći nelinearno preslikavanje $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^m, m > d$ (gdje je $\mathbb{R}^d$ ulazni prostor dimenzionalnosti d, a $\mathbb{R}^m$ prostor značajki dimenzionalnosti m veće od d) takvo da primjeri postanu linearno odvojivi u prostoru $\mathbb{R}^m$. Ne ulazeći u detalje samog algoritma SVM, skalarni produkt dvaju vektora $\langle \mathbf{x_i}, \mathbf{x_j} \rangle$ u ulaznom prostoru može se zamijeniti skalarnim produktom slika tih vektora u visokodimenzionalnom prostoru, $\langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle$. Jezgrena funkcija $k(\mathbf{x_i}, \mathbf{x_j})$ je funkcija koja odgovara skalarnom produktu vektora u visokodimenzionalnom prostoru $\mathbb{R}^m$, $k(\mathbf{x_i}, \mathbf{x_j}) = \langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle$, a omogućava da se klasifikacijski problem rješava u prostoru više dimenzije bez eksplicitnog izračuna funkcije preslikavanja $\phi(\mathbf{x})$. Ne predstavlja, međutim, svaka funkcija $k(\mathbf{x_i}, \mathbf{x_j})$ skalarni produkt vektora $\mathbf{x_i}$ i $\mathbf{x_j}$ preslikanih u prostor $\mathbb{R}^m$. Da bi funkcija mogla biti korištena kao jezgrena funkcija u algoritmu SVM ili nekom drugom algoritmu iz razreda algoritama jezgrenih strojeva (engl. *kernel machines*), mora zadovoljavati svojstvo pozitivne definitnosti[1] (Schölkopf i Smola, 2002). Važno je, međutim, naglasiti kako je uvjet pozitivne definitnosti jezgrenih funkcija bitan samo kad se koriste za klasifikaciju u algoritmima jezgrenih strojeva. Ukoliko se jezgrene funkcije koriste samo kao mjera sličnosti između dva vektora (ili dvije strukture), kriterij pozitivne definitnosti ne mora nužno biti zadovoljen.

Jezgrene metode, međutim, moguće je primijeniti i na strukture (npr. stabla, grafovi), a ne isključivo na vektorske podatke (Schölkopf i dr., 1999). Korištenjem jezgrenog trika, strukture poput grafova moguće je uspoređivati izravno, tj. bez njihova eksplicitnog pretvaranja u vektore. Jezgrene funkcije nad strukturiranim podacima spadaju u razred R-konvolucijskih jezgrenih funkcija (Haussler, 1999), koje predstavljaju generički radni okvir za usporedbu struktura koje se mogu dekomponirati na manje strukture. U ovoj će se disertaciji za usporedbu grafova događaja koristiti dvije različite jezgrene funkcije: jezgrena funkciju umnoška grafova (engl. *product graph kernel*) (Gärtner i dr., 2003) i jezgrena funkciju težinske dekompozicije grafova (Menchetti i dr., 2005). Ove konkretne jezgrene funkcije odabrane su (1) jer imaju općeniti oblik pogodan za izražavanje semantike događaja i (2) jer su se pokazale vrlo uspješnima u drugim primjenama (npr. na zadatcima iz kemoinformatike) (Menchetti i dr., 2005; Mahé i dr., 2005).

[1] Kaže se da je funkcija pozitivno definitna ukoliko joj je pripadna Gram matrica pozitivno definitna za bilo koji skup vektora $D = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ iz ulaznog prostora χ. Gram matrica G funkcije $k : \chi^2 \mapsto \mathbb{R}$ jest matrica čiji elementi, za skup primjera D, iznose $G_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$.

4.2.3 Jezgrena funkcija umnoška grafova

Jezgrena funkcija umnoška grafova (engl. *product graph kernel*) broji podudarne šetnje koje "slučajni šetač" (engl. *random walker*) napravi nad zadanim grafovima (Gärtner i dr., 2003). Kao što naziv sugerira, ova jezgrena funkcija računa se na temelju umnoška dvaju zadanih grafova (engl. *graph product*). Broj podudarnih šetnji mjeri se na rezultantnom grafu koji nastaje umnoškom dvaju grafova, budući da takav graf sadrži sve šetnje koje su zajedničke ulaznim grafovima.

Umnožak grafova

Umnožak dvaju grafova s neoznačenim vrhovima G i G' jest graf $G_P = G \times G'$ kojemu je skup vrhova Kartezijev umnožak vrhova ulaznih grafova G i G' (Hammack i dr., 2011):

$$V_P = \{(v, v') \mid v \in V_G, v' \in V_{G'}\}. \tag{4.4}$$

Za ulazne grafove koji imaju označene vrhove, kakvi su grafovi događaja, skup vrhova grafa umnoška čine samo oni parovi vrhova $v \in V_G$ i $v' \in V'_G$ koji imaju podudarne oznake, tj.:

$$V_P = \{(v, v') \mid v \in V_G, v' \in V_{G'}, \delta(v, v')\} \tag{4.5}$$

gdje je je $\delta(v, v')$ logički predikat koji je zadovoljen kada vrhovi v i v' imaju podudarne oznake. U konkretnom slučaju, gdje su na ulazu grafovi događaja $G = (V, E, A, m, r)$ i $G' = (V', E', A', m', r')$ u kojima vrhovi predstavljaju spominjanja događaja, logički predikat koji uspoređuje vrhove mora biti predikat koji uspoređuje spominjanja događaja. Kako su spominjanja događaja podudarna ako i samo ako se odnose na isti događaj u stvarnome svijetu, za usporedbu vrhova grafova G i G' koristit ćemo razrješavanje koreferencije između spominjanja događaja, tj. $\delta(v, v') \equiv coref(m(v), m'(v'))$, gdje je $coref(m_1, m_2)$ predikat koji je zadovoljen ako i samo ako su spominjanja događaja m_1 i m_2 koreferentna.

Skup bridova grafa koji nastaje umnoškom dvaju grafova ovisi o vrsti umnoška koji se računa (Hammack i dr., 2011). U okviru ove disertacije razmatramo dva različita umnoška grafova: *tenzorski umnožak* (engl. *tensor graph product*, poznat i kao *direktni* ili *Kroneckerov umnožak* ili kao *konjunkcija grafova*) i *konormalni umnožak* (engl. *conormal graph product*, poznat i kao *ILI-umnožak* ili *disjunkcija grafova*).

U tenzorskom umnošku postoje samo oni bridovi za koje odgovarajući bridovi postoje u oba ulazna grafa, tj.:

$$E_P = \left\{ \big((v, v'), (w, w')\big) \in V_{G_P} \times V_{G_P} \mid (v, w) \in E_G, (v', w') \in E'_G \right\} \tag{4.6}$$

Za grafove koji imaju označene bridove, kakvi su grafovi događaja, postavlja se i dodatni uvjet podudarnosti oznaka odgovarajućih bridova u ulaznim grafovima, tj.:

$$E_P = \Big\{ \big((v,v'),(w,w')\big) \in V_{G_P} \times V_{G_P} \mid (v,w) \in E_G, (v',w') \in E'_G, \delta_e\big((v,w),(v',w')\big) \Big\} \tag{4.7}$$

gdje je je $\delta_e\big((v,w),(v',w')\big)$ logički predikat koji je zadovoljen kada bridovi $(v,w) \in E_G$ i $(v',w') \in E_{G'}$ imaju podudarne oznake. Kada su na ulazu grafovi događaja $G = (V,E,A,m,r)$ i $G' = (V',E',A',m',r')$, logički predikat koji uspoređuje bridove je zadovoljen ako i samo ako bridovi označavaju isti odnos između spominjanja događaja, tj. $\delta_e\big((v,w),(v',w')\big) \equiv r(v,w) = r'(v',w')$.

Za razliku od tenzorskog umnoška, u konormalnom umnošku postoje oni bridovi za koje odgovarajući brid postoji u barem jednome od ulaznih grafova, tj.:

$$E_P = \Big\{ \big((v,v'),(w,w')\big) \in V_{G_P} \times V_{G_P} \mid (v,w) \in E_G \vee (v',w') \in E'_G \Big\} \tag{4.8}$$

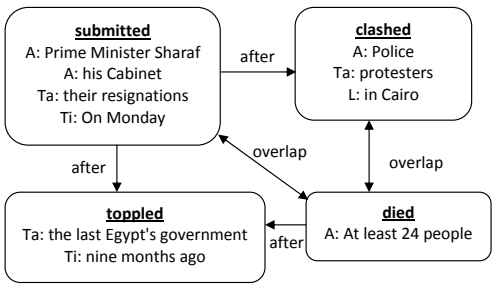
Za bridove konormalnog umnoška grafova događaja za koje odgovarajući bridovi postoje u oba ulazna grafa ne zahtijevamo da ti bridovi imaju podudarne oznake (tj. ne mora vrijediti $r(v,w) = r'(v',w')$).

Promotrimo sada dva kratka dokumenta s pripadnim grafovima događaja (slike 4.2a i 4.2b).

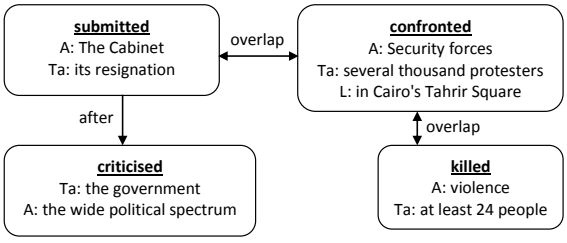
> ***Vijest 1***. *Prime Minister Sharaf and his Cabinet have <u>submitted</u> their resignations to the ruling military council on Monday after police <u>clashed</u> protesters in Cairo third day in a row. At least 24 people have <u>died</u> since the last Egypt's government was <u>toppled</u> nine months ago.*

> ***Vijest 2***. *The Cabinet has <u>submitted</u> its resignation to the ruling military council after the government has been consistently <u>criticized</u> by the wide political spectrum. Security forces <u>confronted</u> Monday several thousand protesters in Cairo's Tahrir Square in the third straight day of violence that has <u>killed</u> at least 24 people.*

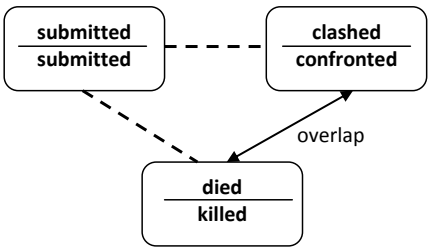
Budući da postoje tri para koreferentnih spominjanja događaja između dokumenata (*submitted*/*submitted*, *clashed*/*confronted* i *died*/*killed*), graf G_P, koji nastaje umnoškom grafova G i G', ima tri vrha, neovisno o tome radi li se o tenzorskom ili konormalnom umnošku grafova (slika 4.2c). Graf koji nastaje tenzorskim umnoškom grafova G i G' imat će samo brid *clashed*/*confronted* $\leftrightarrow$ *died*/*killed* (iscrtan punom crtom na slici 4.2c) jer postoji brid koji označava vremenski odnos OVERLAP između vrhova *clashed* i *died* u grafu G te između vrhova *confronted* i *killed* u grafu G'. Graf koji nastaje konormalnim umnoškom grafova G i G' dodatno ima još dva brida – brid *submitted*/*submitted* $\leftrightarrow$ *died*/*killed* (zbog brida *submitted* $\leftrightarrow$ *died* u grafu



(a) Graf događaja G (Vijest 1)



(b) Graf događaja G' (Vijest 2)



(c) Umnožak G_P grafova G i G'

Slika 4.2: Primjer umnoška grafova. Iscrtkani bridovi na slici 4.2c postoje isključivo u grafu koji je rezultat konormalnog umnoška. U grafu koji nastaje tenzorskim umnoškom postoji samo brid iscrtan punom crtom.

G) te brid *submitted/submitted* $\leftrightarrow$ *clashed/confronted* (zbog brida *submitted* $\rightarrow$ *clashed* u grafu G odnosno brida *submitted* $\leftrightarrow$ *confronted* u grafu G'). Iako odgovarajući bridovi postoje u oba ulazna grafa, brid *submitted/submitted* $\leftrightarrow$ *clashed/confronted* ne postoji u rezultatu tenzorskog umnoška jer oznake odgovarajućih bridova u ulaznim grafovima nisu podudarne – brid *submitted* $\rightarrow$ *clashed* u grafu G ima oznaku AFTER dok brid *submitted* $\leftrightarrow$ *confronted* u grafu G' ima oznaku OVERLAP.

Računanje jezgrene funkcije

Nakon što je umnožak grafova izgrađen, sličnost između ulaznih grafova mjeri se jezgrenom funkcijom na način kako slijedi. Neka je $\mathbf{A_P}$ matrica susjedstva grafa G_P koji je rezultat umnoška ulaznih grafova G i G'. Jezgrenu funkciju umnoška grafova $k_P(G,G')$ tada računamo prema sljedećem izrazu:

$$k_P(G,G') = \sum_{i,j=1}^{|V_P|} \left[\sum_{k=0}^{\infty} \lambda^k \mathbf{A_P}^k \right]_{ij} = \sum_{i,j=1}^{|V_P|} \left[(\mathbf{I} - \lambda \mathbf{A_P})^{-1} \right]_{ij} \tag{4.9}$$

gdje je λ početna vrijednost geometrijskog niza $\lambda_k = \lambda^k$. Član $\lambda^k A_P^k$ predstavlja doprinos šetnji duljine k, dok druga jednakost proizlazi iz zbroja geometrijskog reda matrica ($\sum_{k=0}^{\infty} A^k =$

$(\mathbf{I}-\mathbf{A})^{-1}$). Red predstavljen gornjim zbrojem konvergirat će (tj. moći ćemo izračunati vrijednost jezgrene funkcije) ukoliko vrijedi $\lambda < \frac{1}{t}$ gdje je t najveći stupanj vrha u umnošku G_P, tj. $t = \Delta_{max}(G_P)$ (Borgwardt, 2007). Vrijednost parametra λ uobičajeno se postavlja na $\lambda = \frac{1}{t+1}$.

Pokažimo sada izračun jezgrene funkcije prema jednadžbi (4.9) na primjeru tenzorskog umnoška G_P sa slike 4.2c. S obzirom da graf G_P ima samo jedan brid, vrijednost parametra λ iznosi $\frac{1}{2}$, a iznos jezgrene funkcije je:

$$\begin{aligned} k_P &= \sum_{i,j=1}^{3} \left[\left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \right)^{-1} \right]_{i,j} \\ &= \sum_{i,j=1}^{3} \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1.33 & 0.67 \\ 0 & 0.67 & 1.33 \end{pmatrix} \right] \\ &= 5 \end{aligned}$$

4.2.4 Jezgrena funkcija težinske dekompozicije grafa

Za razliku od jezgrene funkcije umnoška grafova koja računa broj zajedničkih šetnji između dva grafa, jezgrena funkcija težinske dekompozicije grafa (engl. *weighted decomposition kernel*) uspoređuje podgrafove proizvoljne veličine koji se nazivaju *selektorima* (engl. *selectors*), koristeći za usporedbu predikat koji definira uvjet podudarnosti između selektora (engl. *selector matching predicate*) (Menchetti i dr., 2005). Svaki par selektora, pri čemu po jedan selektor para dolazi iz svakog od ulaznih grafova, doprinosi ukupnom iznosu jezgrene funkcije sukladno sličnosti između konteksta tih dvaju selektora. Za svaki se selektor definira vektor konteksta koji su, kao i sami selektori, u općenitoj definiciji jezgrene funkcije težinske dekompozicije grafa podgrafovi proizvoljne veličine.

Neka je $S(G)$ skup svih uređenih parova $(s,\mathbf{z})$ gdje je s selektor (podgraf proizvoljne veličine), a $\mathbf{z}$ vektor konteksta selektora s. Općenit oblik jezgrene funkcije težinske dekompozicije grafa onda je:

$$k_{WD}(G,G') = \sum_{\substack{(s,\mathbf{z})\in S(G) \\ (s',\mathbf{z}')\in S(G')}} \delta(s,s') \sum_{d=1}^{D} \kappa_d(z_d, z'_d) \tag{4.10}$$

gdje je $\delta(s,s')$ indikatorska funkcija koja definira jesu li selektori s i s' podudarni, D dimenzija vektora konteksta $\mathbf{z}$ i $\mathbf{z}'$ (tj. broj konteksta pojedinog selektora), a κ_d jezgrena funkcija koja računa sličnost konteksta z_d i z'_d podudarnih selektora s i s'.

Da bi bilo moguće računati jezgrenu funkciju težinske dekompozicije grafova za konkretne grafove, potrebno je definirati što su selektori, a što njihovi konteksti. Za graf događaja $G = (V,E,A,m,r)$ selektorima smatramo pojedinačne vrhove grafa, tj. svaki vrh $v \in V_G$ jedan je selektor. Dva su razloga za ovakav odabir selektora:

1. Budući da vrhovi grafa događaja predstavljaju spominjanja događaja, koreferenciju spominjanja događaja možemo izravno koristiti kao predikat podudarnosti selektora. Drugim riječima, predikat podudarnosti selektora $\delta(v, v')$ smatramo zadovoljenim ako i samo ako su spominjanja događaja $m(v)$ i $m'(v')$ koreferentna;
2. Kada bismo kao selektore odabirali podgrafove s dva ili više vrhova tada bismo trebali definirati kompleksniji predikat podudarnosti selektora δ. Takav predikat bi, u smislu događaja i odnosa među njima, bio manje intuitivan od koreferencije pojedinačnih spominjanja događaja.

Za svaki selektor v u grafovima događaja definiramo samo jedan kontekst Z_v (tj. vektor konteksta $\mathbf{z}$ ima samo jednu komponentu Z_v) kao podgraf koji sadrži neposredno susjedstvo selektora v (dakle, vrh v i sve njegove susjedne vrhove). Iz perspektive događaja, kontekst čine svi događaji koji su u izravnom vremenskom odnosu s događajem predstavljenim selektorom. U skladu s navedenim, za grafove događaja izraz (4.10) se pojednostavljuje te se jezgrena funkcija težinske dekompozicije grafa računa na sljedeći način:

$$k_{WD}(G, G') = \sum_{v \in V_G, v' \in V_{G'}} coref(m(v), m'(v'))\ \kappa(Z_v, Z'_{v'}) \tag{4.11}$$

pri čemu je $coref(m(v), m'(v'))$ funkcija čija je vrijednost jednaka 1 ako i samo ako su spominjanja događaja $m(v)$ i $m'(v')$ koreferentna, a 0 inače. $\kappa(Z_v, Z'_{v'})$ je jezgrena funkcija koja mjeri sličnost konteksta Z_v i $Z'_{v'}$. Jezgrenu funkcija konteksta $\kappa(Z_v, Z'_{v'})$ računa broj parova koreferentnih spominjanja koji postoje između konteksta, pri čemu je taj broj normaliziran brojem vrhova većeg od dvaju konteksta:

$$\kappa(Z_v, Z'_{v'}) = \frac{\sum_{w \in V_{Z_v}, w' \in V_{Z'_{v'}}} coref(m(w), m'(w'))}{\max(|V_{Z_v}|, |V_{Z'_{v'}}|)}. \tag{4.12}$$

Ovakva definicija jezgrene funkcije konteksta motivirana je intuicijom da doprinos para koreferentnih spominjanja događaja $m(v)$ i $m'(v')$ ukupnoj sličnosti između grafova događaja treba biti razmjeran broju parova koreferentnih spominjanja $(m(w), m'(w'))$, gdje je $m(w)$ u izravnom vremenskom odnosu s $m(v)$, a $m'(w')$ u izravnom vremenskom odnosu s $m'(v')$. Drugim riječima, par koreferentnih spominjanja događaja je to važniji što je više parova podudarnih spominjanja događaja između skupova događaja s kojima su u izravnim vremenskim odnosima.

Pokažimo sada izračun jezgrene funkcije težinske dekompozicije grafa na primjeru grafova G i G' prikazanih na slikama 4.2a i 4.2b. Kao što smo već vidjeli kod izračuna jezgrenih funkcija umnoška grafova, parovi koreferentnih spominjanja događaja u tekstovima koji odgovaraju grafovima G i G' su: $submitted_G/submitted_{G'}$, $clashed_G/confronted_{G'}$ i $died_G/killed_{G'}$. Odre-

dimo prvo kontekste svih spominjanja događaja ovih koreferentnih parova:

$$
\begin{aligned}
Z_{submitted_G} &= \{submitted_G, toppled_G, clashed_G, died_G\} \\
Z_{clashed_G} &= \{clashed_G, submitted_G, died_G\} \\
Z_{died_G} &= \{died_G, submitted_G, toppled_G, clashed_G\} \\
Z_{submitted_{G'}} &= \{submitted_{G'}, confronted_{G'}, criticized_{G'}\} \\
Z_{confronted_{G'}} &= \{confronted_{G'}, submitted_{G'}, killed_{G'}\} \\
Z_{killed_{G'}} &= \{killed_{G'}, confronted_{G'}\}
\end{aligned}
$$

Sada možemo izračunati iznose jezgrenih funkcija konteksta (tj. težine) za sve parove koreferentnih spominjanja događaja:

$$
\begin{aligned}
\kappa(Z_{submitted_G}, Z_{submitted_{G'}}) &= \frac{|\{submitted_G/submitted_{G'}, clashed_G/confronted_{G'}\}|}{\max\left(|Z_{submitted_G}|, |Z_{submitted_{G'}}|\right)} \\
&= \frac{2}{4} \\
\kappa(Z_{clashed_G}, Z_{confronted_{G'}}) &= \frac{|\{clashed_G/confronted_{G'}, submitted_G/submitted_{G'}, died_G/killed_{G'}\}|}{\max\left(|Z_{clashed_G}|, |Z_{confronted_{G'}}|\right)} \\
&= \frac{3}{3} \\
\kappa(Z_{died_G}, Z_{killed_{G'}}) &= \frac{|\{died_G/killed_{G'}, clashed_G/confronted_{G'}\}|}{\max\left(|Z_{died_G}|, |Z_{killed_{G'}}|\right)} \\
&= \frac{2}{4}
\end{aligned}
$$

Konačno, računamo iznos jezgrene funkcije težinske dekompozicije grafova:

$$
\begin{aligned}
k_{WD}(G, G') &= \kappa(Z_{submitted_G}, Z_{submitted_{G'}}) + \kappa(Z_{clashed_G}, Z_{confronted_{G'}}) + \kappa(Z_{died_G}, Z_{killed_{G'}}) \\
&= \frac{2}{4} + \frac{3}{3} + \frac{2}{4} \\
&= 2.
\end{aligned}
$$

Poglavlje 5

Tekstna zbirka označena događajima

Za učenje modela temeljenih na nadziranom strojnom učenju kao i za razvoj modela temeljenih na pravilima potrebno je raspolagati tekstovima u kojima su ljudi označili (engl. *gold annotations*) one izraze koje model treba moći automatski prepoznati (npr. sidra događaja ili argumente događaja). Ručno označeni tekstovi potrebni su i za vrednovanje modela koji automatski crpe informacije iz teksta. U ovom poglavlju opisana je izrada velike tekstne zbirke u kojoj su ljudi označili sidra činjeničnih spominjanja događaja, a potom na podskupu dokumenata te zbirke označili i cjelovite grafove događaja – argumente događaja te vremenske odnose i koreferenciju između spominjanja događaja.

5.1 Označavanje sidara spominjanja događaja

Ova se disertacija bavi crpljenjem činjeničnih spominjanja događaja. Kako zbirka TimeBank, kao najveća zbirka ručno označena spominjanjima događaja, sadrži i nečinjenične (npr. negirane, hipotetske) događaje te stanja (v. odjeljak 3.2.1), bilo je potrebno izgraditi zbirku označenu isključivo činjeničnim događajima. To je moguće ostvariti na dva načina:

1. Prilagodbom zbirke TimeBank na način da se uklone sva označena spominjanja koja se odnose na stanja i nečinjenične događaje;
2. Izgradnjom vlastite tekstne zbirke označene isključivo činjeničnim spominjanjima događaja.

Uvažavajući činjenicu da se kritike na račun zbirke TimeBank u dobroj mjeri odnose upravo na njenu ograničenu veličinu (Boguraev i Ando, 2005), za potrebe istraživanja u okviru ove doktorske disertacije izgrađena je vlastita tekstna zbirka označena isključivo činjeničnim spominjanjima događaja. Tekstnu zbirku, nazvanu EVEXTRA, čini 759 novinskih članaka odnosno ukupno 330.000 pojavnica, automatski prikupljenih putem internetske usluge EMM NewsBrief, koja prikuplja i grupira novinske objave s nekoliko stotina različitih novinskih portala.[1] U ta-

[1] `http://emm.newsbrief.eu/NewsBrief/clusteredition/en/latest.html`

Tablica 5.1: Usporedba veličine tekstnih zbirki TimeBank i EVEXTRA

Razred događaja	#TimeBank	#EVEXTRA
OCCURRENCE	4452	12895
REPORTING	1010	5263
I-ACTION	668	4610
PERCEPTION	51	250
STATECHANGE	–	961
STATE	1181	–
I-STATE	586	–
ASPECTUAL	295	–
Ukupno	8243	23979

blici 5.1 dana je usporedba zbirki TimeBank i EVEXTRA prema broju označenih spominjanja događaja koja zorno pokazuje da je zbirka EVEXTRA otprilike tri puta veća od zbirke TimeBank.[2].

5.1.1 Postupak označavanja

Sidra činjeničnih spominjanja događaja u tekstovima zbirke EVEXTRA označavalo je ukupno šest označivača prema pomno sastavljenim uputama za označavanje (v. dodatak B). Označavanje se sastojalo od dva zadatka: (1) označavanja riječi koje su sidra činjeničnih spominjanja događaja i (2) određivanja semantičkih razreda kojima činjenična spominjanja događaja pripadaju (npr. REPORTING; v. odjeljak 4.1.2).

Označavanje je bilo podijeljeno u pet faza. U prvoj je fazi (fazi usklađivanja označivača) svih šest označivača označavalo isti skup dokumenata veličine 10.000 pojavnica (skup za usklađivanje), gdje su označivači uvježbavali označavanje u skladu s uputama za označavanje. Analizom neslaganja označivača na skupu za usklađivanje prepoznata su sustavna odstupanja pojedinih označivača te su razrješavanjem neslaganja konsenzusom označivači dodatno uskladili način označavanja. Nakon usklađivanja, u prvom krugu označavanja označivači su podijeljeni u tri para, pri čemu su označivači svakog para nezavisno označavali skup dokumenata veličine 50.000 pojavnica (oba označivača para nezavisno su jedan od drugoga označavali identične dokumente). Neslaganja označivača za svaki od parova razrješavao je treći označivač. Da

[2]Zbirka EVEXTRA dostupna je na adresi `http://takelab.fer.hr/grapheve` pod licencijom CC BY-SA-NC 3.0

Tablica 5.2: Faze označavanja sidara činjeničnih događaja u zbirci EVEXTRA

Faza	Pojavnica po osobi	Pojavnica ukupno	Trajanje
Usklađivanje	10.000	10.000	3 tjedna
Po parovima #1	50.000	150.000	4 tjedna
Zajednički #1	10.000	10.000	2 tjedna
Po parovima #2	50.000	150.000	4 tjedna
Zajednički #2	10.000	10.000	1 tjedan
Ukupno	130.000	330.000	14 tjedana

bismo se uvjerili da je način označavanja označivača i dalje usklađen, nakon prve runde označavanja svi su označivači označili isti skup dokumenata (prvi zajednički skup) veličine 10.000 pojavnica. Potom je uslijedila druga runda označavanja u kojoj su označivači ponovno podijeljeni u parove (s time da su označivači upareni drugačije nego u prvom krugu označavanja) i gdje je svaki od označivača ponovno označio 50.000 pojavnica. Neslaganja svakog od parova ponovno je razrješavao jedan od preostalih označivača. Konačno, kako bismo utvrdili slaganje svih označivača na kraju označavanja, svi su označivači još jednom označili isti skup dokumenata (drugi zajednički skup). Postupak označavanja sa svim svojim fazama sažeto opisuje tablica 5.2.

5.1.2 Slaganje označivača

Kako bi se koristili za izgradnju ili vrednovanje automatiziranih modela, ručno označeni podaci moraju biti pouzdani. U područjima računalne lingvistike i obrade prirodnog jezika smatra se da su podaci pouzdani ukoliko je slaganje u oznakama između označivača dovoljno izraženo (Artstein i Poesio, 2008). Budući da su dokumente iz skupa za usklađivanje te zajedničkih skupova označavali svi označivači, slaganje među označivačima mjereno je na uniji tih triju skupova, tj. na skupu veličine 30.000 pojavnica. Slaganje u oznakama između dva označivača mjerimo mjerom F_1 (harmonijskom sredinom *preciznosti* i *odziva*), standardnom mjerom za vrednovanje klasifikatora u strojnom učenju (van Rijsbergen, 1979). Neka je *TP* broj pojavnica koje su oba označivača označili kao sidra događaja. Neka je *FP* broj pojavnica koje je samo prvi označivač označio kao sidro događaja (a drugi označivač nije), a *FN* broj pojavnica koje je samo drugi označivač označio kao sidra događaja (a prvi nije). Tada preciznost P, odziv R i

Tablica 5.3: Slaganje označivača pri označavanju sidara činjeničnih spominjanja događaja

	Semantički razred događaja				
Sidro	OCCURRENCE	I-ACTION	REPORTING	PERCEPTION	STATECHANGE
87,4	65,7	59,0	86,0	45,1	42,5

mjeru F_1 računamo na sljedeći način:

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F_1 = 2 \cdot \frac{P \cdot R}{P+R}. \tag{5.1}$$

Slaganje označivača na zadatcima prepoznavanja sidara činjeničnih spominjanja događaja te njihova svrstavanja u semantičke razrede prikazano je u tablici 5.3. Prikazana su prosječna slaganja dvaju označivača, dobivena uprosječivanjem slaganja za svih 15 parova označivača (broj kombinacija odabira dva između šest označivača). Slaganje označivača na zadatku određivanja koje su riječi sidra događaja vrlo je visoko. Razmjer neslaganja u određivanju semantičkog razreda događaja je, međutim, značajan, čak i uz dva kruga usklađivanja, što navodi na zaključak da je određivanje semantičkih razreda vrlo zahtjevan zadatak čak i za ljude. Neslaganja su posebno izražena za razrede STATECHANGE i PERCEPTION jer su označivači često griješili i spominjanja koja spadaju u te razrede svrstavali u općenitiji i znatno brojniji razred OCCURRENCE. Stvarna kvaliteta oznaka u zbirci EVEXTRA je, međutim, bitno bolja od kvalitete koju impliciraju slaganja dana u tablici 5.3 budući da je nakon mjerenja slaganja označivača uslijedilo razrješavanje neslaganja (treći označivač razrješavao je neslaganja između dvojice izvornih označivača), čime je ispravljeno mnogo pogrešnih oznaka izvornih označivača.

Slaganje označivača u označavanju činjeničnih sidara događaja na zbirci EVEXTRA (87% F_1) puno je bolje od slaganja označivača na zbirci TimeBank (78% prosjek preciznosti i odziva),[3] što sugerira bolju kvalitetu oznaka zbirke EVEXTRA u odnosu na zbirku TimeBank. Ova razlika još je vrjednija uzme li se u obzir: (1) da je označavanje činjeničnih spominjanja događaja zahtjevnije od označavanja provedenog kod izgradnje zbirke TimeBank jer su označivači morali dodatno razlikovati stanja od događaja te činjenične od nečinjeničnih spominjanja događaja; (2) da je procjena slaganja označivača kod zbirke EVEXTRA puno pouzdanija jer se radi o prosječnim slaganjima 15 parova označivača, dok je na zbirci TimeBank slaganje izmjereno samo za jedan par označivača; (3) da je slaganje na zbirci EVEXTRA izraženo mjerom F_1 (harmonijskom sredinom preciznosti i odziva), koja daje konzervativniju procjenu slaganja od prosjeka (tj. aritmetičke sredine) preciznosti i odziva, kojim je mjereno slaganje na zbirci TimeBank.

[3] `http://timeml.org/site/timebank/documentation-1.2.html#iaa`

Tablica 5.4: Statistika označavanja argumenata događaja

	Skup	AGENT	TARGET	TIME	LOCATION
Broj	Skup za razvoj	978	706	218	205
	Skup za vrednovanje	2063	1589	511	551
	Ukupno	3041	2295	729	756
Prosječna duljina izraza (broj pojavnica)		2,78	3,03	4,28	2,41

5.2 Označavanje argumenata

Za potrebe razvoja postupka za crpljenje argumenata događaja kao i za vrednovanje točnosti tog postupka bilo je potrebno ručno označiti argumente događaja na podskupu dokumenata zbirke EVEXTRA. Za označavanje argumenata događaja slučajno je odabrano 105 dokumenata zbirke EVEXTRA u kojima su argumente događaja označavala dva označivača. Označivači su za prethodno označena sidra događaja trebali odrediti izraze u tekstu koji predstavljaju argumente tih događaja, koji pak imaju jednu od četiri semantičke uloge (v. poglavlje 4.1.2): AGENT, TARGET, TIME i LOCATION. Po uzoru na upute za označavanje spominjanja entiteta za potrebe razrješavanja koreferencije entiteta (Pradhan i dr., 2011), označivačima je dan naputak da za svaki argument označe izraz s najvećim mogućim prostiranjem (engl. *extent*). Na primjer, u rečenici "*The vessel Low Speed Chase competed on Saturday in a race around South Farallon Island*", za sidro "*competed*" kao argument sa semantičkom ulogom AGENT potrebno je označiti izraz "*The vessel Low Speed Chase*", a ne izraze s manjim prostiranjem poput "*Chase*" ili "*Low Speed Chase*".

Kao i kod označavanja sidara činjeničnih događaja, i kod označavanja argumenata događaja, prvo je odabrano deset dokumenata na kojima su se označivači usuglasili sukladno uputama za označavanje. Nakon što su uklonjena sustavna neslaganja zbog pogrešnog ili nepotpunog razumijevanja uputa za označavanje, oba su označivača međusobno nezavisno označila još 15 dokumenata koji zajedno s 10 dokumenata skupa za usklađivanje čine skup za razvoj modela za crpljenje argumenata događaja. Nakon toga, svaki je od označivača označio zaseban skup od 40 dokumenata (tj. ukupno 80 dokumenata), čime je dobiven skup za vrednovanje postupka crpljenja argumenata događaja. Tablica 5.4 sadrži podatke o broju i prosječnoj duljini argumenata (po pojedinim semantičkim ulogama argumenata) u označenom podskupu od 105 dokumenata zbirke EVEXTRA.

U tablici 5.5 prikazano je slaganje između označivača izmjereno mjerom F_1 na 25 dokumenata skupa za razvoj, nezavisno označenih od strane oba označivača. Podudaranje argumenata podrazumijeva podudarnost semantičkih uloga, a mjereno je na dva načina – blago i strogo.

Tablica 5.5: Slaganje označivača pri označavanju argumenata događaja

	AGENT	TARGET	TIME	LOCATION	Prosjek (po primjerima)
Blago	89,2	79,4	76,0	75,8	86,2
Strogo	77,6	57,3	55,7	57,0	68,1

Kod blagog mjerenja podudarnosti (prvi redak tablice 5.5) označeni se argumenti događaja smatraju podudarnima ukoliko sadrže barem jednu zajedničku pojavnicu. Kod strogog podudaranja (drugi redak tablice 5.5) označeni su argumenti podudarni isključivo ako su identični, tj. ako se podudaraju u svim pojavnicama. Ostvareno slaganje označivača od preko 86% uz blago podudaranje vrlo je visoko, što znači da je skup dokumenata koji se koristi za razvoj automatskog pristupa za crpljenje argumenata događaja prilično konzistentno označen. Broj argumenata koji se potpuno podudaraju u prostiranju (strogo podudaranje), međutim, značajno je manji. Pregled primjera u kojima postoji blago, ali ne i strogo podudaranje između označenih argumenata otkriva da se u velikom broju slučajeva radi o informacijski nebitnim neslaganjima poput izostavljanja članova (npr. "*The vessel Low Speed Chase*" nasuprot "*vessel Low Speed Chase*").

5.3 Označavanje cjelovitih grafova događaja

Kako bi bilo moguće cjelovito vrednovati automatski izgrađene grafove događaja (v. poglavlje 7), bilo je potrebno pripremiti skup ručno označenih (tj. referentnih) grafova događaja. U cilju minimizacije troška i vremena označavanja, za ručno su označavanje grafova događaja angažirani isti oni označivači koji su prethodno označavali argumente događaja, a ručna izgradnja grafova provedena je na 105 dokumenata s već označenim sidrima i argumentima događaja. Na taj je način izgradnja cjelokupnih grafova događaja u posljednjem koraku svedena na označavanje vremenskih odnosa i odnosa koreferencije između spominjanja događaja.[4]

Dva označivača prvo su nezavisno označila istih 25 dokumenata, a potom je svaki od njih označio zaseban skup od 40 dokumenata. Slaganje između označivača izmjereno je na 25 dokumenata označenih od strane oba označivača, mjerama ROS i LCC (v. poglavlje 7) – istim mjerama koje su u ovoj disertaciji predložene za mjerenje ukupne kvalitete automatski izgrađenih grafova događaja. Izmjereno slaganje na 25 dokumenata iznosi $0,47 \pm 0,13$ LCC i $0,69 \pm 0,08$ ROS. Statistički podaci poput prosječnog broja vrhova i bridova ručno označenih grafova događaja dani su u tablici 5.6 (prosjek i devijacija za 105 ručno označenih grafova događaja).

[4]Skup od 105 ručno izgrađenih grafova događaja dostupan je na adresi `http://takelab.fer.hr/grapheve` pod licencijom CC BY-SA-NC 3.0.

Tablica 5.6: Statistički podaci ručno označenih grafova događaja

Svojstvo	Iznos
Broj vrhova	41,7 ± 21,7
Broj bridova	62,4 ± 36,2
Prosječan stupanj vrha	3,0 ± 0,9
Broj vrhova najveće povezane komponente	39,8 ± 21,0

Poglavlje 6

Izgradnja grafova događaja

U ovom poglavlju detaljno opisujemo postupak izgradnje grafova događaja iz teksta. Postupak izgradnje grafova događaja usklađen je s pretpostavkama uvedenima u odjeljku 4.1.2, što znači da povezuje modele za crpljenje činjeničnih spominjanja događaja (crpljenje sidara i argumenata događaja) i modele za prepoznavanje odnosa među spominjanjima događaja (vremenski odnosi i koreferencija). Cjelokupni postupak izgradnje grafova događaja iz novinskih članaka opisan je u znanstvenom radu (Glavaš i Šnajder, 2014a).

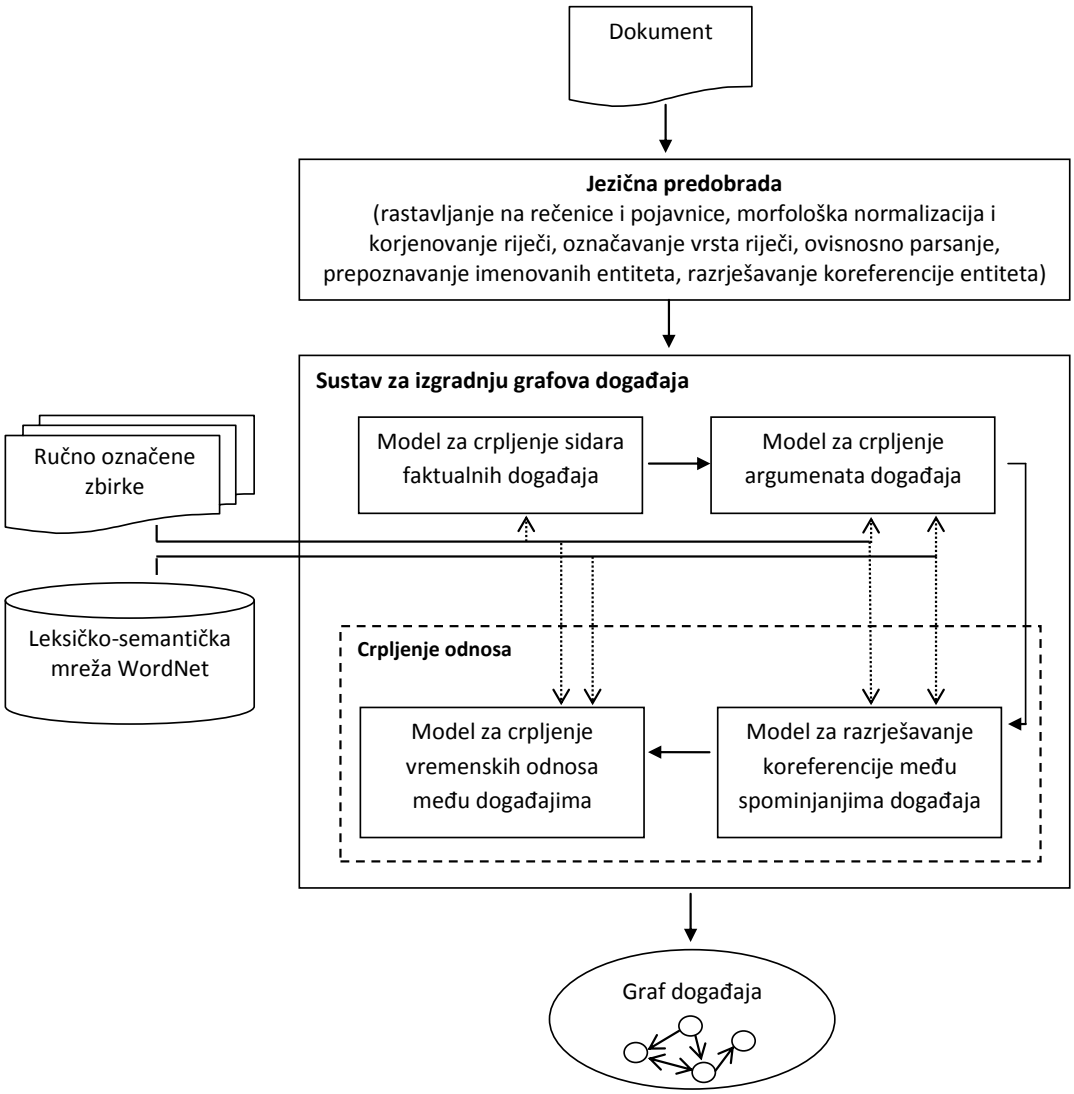
Postupak izgradnje grafova događaja povezuje sljedeće modele:

1. Model nadziranog strojnog učenja za crpljenje sidara činjeničnih spominjanja događaja;
2. Model za crpljenje argumenata događaja temeljen na pravilima;
3. Model nadziranog strojnog učenja za crpljenje vremenskih odnosa među događajima;
4. Model nadziranog strojnog učenja za određivanje koreferencije među spominjanjima događaja.

Arhitektura sustava za izgradnju grafova događaja prikazana je slikom 6.1. Postupak izgradnje grafa događaja iz teksta oslanja se na predobradbene alate (engl. *preprocessing tools*). Sustav za izgradnju grafova događaja predstavljen u ovom poglavlju oslanja se na alate iz programskog paketa Stanford CoreNLP,[1] koji služe za obradu teksta pisanoga engleskim jezikom. Konkretno, modeli uključeni u sustav za izgradnju grafova događaja koriste sljedeće postupke obrade prirodnog jezika:

- Rastavljanje teksta na rečenice (engl. *sentence splitting*);
- Rastavljanje rečenica na pojavnice (engl. *tokenization*);
- Lematizaciju (engl. *lemmatization*) i korjenovanje riječi (engl. *stemming*);
- Označavanje vrsta riječi (engl. *part-of-speech tagging, POS tagging*) (Toutanova i Manning, 2000);
- Ovisnosno parsanje rečenica (engl. *dependency parsing*) (De Marneffe i dr., 2006);
- Prepoznavanje imenovanih entiteta (engl. *named entity recognition, NER*) (Finkel i dr.,

[1] `http://nlp.stanford.edu/software/corenlp.shtml`



Slika 6.1: Arhitektura sustava za izgradnju grafova događaja iz teksta

2005);

- Razrješavanje koreferencije entiteta (engl. *entity coreference resolution*) (Raghunathan i dr., 2010; Lee i dr., 2011).

Za učenje modela koji čine sustav za izgradnju grafova događaja potrebne su ručno označene zbirke tekstova (v. poglavlje 5). Modeli koji čine sustav za izgradnju grafova događaja svoj rad temelje i na semantičkim odnosima među riječima koji su opisani u leksičko-semantičkoj ontologiji WordNet (Fellbaum, 2010). U WordNetu su koncepti, definirani kao skupovi sinonima (engl. *synonym set, synset*), povezani u hijerarhiju prvenstveno na temelju odnosa hiperonim-hiponim (engl. *is-a relation*) – odnosa koji označava da je jedan koncept vrsta drugog koncepta (npr. koncept *mačka* hiponim je koncepta *životinja*).

U nastavku je detaljno opisan svaki od pojedinačnih modela koji čine sustav za izgradnju grafova događaja te eksperimenti kojima je mjerena uspješnost svakog od tih modela zasebno (tj. neovisno o rezultatu ostalih modela).

6.1 Crpljenje sidara činjeničnih događaja

Prepoznavanje pojavnica u tekstu koje predstavljaju sidra činjeničnih spominjanja događaja prvi je korak izgradnje grafova događaja. Crpljenje sidara događaja obavlja se u dva koraka. U prvome koraku za svaku se pojavnicu u tekstu odlučuje je li sidro događaja. Sidro događaja uvijek je točno jedna pojavnica, a dvije susjedne pojavnice prepoznate kao sidra događaja smatramo sidrima različitih spominjanja događaja, tj. susjedna sidra događaja se ne spajaju (npr. "*accident*" i "*happened*" u rečenici "*Accident happened early in the afternoon*" smatramo dvama sidrima događaja). U drugome koraku svako je sidro događaja potrebno svrstati u jedan od pet semantičkih razreda (v. poglavlje 4.1.2): OCCURRENCE, I-ACTION, PERCEPTION, REPORTING i STATECHANGE.

6.1.1 Modeli i značajke

U prvome koraku radi se o binarnom klasifikacijskom problemu (pojavnica ili jest ili nije sidro događaja), dok se o drugom koraku radi o višeklasnom klasifikacijskom problemu gdje je sidru događaja potrebno pridijeliti semantički razred događaja. Oba problema rješavamo modelima nadziranog strojnog učenja, pri čemu u oba slučaja koristimo isti skup jezično motiviranih značajki:

- Riječ (f_1^S), lema (f_2^S), korijen (f_3^S) i završetak (f_4^S) pojavnice. Lema je morfološki normaliziran oblik riječi (npr. *dog* je lema riječi *dogs*), dok je završetak pojavnice sufiks riječi od zadnjeg samoglasnika ili od predzadnjega ukoliko je zadnje slovo riječi samoglasnik (npr. *-ing* je završetak riječi *swimming*);
- Detaljna vrsta riječi pojavnice (f_5^S) i gruba vrsta riječi pojavnice (f_6^S). Kao detaljne vrste riječi koristimo one koje su označene u banci stabala Penn (Marcus i dr., 1993) (npr. NN, VBD, JJS, PRP), dok kao grube vrste riječi koristimo *imenice*, *glagole*, *pridjeve* i *ostalo*;
- Kontekstne značajke za trenutnu pojavnicu su značajke $f_1^S - f_6^S$ izračunate za dvije pojavnice koje prethode promatranoj pojavnici ($f_7^S - f_{18}^S$) i dvije pojavnice koje slijede nakon promatrane pojavnice ($f_{19}^S - f_{30}^S$). Ako, na primjer, računamo značajke za pojavnicu "*did*" u rečenici "*Stallion did not finish the race*", tada će među kontekstnim značajkama, primjerice, biti završetak pojavnice "*Stallion*" (*-on*), koja prethodi pojavnici "*did*", ili pak gruba vrsta riječi pojavnice "*finish*" (*glagol*), koja slijedi nakon pojavnice "*did*";
- Skup svih ovisnosnih sintaktičkih relacija promatrane pojavnice (f_{31}^S). Ovisnosne relacije koje uzimamo u obzir su one koje se mogu dobiti ovisnosnim parsanjem koristeći Stanfordov ovisnosni parser (De Marneffe i Manning, 2008). Primjerice, ovisnosno stablo rečenice "*Stallion finished the race*" sadrži sljedeće ovisnosne relacije: *nsubj*("*finished*", "*Stallion*"), *det*("*race*", "*the*") i *dobj*("*finished*", "*race*"). Vrijednost značajke f_{31}^S za pojavnicu "*finished*" stoga je skup {*nsubj*, *dobj*};

- Vrsta sintaktičkog odsječka (engl. *syntactic chunk*) kojem pojavnica pripada (f^S_{32}) te vrsta prvog sintaktičkog odsječka koji prethodi odsječku pojavnice (f^S_{33}) i vrsta prvog sintaktičkog odsječka koji slijedi nakon odsječka pojavnice (f^S_{34}). Rečenica "*Low Speed Chase had won the race in Auckland*", primjerice, sastoji se od četiri sintaktička odsječka: imeničkog odsječka (engl. *noun phrase*, NP) "*Low Speed Chase*", glagolskog odsječka (engl. *verbal phrase*, VP) "*had won*", imeničkog odsječka "*the race*" i prijedložnog odsječka (engl. *prepositional phrase*, PP) "*in Auckland*". Vrijednosti značajki za, primjerice, pojavnicu "*race*" stoga su: $f^S_{32} = \text{NP}$, $f^S_{33} = \text{VP}$, $f^S_{34} = \text{PP}$;
- Binarna značajka koja označava je li pojavnica predikat koji ima izravan objekt, tj. postoji li ovisnosna relacija *dobj* kojom promatrana pojavnica upravlja (f^S_{35});
- Binarna značajka koja označava je li pojavnica nominalni subjekt nekog predikata, tj. postoji li ovisnosna relacija *nsubj* u kojoj predikat upravlja promatranom pojavnicom (f^S_{36});
- Binarna značajka koja označava je li pojavnica izravni objekt nekog predikata, tj. postoji li ovisnosna relacija *dobj* u kojoj predikat upravlja promatranom pojavnicom (f^S_{37});
- Modalni glagol koji se odnosi na pojavnicu (f^S_{38}). Ova značajka može poprimiti vrijednost nekog modalnog glagola (*can*, *could*, *will*, *would*, *may*, *might*, *shall*, *should*, *must*) ili posebnu vrijednost *none* ukoliko pojavnica nije modificirana modalnim glagolom. Ova je značajka vrlo korisna za razlikovanje činjeničnih od nečinjeničnih spominjanja događaja;
- Pomoćni glagol koji se odnosi na pojavnicu (npr. *is*, *are*, *has*, *have*, *had*), ukoliko postoji (f^S_{39});
- Binarna značajka koja označava je li pojavnica negirana (poput "*win*" u rečenici "*Stallion did not win the race*") (f^S_{40});
- Član (npr. *a*, *an*, *the*, *each*, *every*) koji se nalazi uz pojavnicu, ukoliko postoji (f^S_{41});
- Binarna značajka koja označava počinje li pojavnica velikim slovom f^S_{42}. Ova značajka doprinosi prepoznavanju spominjanja događaja koja su vlastita imena (npr. "*World Cup*", "*Olympics*");
- Binarna značajka koja označava je li pojavnica na skupu za učenje češće sidro događaja ili to češće nije (npr. riječ "*basketball*" gotovo nikad nije sidro događaja na skupu za učenje, dok je riječ "*killed*" gotovo uvijek sidro događaja na skupu za učenje) (f^S_{43});
- Binarna značajka koja označava je li lema pojavnice na skupu za učenje češće sidro događaja ili to češće nije (f^S_{44});
- Najčešći semantički razred događaja koji je istoj pojavnici pridijeljen na skupu za učenje (npr. pojavnica "*said*" najčešće je sidro semantičkog razreda REPORTING) (f^S_{45});
- Najčešći semantički razred događaja koji je pridijeljen lemi pojavnice na skupu za učenje (npr. lema "*say*" najčešće je sidro semantičkog razreda REPORTING) (f^S_{46}).

Iako je gore navedeno ukupno 46 različitih značajki, mnoge od njih su kategoričke značajke odnosno značajke koje mogu poprimiti konačan skup diskretnih vrijednosti. Neki od najuspješnijih i najčešće korištenih klasifikacijskih modela strojnog učenja (npr. logistička regresija, SVM, višeslojni perceptron), međutim, ne mogu izravno raditi s kategoričkim značajkama već je te značajke potrebno pretvoriti u niz binarnih značajki. Kategoričku značajku f_{MN} koja može poprimiti jednu od N diskretnih vrijednosti (v_1, v_2, ..., v_N) pretvaramo u N binarnih značajki (f_{B_1}, f_{B_2}, ..., f_{B_N}) gdje značajka f_{B_i} poprima vrijednost logičke istine ako i samo ako početna kategorička značajka f_{MN} poprimi diskretnu vrijednost v_i. Kako neke od gore navedenih kategoričkih značajki (npr. f_1^S, f_2^S i f_3^S) imaju i po nekoliko tisuća diskretnih vrijednosti koje mogu poprimiti, konačni prostor značajki s kojim se raspolaže ima kardinalitet veći od 100.000. S obzirom na tako veliku dimenzionalnost ulaznog prostora, pretpostavlja se da su primjeri za učenje približno linearno odvojivi, pa kao model strojnog učenja koristimo linearni diskriminativni klasifikacijski model, točnije, L2-regulariziranu logističku regresiju. U eksperimentima je korištena implementacija regularizirane logističke regresije iz programskog paketa LibLinear (Fan i dr., 2008).

6.1.2 Vrednovanje modela na zbirci EVEXTRA

Oba klasifikacijska modela – binarni koji odlučuje je li pojavnica sidro događaja te višeklasni koji sidrima dodjeljuje semantički razred – vrednovani su na isti način. Označeni skup dokumenata zbirke EVEXTRA podijeljen je na skup za učenje i skup za ispitivanje u omjeru 70%-30%. Hiperparametri modela logističke regresije optimiraju se deseterostrukom unakrsnom provjerom (engl. *10-fold cross validation*) na skupu za učenje uz korištenje pretraživanja po rešetki (engl. *grid search*) za pronalaženje optimalnih vrijednosti hiperparametara. U okviru ove disertacije, isti postupak optimizacije hiperparametara koristimo za sve klasifikacijske modele nadziranog strojnog učenja.

Modele za crpljenje sidara činjeničnih događaja optimirane unakrsnom provjerom na skupu za učenje vrednujemo na skupu za ispitivanje. U cilju bolje ocjene uspješnosti promatranih klasifikacijskih modela, njihovi rezultati na ispitnome skupu uspoređuju se s rezultatima jednostavne temeljne metode (engl. *baseline method*). Za probleme identifikacije sidara događaja i njihove klasifikacije u semantičke razrede predložena je temeljna metoda MEMORIZE (Bethard, 2008), koja za svaku pojavnicu predviđa onaj razred s kojim se ta pojavnica najčešće pojavljuje na skupu za učenje. Ako se, primjerice, riječ "*played*" na skupu za učenje najčešće pojavljuje kao sidro događaja, tada će model MEMORIZE svako pojavljivanje te riječi na skupu za ispitivanje okarakterizirati kao sidro događaja. Nadalje, riječima koje su na skupu za učenje najčešće sidra događaja model MEMORIZE dodijelit će onaj semantički razred s kojim se ta riječ najčešće pojavljuje u skupu za učenje (npr. razred REPORTING za riječ "*said*").

Rezultati vrednovanja modela za crpljenje sidara činjeničnih događaja (GRAF-SIDRA) iz

Tablica 6.1: Vrednovanje modela za crpljenje sidara činjeničnih događaja i njihovu klasifikaciju u semantičke razrede na zbirci EVEXTRA

Identifikacija sidara	P	R	F_1			
Temeljna metoda (MEMORIZE)	77,6	67,2	72,0			
GRAF-SIDRA	83,4	76,8	**79,9**			
Klasifikacija u semantičke razrede[a]	OCC	IACT	REP	PER	SCH	Prosjek F_1
Temeljna metoda (MEMORIZE)	83,5	47,4	89,5	52,3	38,9	62,3
GRAF-SIDRA	84,0	62,4	90,6	83,5	55,1	**75,2**

[a] OCC – OCCURRENCE; IACT – I-ACTION; PER – PERCEPTION; REP – REPORTING; SCH – STATECHANGE; Posljednji stupac predstavlja prosjek rezultata F_1 po razredima (engl. *macro-average* F_1)

teksta zajedno s rezultatima vrednovanja jednostavne temeljne metode MEMORIZE prikazani su u tablici 6.1. Model GRAF-SIDRA daje značajno bolje rezultate od temeljne metode MEMORIZE na oba klasifikacijska problema, pri čemu je važno uočiti da je s rezultatom od preko 70% F_1 na zadatku identifikacije sidara temeljna metoda MEMORIZE vrlo kompetitivna. Uspješnost modela GRAF-SIDRA za identifikaciju sidara činjeničnih događaja ($\approx$ 80% F_1; gornji dio tablice 6.1) sumjerljiva je slaganju označivača na istome zadatku ($\approx$ 87% F_1), što znači da uz trenutnu kakvoću referentnih oznaka ne preostaje puno prostora za unaprjeđenje modela GRAF-SIDRA. Na zadatku klasifikacije sidara u semantičke razrede događaja (donji dio tablice 6.1) razlika između uspješnosti modela GRAF-SIDRA i temeljne metode MEMORIZE još je veća, pri čemu se povećanje uspješnosti najviše očituje za razrede s najmanje primjera za učenje – PERCEPTION i STATECHANGE. Gledano po apsolutnoj uspješnosti, najbolji rezultat modeli ostvaruju za razred REPORTING, što je donekle i očekivano budući da je kod sidara događaja razreda REPORTING leksička varijacija najmanje izražena (većina događaja razreda REPORTING ostvarena je skupom od svega nekoliko glagola poput "*say*", "*tell*", "*report*", "*add*"). Važno je uočiti kako je za sve razrede uspješnost klasifikacije modelom GRAF-SIDRA veća od izmjerenog slaganja označivača (v. poglavlje 5.1), što potvrđuje pretpostavku da je razrješavanje neslaganja prilikom označavanja zbirke EVEXTRA značajno unaprijedilo kakvoću referentnih oznaka. Ovo je posebno uočljivo za malobrojne razrede (npr. PERCEPTION) za koje ispravljanje svega nekoliko netočno označenih primjera dovodi do značajnog poboljšanja konzistentnosti oznaka te posljedično do značajnog poboljšanja uspješnosti klasifikacije mjerene mjerom makro-F_1.

Kako je zbirka EVEXTRA otprilike tri puta veća od standardno korištene zbirke TimeBank, zanimljivo je vidjeti kako veličina skupa za učenje utječe na uspješnost modela za crpljenje sidara događaja. Tablica 6.2 prikazuje uspješnost modela GRAF-SIDRA na skupu za ispitivanje

Tablica 6.2: Utjecaj veličine skupa za učenje na uspješnost modela za crpljenje sidara činjeničnih događaja

	Identifikacija	Klasifikacija					
% skupa za učenje	Sidro	OCC	IACT	REP	PER	SCH	Prosjek F_1
25%	75,0	78,5	54,3	87,8	76,3	39,2	67,2
50%	77,8	82,7	60,2	89,1	75,9	51,0	71,8
75%	79,2	83,6	61,8	90,5	79,5	55,0	74,1
100%	79,9	84,0	62,4	90,6	83,5	55,1	75,2

ovisno o veličini skupa za učenje, pri čemu su korištene četiri različite veličine skupa za učenje – 25%, 50%, 75% i 100% skupa za učenje korištenog u prethodnim eksperimentima. Analiza uspješnosti klasifikatora ovisno o veličini skupa za učenje daje naslutiti koliko bi se povećanje uspješnosti moglo očekivati kada bismo dalje povećavali ručno označenu zbirku EVEXTRA. Značajno povećanje uspješnosti klasifikacijskih modela uočljivo je prilikom povećanja veličine skupa za učenje sa 25% na 50% te sa 50% na 75% (povećanje uspješnosti za 3–4%). Međutim, pri daljnjem povećanju veličine skupa za učenje (sa 75% skupa za učenje na potpuni skup za učenje) povećanje uspješnosti klasifikatora značajno je manje ($\approx$ 1%), na temelju čega zaključujemo da je malo vjerojatno da bismo postigli značajna poboljšanja uspješnosti klasifikacije kada bismo sidrima događaja ručno označili dodatne dokumente. S druge strane, povećanje kvalitete postojećih oznaka u zbirci EVEXTRA moglo bi dovesti do većih poboljšanja u uspješnosti automatskog crpljenja sidara činjeničnih događaja.

6.1.3 Analiza pogrešaka

Detaljna analiza primjera na kojima klasifikator za prepoznavanje sidara događaja griješi otkriva nekoliko čestih tipova pogrešaka:

1. Imenice koje u većini slučajeva opisuju generičku situaciju i kao takve nisu označene kao sidra događaja (npr. "*violence*", "*war*") klasifikator ne prepoznaje kao sidra događaja niti onda kada se odnose na konkretan događaj (npr. "*at least 41 people, mostly civilians, have been killed in violence that prompted UN...*"). Ovakvi slučajevi predstavljaju lažne negativne primjere (sidra događaja koje klasifikator ne prepoznaje kao takve) i smanjuju odziv klasifikatora;
2. Glagoli u infinitivu podjednako često instanciraju i činjenična (npr. "*Violence forced the Arab League to end its own monitoring mission*") i nečinjenična spominjanja događaja (npr. "*Clinton underlined that the United States want to see continuing progress*"). Klasi-

fikator često nije u stanju razlikovati odnosi li se glagol u infinitivu na konkretan ostvareni događaj ili ne, što uzrokuje pojavu kako lažnih pozitivnih primjera (infinitiv koji nije sidro činjeničnog događaja, a klasifikator ga proglasi takvime), tako i lažnih negativnih primjera (infinitivi koji instanciraju stvarne događaje, ali ih klasifikator ne prepoznaje kao takve);

3. Nečinjenična spominjanja događaja ponekad imaju udaljene i implicitne indikatore činjeničnosti koje klasifikator nije u stanju prepoznati (npr. "*This raises the possibility that Al-Qaeda fighters are inflitrating across the border*"). Ovakvi primjeri rezultiraju lažnim pozitivnim primjerima i smanjuju preciznost klasifikatora.

6.1.4 Vrednovanje modela na zbirci TimeBank

Rezultati modela za crpljenje sidara činjeničnih događaja nisu izravno usporedivi s rezultatima modela nadziranog strojnog učenja za crpljenje događaja koji su učeni na zbirci TimeBank zbog razlike između definicije događaja koje se držimo u ovoj disertaciji (ne razmatraju se stanja i nečinjenična spominjanja događaja) i definicije događaja u standardu TimeML (v. odjeljak4.1.2). Unatoč tome, a isključivo u cilju usporedbe s najuspješnijim modelima, model GRAF-SIDRA vrednujemo i na zbirci TimeBank. Preciznije, vrednujemo dva modela: (1) model GRAF-SIDRA-TIMEBANK, koji koristi sve prethodno navedene značajke, ali na zbirci TimeBank; (2) model GRAF-SIDRA izgrađen na zbirci EVEXTRA vrednovan na zbirci TimeBank. Uspješnost tih modela na zbirci TimeBank dana je u tablici 6.3 zajedno s rezultatima najuspješnijih modela za crpljenje događaja na zbirci TimeBank. Rezultati pokazuju da je model GRAF-SIDRA-TIMEBANK po uspješnosti sumjerljiv najuspješnijim modelima koji ekstrahiraju sidra događaja na temelju zbirke TimeBank, iako značajke oblikovane sa svrhom da razlikuju događaje od stanja te činjenične od nečinjeničnih spominjanja događaja, a koje taj model koristi, na zbirci TimeBank imaju zanemariv utjecaj. Važno je uočiti kako pri prepoznavanju sidara događaja model GRAF-SIDRA-TIMEBANK ima značajno bolju preciznost od najuspješnijih modela iz literature (Llorens i dr., 2010; Grover i dr., 2010). Međutim, odziv modela GRAF-SIDRA-TIMEBANK lošiji je od odziva najuspješnijih modela, što je vjerojatno posljedica nedostatka značajki usmjerenih na prepoznavanje stanja i nečinjeničnih spominjanja događaja. Izvorni model GRAF-SIDRA, koji je učen na oznakama zbirke EVEXTRA, također ima visoku preciznost na oznakama zbirke TimeBank. Njegov je odziv, međutim, očekivano znatno lošiji od odziva modela GRAF-SIDRA-TIMEBANK, koji je i učen na oznakama zbirke TimeBank, budući da su mu na skupu za učenje stanja i nečinjenični događaji dani kao negativni primjeri, dok su na skupu za ispitivanje takvi primjeri označeni kao pozitivni.

Tablica 6.3: Uspješnost modela za crpljenje sidara događaja na zbirci TimeBank

Identifikacija sidara	P	R	F_1					
Llorens et al. (2010)	81,0	**86,0**	**83.0**					
Grover et al. (2010)	75,0	85,0	80,0					
GRAF-SIDRA-TIMEBANK	**87,0**	75,1	80,6					
GRAF-SIDRA	84,2	61,0	70,8					
Klasifikacija u semantičke razrede[a]	OCC	IST	ST	REP	PER	IAC	AS	Prosjek F_1
Llorens et al. (2010)	–	–	–	–	–	–	–	**79,0**
UzZaman and Allen (2010)	–	–	–	–	–	–	–	77,0
EVGRAPH-ANCHORS-TIMEBANK	84,7	62,0	50,2	91,4	72,0	44,1	68,5	78,3

[a]OCC – OCCURRENCE; IST – I-STATE; ST – STATE; REP – REPORTING; PER – PERCEPTION; IAC – I-ACTION; AS – ASPECTUAL. Posljednji stupac predstavlja prosjek rezultata F_1 po razredima.

6.2 Crpljenje argumenata događaja

Budući da želimo ekstrahirati argumente događaja četiriju općenitih semantičkih uloga (AGENT, TARGET, TIME, LOCATION), za crpljenje argumenata događaja odabran je pristup koji se temelji na relativno malenom skupu sintaktičkih pravila koja pouzdano prepoznaju argumente događaja i njihove semantičke uloge na temelju ovisnosnih sintaktičkih relacija između argumenta i sidra događaja. Ovisnosne relacije, kao rezultat ovisnosnog parsanja, određuju sintaktičku strukturu rečenice. Preciznije, rezultat ovisnosnog parsanja rečenice jest ovisnosno stablo čiji su bridovi pojedinačne ovisnosne relacije. Svaka ovisnosna relacija definira konkretan sintaktički odnos između dvije riječi rečenice, upravljačke riječi i zavisne riječi, gdje upravljačka riječ (engl. *governing word*) sintaktički upravlja zavisnom rječju (engl. *dependent word*) (Agić, 2012b). Model za crpljenje argumenata događaja na svom ulazu očekuje tekst u kojem su označena sidra događaja te koristi ovisnosne sintaktičke relacije sidara događaja za određivanje izraza koji odgovaraju argumentima tih događaja. Svako pojedinačno pravilo za crpljenje argumenata definira uzorak ovisnosnih sintaktičkih relacija (engl. *dependency pattern*) koje određuju argument događaja i skup semantičkih uloga koje tako određen argument može imati. Svaki uzorak definira skup ovisnosnih sintaktičkih relacija koje povezuju sidro događaja i glavnu riječ argumenta (engl. *argument head word*).

Crpljenje argumenata provodi se u dva koraka: (1) prepoznavanjem argumenata, gdje se temeljem sintaktičkih uzoraka određuju kandidati za argumente događaja i (2) klasifikacijom semantičkih uloga argumenata. Neki su sintaktički uzorci jednoznačni u smislu semantičke uloge argumenta koju određuju (tj. istovremeno identificiraju izraz koji je argument događaja i odre-

đuju semantičku ulogu tog argumenta). Na primjer, sintaktička uloga nominalnog subjekta u pravilu odgovora semantičkoj ulozi AGENT. Druga su pravila, međutim, višeznačna po pitanju semantičke uloge predikata. Tako, primjerice, prijedložni objekt sidra događaja može biti argument s ulogom TARGET (npr. "... *competing against 48 opponents*"), argument s ulogom TIME (npr. "... *competing on Saturday*") ili pak argument s ulogom LOCATION (npr. "... *competing around South Farallon Island*"). U takvim slučajevima potrebno je provesti razrješavanje višeznačnosti semantičke uloge argumenta što se obavlja prepoznavanjem imenovanih entiteta, prepoznavanjem vremenskih izraza te usporedbom glavne riječi argumenta s lokacijskim i vremenskim konceptima iz leksičko-semantičke mreže WordNet, ali i na temelju prijedloga koji povezuju sidro i argument.

6.2.1 Prepoznavanje argumenata

Sintaktički motivirana pravila za prepoznavanje argumenata događaja definirana su pomoću ovisnosnih sintaktičkih relacija koje kao izlaz daje Stanfordov ovisnosni parser (De Marneffe i dr., 2006; De Marneffe i Manning, 2008). Sintaktički uzorci za prepoznavanje argumenata događaja oblikovani su na temelju analize ručno označenih argumenata događaja na skupu od 25 dokumenata korištenih za razvoj modela (v. odjeljak 5.2). Skup uzoraka za crpljenje argumenata događaja građen je inkrementalno, dodavanjem uzoraka koji povećaju odziv, a pritom ne narušavaju značajno preciznost. Pri tome je prednost dana općenitijim uzorcima (tj. uzorcima s manje ovisnosnih relacija) kako bi se smanjio rizik od prenaučenosti modela na primjere iz 25 dokumenata skupa za razvoj pravila. Također u cilju smanjenja rizika od prenaučenosti modela, nijedan korišteni sintaktički uzorak nije leksikaliziran (tj. uzorak je definiran samo skupom sintaktičkih relacija, a ne riječima ili lemama koje sudjeluju u tim relacijama).

Sintaktičke uzorke za prepoznavanje argumenata događaja dijelimo na *ekstraktivne uzorke* i *distributivne uzorke*. Ekstraktivni uzorci identificiraju glavnu riječ argumenta (engl. *head word*) i povezuju je sa sidrom događaja. Ukupno je definirano 11 ekstraktivnih uzoraka za određivanje argumenata događaja (svi ekstraktivni uzorci koncizno su prikazani u tablici 6.4):

- *Nominalni subjekt* (engl. *nominal subject*) uzorak je kojim se definira da je svaka riječ sa sintaktičkom ulogom nominalnog subjekta za predikat koji je sidro događaja ujedno i glavna riječ argumenta tog događaja sa semantičkom ulogom AGENT. U rečenici "*A wave swept four crew members*", riječ "*wave*" glavna je riječi argumenta "*A wave*" sa semantičkom ulogom AGENT za događaj određen sidrom "*swept*" zato što je "*wave*" nominalni subjekt predikata "*swept*" (tj. postoji sintaktička relacija *nsubj*("*swept*", "*wave*"));
- *Nominalni subjekt pasivnog predikata* (engl. *passive nominal subject*) uzorak je koji definira da je nominalni subjekt predikata u pasivu (uz pretpostavku da je taj predikat sidro događaja) argument događaja sa semantičkom ulogom TARGET. Tako je u rečenici "*The boat was pushed on the rocks*" riječ "*boat*" nositelj argumenta sa semantičkom ulogom

TARGET jer postoji sintaktička relacija *nsubjpass*(“*pushed*”, “*boat*”) po kojoj je “*boat*” nominalni subjekt pasivnog predikata “*pushed*”;

- *Izravni objekt* (engl. *direct object*) uzorak je koji određuje da je izravni objekt predikata koji je sidro događaja argument tog događaja sa semantičkom ulogom TARGET ili LOCATION. Tako je, na primjer, u rečenici “*A wave swept four crew members*” riječ “*members*” glavna riječ argumenta za događaj ukorjenjen sidrom “*swept*” zato što postoji ovisnosna relacija *dobj*(“*swept*”, “*members*”) koja definira da je “*members*” izravni objekt predikata “*swept*”;
- *Neizravni objekt* (engl. *indirect object*) uzorak je koji određuje da je neizravni sintaktički objekt predikata koji je sidro događaja ujedno argument tog događaja sa semantičkom ulogom TARGET. U primjeru “*Stallion allowed Chase a head start*”, riječ “*Chase*” argument je događaja (sa semantičkom ulogom TARGET) instanciranog sidrom “*allowed*” jer postoji ovisnosna relacija *iobj*(“*allowed*”, “*Chase*”) koja određuje da je “*Chase*” neizravni sintaktički objekt predikata “*allowed*”;
- *Prijedložni objekt* (engl. *prepositional object*) uzorak je koji definira da je izraz koji je s predikatom povezan preko prijedloga kandidat za argument događaja. Prijedložni objekti predikata mogu imati sve četiri semantičke uloge (AGENT, TARGET, TIME i LOCATION). U rečenici “*Stallion raced on Sunday*”, riječ “*Sunday*” prijedložni je objekt predikata “*raced*” (postoje ovisnosne relacije *prep*(“*raced*”, “*on*”) i *pobj*(“*on*”, “*Sunday*”)), pa je ujedno i argument događaja ukorjenjenog sidrom “*raced*”;
- *Modifikacija participnom podrečenicom* (engl. *participial modifier*) uzorak je kojim se definira da je argument događaja svaka riječ modificirana podrečenicom u kojoj je sidro tog događaja predikat, pri čemu takav argument onda može imati semantičku ulogu AGENT ili TARGET. Na primjer, u rečenici “*The boat suffering damage*”, riječ “*boat*” modificirana je podrečenicom “*suffering damage*” (postoji ovisnosna relacija *partmod*(“*boat*”, “*suffering*”)) u kojoj je sidro događaja “*suffering*” predikat;
- *Modifikacija relativnom podrečenicom* (engl. *relative clause modifier*) uzorak je kojim se definira da je glavna riječ argumenta događaja ona riječ koja je modificirana relativnom podrečenicom (tipično podrečenice uvedene s “*which*”, “*that*” ili “*who*”) u kojoj je sidro događaja predikat. Pri tome argument ima semantičku ulogu AGENT ukoliko je predikat u aktivu, a ulogu TARGET ako je predikat u pasivu. U rečenici “*The boat which won the race*”, riječ “*boat*” argument je događaja “*won*” jer relativna podrečenica u kojoj je “*won*” predikat ima funkciju pobližeg označavanja riječi “*boat*” (tj. postoje ovisnosne relacije *rcmod*(“*boat*”, “*won*”) i *nsubj*(“*won*”, “*which*”));
- *Modifikacija infinitivnom podrečenicom* (engl. *infinitival modifier*) uzorak je po kojem je argument događaja sa semantičkom ulogom AGENT onaj izraz koji je modificiran podrečenicom koja je uvedena predikatom u infinitivu (a koji je sidro događaja). U primjeru

"*The first boat to beat Stallion was Chase*", argument s ulogom AGENT događaja "*beat*" je "*Chase*", budući da se upravo na tu vlastitu imenicu odnosi podrečenica "*The first boat to beat Stallion*" uvedena predikatom u infinitivu ("*to beat*") (tj. postoje ovisnosne relacije *infmod*("*boat*", "*beat*") i *nsubj*("*Chase*", "*boat*"));

- *Posvojni modifikator* (engl. *possession modifier*) uzorak je po kojem vremenske izraze i imenovane entitete koji su posvojni modifikatori imeničkih sidara događaja proglašavamo argumentima tih događaja sa semantičkom ulogom TIME ili AGENT. U izrazu "*Yesterday's race*", riječ "*Yesterday*" ima sintaktičku ulogu posvojnog modifikatora imenice "*race*", koja je sidro događaja. Budući da je "*Yesterday*" ujedno i vremenski izraz, smatramo ga argumentom s ulogom TIME za događaj "*race*";
- *Imenički izraz* (engl. *noun compound*) uzorak je kojim se kao argumenti događaja prepoznaju izrazi koji su vlastita imena ili vremenski izrazi, a dio su kompozicije imenica u kojoj je glavna imenica sidro događaja. Na primjer, u imeničkom izrazu "*Sunday race*", riječ "*Sunday*" argument je događaja "*race*" jer je vremenski izraz koji je u imeničkoj kompoziciji sa sidrom događaja (tj. postoji ovisnosna relacija *nn*("*race*", "*Sunday*"));
- *Otvoreni komplement podrečenice* (engl. *open clausal complement*) uzorak je kojim sidru događaja koje je predikat podrečenice pridjeljujemo argument sa semantičkom ulogom AGENT koji je sintaktički izravno vezan na predikat glavne rečenice, a koji može i ne mora biti sidro događaja. U primjeru "*Stallion managed to get past Chase*", riječ "*Stallion*" subjekt je predikata glavne rečenice "*managed*", koji pak upravlja sidrom događaja "*get*", koje je predikat podrečenice "*to get past Chase*" (tj. postoje ovisnosne relacije *xcomp*("*managed*", "*get*") i *nsubj*("*managed*", "*Stallion*")), pa stoga "*Stallion*" proglašavamo argumentom sa semantičkom ulogom AGENT za događaj "*get*".

Ekstraktivnim uzorcima prepoznajemo glavne riječi argumenta, ali ne i cijele izraze (tj. cjelokupna prostiranja) argumenata. Cjelokupni izraz argumenta događaja određujemo kao sintaktički odsječak kojem prethodno prepoznata glavna riječ argumenta pripada. Za potrebe plitke sintaktičke analize kojom se određuju sintaktički odsječci rečenice koristi se plitki sintaktički analizator za engleski jezik (engl. *chunker*) iz programske biblioteke OpenNLP.[2]

Distributivni uzorci za crpljenje argumenata događaja potrebni su nam u situacijama kada je neki izraz argument više događaja, što ćemo nazivati *dijeljenjem argumenata* ili kada jedno sidro događaja na sebe veže više argumenata s istom semantičkom ulogom, što ćemo nazivati *konjunkcijom argumenata*. Dijeljenje argumenata primjenjujemo u slučajevima kada je neki izraz argument za dva ili više događaja, ali je sintaktički vezan na sidro samo jednog od događaja. Tada pravila za dijeljenje argumenata koristimo kako bismo promatrani izraz označili argumentom ostalih događaja. Promotrimo sljedeći primjer u kojem se pojavljuju dva spominjanja događaja:

[2] `http://opennlp.apache.org/`

Tablica 6.4: Ekstraktivni uzorci za crpljenje argumenata događaja

Naziv uzorka	Ov. relacije	Primjer	Uloga
Nominalni subjekt	$nsubj(x,y)$	"***A wave*** *swept four crew members.*" ($x =$ "*swept*", $y =$ "*wave*")	Agent
Nom. sub. pasiv	$nsubjpass(x,y)$	"***The boat*** *was pushed on the rocks.*" ($x =$ "*pushed*", $y =$ "*boat*")	Target
Izravni objekt	*dobj(x, y)*	"*A wave swept* ***four crew members***." ($x =$ "*swept*", $y =$ "*members*")	Target
		"*Stallion rounded* ***Farallon Island***" ($x =$ "*rounded*", $y =$ "*Island*")	Location
Neizravni objekt	$iobj(x,y)$	"*Stallion allowed* ***Chase*** *a head start*" ($x =$ "*allowed*", $y =$ "*Chase*")	Target
Prijedložni objekt	$prep(x,z)$, $pobj(z,y)$	"*Stallion raced* ***on Sunday***" ($x =$ "*raced*", $y =$ "*Sunday*", $z =$ "*on*")	Time
		"*Stallion raced* ***around Farallon Island***" ($x =$ "*raced*", $y =$ "*Island*", $z =$ "*around*")	Location
		"*Stallion competed against* ***Chase***" ($x =$ "*competed*", $y =$ "*Chase*", $z =$ "*against*")	Target
		"*The conflict was initiated by* ***Chase***" ($x =$ "*initiated*", $y =$ "*Chase*", $z =$ "*by*")	Agent
Modifikacija participnom podrečenicom	$partmod(x,y)$	"***The boat*** *suffering damage*" ($x =$ "*boat*", $y =$ "*suffering*")	Agent
		"***The boat*** *sunk on the shore*" ($x =$ "*boat*", $y =$ "*sunk*")	Target
Modifikacija relativnom podrečenicom	$rcmod(x,y)$, $nsubj(y,z)$	"***The boat*** *which won the race*" ($x =$ "*boat*", $y =$ "*won*", $z =$ "*which*")	Agent
		"***The crewmen*** *storm swept*" ($x =$ "*crewmen*", $y =$ "*swept*", $z =$ "*storm*")	Target
Modifikacija infinitivnom podrečenicom	$infmod(x,y)$, $nsubj(z,x)$	"*The first boat to beat Stallion was* ***Chase***." ($x =$ "*boat*", $y =$ "*beat*", $z =$ "*Chase*")	Agent
Posvojni modifikator	$poss(x,y)$	"***Yesterday****'s race*" ($x =$ "*race*", $y =$ "*Yesterday*")	Time
		"***Stallion****'s victory*" ($x =$ "*victory*", $y =$ "*Stallion*")	Agent
Imenički izraz	$nn(x,y)$	" ***Farallon Island*** *race*" ($x =$ "*race*", $y =$ "*Island*")	Location
		"***Sunday*** *competition*" ($x =$ "*competition*", $y =$ "*Sunday*")	Time
		"***Stallion*** *initiative*" ($x =$ "*initiative*", $y =$ "*Stallion*")	Agent
Otv. kompl. podrečenice	$xcomp(x,y)$, $nsubj(x,z)$	"***Stallion*** *managed to get past Chase*" ($x =$ "*managed*", $y =$ "*get*", $z =$ "*Stallion*")	Agent

(19) *Low Speed Chase hit and damaged the Stallion.*

Izraz "*Low Speed Chase*" u prethodnom je primjeru argument sa semantičkom ulogom AGENT, kako za događaj "*hit*", tako i za događaj "*damaged*". Slično, izraz "*the Stallion*" argument je s ulogom TARGET za oba događaja. Ekstraktivnim uzorcima za crpljenje argumenata događaja, međutim, izraz "*Low Speed Chase*" bit će prepoznat kao argument sa semantičkom ulogom AGENT samo za događaj "*hit*" (uzorak *Nominalni subjekt*), a izraz "*the Stallion*" kao argument sa semantičkom ulogom TARGET samo za događaj "*damaged*" (uzorak *Izravni objekt*). Stoga je potrebno definirati pravila koja će izraz "*Low Speed Chase*" pridijeliti kao argument sidru "*damaged*" te izraz "*the Stallion*" kao argument sidru "*hit*". Dijeljenje argumenata provodimo kada postoje sljedeće sintaktičke veze između sidara događaja:

1. *Konjunkcija* označava postojanje ovisnosne sintaktičke relacije *conj* između sidara događaja (npr. *conj*("*hit*", "*damaged*") u primjeru "*Low Speed Chase hit and damaged the Stallion.*").
2. *Otvoreni komplement podrečenice*, označen sintaktičkom relacijom *xcomp* između sidara događaja označava da je jedno sidro događaja predikat podrečenice koja je komplement glavne rečenice u kojoj je drugo sidro predikat (npr. *xcomp*("*managed*", "*avoid*") u rečenici "*Stallion managed to avoid the disaster*").

Pravila za dijeljenje argumenata primjenjuju se samo ako događaj kojem se treba pridružiti argument nema argument s istom semantičkom ulogom koji mu je pridružen prethodnom primjenom nekog od ekstraktivnih uzoraka. Razmotrimo primjer

(20) *The Stallion provoked and the Low Speed Chase answered.*

U ovom će slučaju na temelju ekstraktivnih uzoraka izraz "*The Stallion*" događaju "*provoked*" biti pridružen kao argument s ulogom AGENT, a izraz "*the Low Speed Chase*" kao argument s istom ulogom događaju "*answered*". Kako će nakon primjene ekstraktivnih uzoraka svaki od događaja već imati pridijeljen argument sa semantičkom ulogom AGENT, izraz "*The Stallion*" nećemo pridružiti kao argument događaju "*answered*" niti ćemo izraz "*the Low Speed Chase*" pridijeliti događaju "*provoked*", premda postoji sintaktička konjunkcija između sidara "*provoked*" i "*answered*".

Pravilo za *konjunkciju argumenata* primjenjuje se u slučajevima kada događaj ima više argumenata između kojih postoji sintaktička konjunkcija. U takvim slučajevima ekstraktivni će uzorci pridijeliti samo jedan argument sidru događaja, pa ovo pravilo služi tome da se događaju pridjele i preostali argumenti s kojima nije u vezi definiranoj nekim od ekstraktivnih uzoraka. Ilustrirajmo ovo na sljedećem primjeru:

(21) *The Stallion and Low Speed Chase were seriously damaged in the crash.*

Izrazi "*The Stallion*" i "*Low Speed Chase*" argumenti su s ulogom TARGET za događaj "*damaged*", ali će primjenom ekstraktivnih uzoraka (konkretno primjenom uzorka *Nominalni subjekt pasivnog predikata*) samo izraz "*Low Speed Chase*" biti pridružen kao argument sidru "*damaged*". Ipak, sintaktička konjunkcija između izraza "*The Stallion*" i "*Low Speed Chase*" prepoznata na temelju ovisnosne relacije *conj*("*Stallion*", "*Chase*")) omogućava da izraz "*The Stallion*" također pridružimo kao argument događaju "*damaged*".

Potrebno je naglasiti da su predstavljeni ekstraktivni i distributivni uzorci za crpljenje argumenata događaja posebno oblikovani za engleski jezik. Prilagodba modela za crpljenje argumenata događaja za druge jezike iziskivala bi izgradnju sličnih sintaktičkih uzoraka za jezike poput hrvatskoga koji imaju značajno drugačiju sintaksu u odnosu na engleski jezik, odnosno u najmanju ruku prilagodbu opisanih uzoraka za jezike koji su sintaktički bliski engleskome jeziku.

6.2.2 Razrješavanje semantičkih uloga

Pojedini ekstraktivni uzorci (u prvom redu uzorak *Prijedložni objekt*) samo određuju glavnu riječ argumenta događaja, ali ne i semantičku ulogu argumenta jer pojedinim sintaktičkim ulogama može odgovarati više semantičkih uloga. Za takve je uzorke potreban dodatni postupak kojim se jednoznačno određuje semantička uloga argumenta događaja. Stoga se za svaki semantički nejednoznačan ekstraktivni uzorak definiraju pravila za razrješavanje semantičke nejednoznačnosti argumenata koja se temelje na (1) prepoznavanju imenovanih entiteta (Finkel i dr., 2005), (2) crpljenju vremenskih izraza (Chang i Manning, 2012) i (3) mjerenju semantičke sličnosti između glavne riječi argumenta i lokacijskih i vremenskih koncepata (Wu i Palmer, 1994).

Na ovom mjestu opisujemo postupak za razrješavanje nejednoznačnosti za uzorak *Prijedložni objekt* koji je semantički najvišeznačniji ekstraktivni uzorak jer prijedložni objekt sidra događaja može biti argument svih četiriju semantičkih razreda argumenata. Pravila postupka za razrješavanje višeznačnosti semantičke uloge prijedložnog objekta su sljedeća (pravila se primjenjuju redoslijedom kojim su navedena):

1. Ukoliko je glavna riječ argumenta dio imenovanog entitet tipa *Lokacija*,[3] argumentu se dodjeljuje semantička uloga LOCATION. Ako je glavna riječ argumenta dio vremenskog izraza, argumentu se dodjeljuje semantička uloga TIME;[4]
2. Ukoliko je glavna riječ argumenta dio imenovanog entiteta tipa *Osoba* ili *Organizacija*, a prijedlog koji povezuje sidro i glavnu riječ argumenta je "*by*", tada se argumentu dodjeljuje semantička uloga AGENT; ako je prijedlog između sidra i glavne riječi argumenta "*against*", tada se argumentu dodjeljuje semantička uloga TARGET;

[3]Za prepoznavanje imenovanih entiteta koristi se alat Stanford NER (Finkel i dr., 2005).
[4]Za crpljenje vremenskih izraza iz teksta koristi se alat SUTime (Chang i Manning, 2012)

3. Ukoliko je prijedložni objekt uveden nekim od vremenskih signala (*before*, *after*, *during*, *since* i *until*), argumentu se dodjeljuje semantička uloga TIME; ako se radi o lokacijskom prijedlogu (*above*, *below*, *behind*, *inside*, *around*, *along*, *across*, *near* i *through*), argumentu se dodjeljuje semantička uloga LOCATION. Neki od prijedloga koji uobičajeno imaju vremensko značenje (npr. "*before*") ponekad mogu biti u službi lokacijskog signala (npr. "*He stood before the building*"), kao što i neki prijedlozi koji uobičajeno imaju lokacijsko značenje (npr. "*near*") mogu biti u funkciji vremenskih signala (npr. "*The bomb exploded near midnight*") (Derczynski i Gaizauskas, 2013). Pragmatičan pristup, koji koristimo u ovoj disertaciji, vremenskim i lokacijskim signalima smatra prijedloge koji su na skupu za razvoj modela (v. odjeljak 5.2) u najvećem broju slučajeva imali funkciju vremenskog odnosno lokacijskog signala.

Ukoliko semantičku ulogu argumenta događaja nije moguće odrediti niti jednim od prethodna tri pravila, tada se pristupa mjerenju semantičke sličnosti između glavne riječi argumenta i lokacijskih odnosno vremenskih koncepata na temelju leksičko-semantičke mreže WordNet. Definiraju se tri skupa koncepata s kojima se uspoređuje glavna riječ argumenta događaja – skup vremenskih koncepata T, skup lokacijskih koncepata L te skup općenitih koncepata G:

$$
\begin{aligned}
L &= \{\mathit{location},\ \mathit{geographical\_area},\ \mathit{vehicle},\ \mathit{building},\ \mathit{transport},\ \mathit{facility}\} \\
T &= \{\mathit{time},\ \mathit{duration},\ \mathit{time\_unit},\ \mathit{clock\_time},\ \mathit{time\_period}\} \\
G &= \{\mathit{object},\ \mathit{concept},\ \mathit{person},\ \mathit{event}\}
\end{aligned}
$$

Koncepti u skupovima L i T predstavljaju najčešće hiperonime glavnih riječi vremenskih i lokacijskih argumenata događaja koji su ručno označeni na skupu za razvoj modela. Na primjer, uočeno je kako su lokacijski argumenti vrlo često vozila (npr. "*in the car*") ili građevine (npr. "*near the church*"). Koncepti u skupu općenitih koncepata G najopćenitiji su koncepti u imeničkoj hijerarhiji leksičko-semantičke mreže WordNet koji nemaju vremensko niti lokacijsko značenje. Semantička sličnost riječi w i skupa koncepata S odgovara najvećoj semantičkoj sličnosti između w i nekog od koncepata iz S:

$$
\mathit{ssim}(w,S) = \max_{s \in S} \left(\mathit{wupalmer}(w,s)\right) \tag{6.1}
$$

gdje je $\mathit{wupalmer}(w,s)$ semantička sličnost između koncepata w i s izmjerena na WordNet hijerarhiji koncepata algoritmom Wua i Palmer (1994). Semantička uloga argumenta određuje se na temelju odnosa semantičke sličnosti glavne riječi argumenta w i skupova koncepata L, T i G. Argumentu se pridjeljuje semantička uloga LOCATION ako vrijedi $\mathit{ssim}(w,L) > \mathit{ssim}(w,T)$ i $\mathit{ssim}(w,L) > \mathit{ssim}(w,G)$. Istovjetno, argumentu se pridjeljuje semantička uloga TIME ako vrijedi $\mathit{ssim}(w,T) > \mathit{ssim}(w,L)$ i $\mathit{ssim}(w,T) > \mathit{ssim}(w,G)$.

6.2.3 Vrednovanje modela

Uspješnost modela za crpljenje argumenata događaja mjerimo na skupu od 80 novinskih članaka ručno označenih argumentima događaja (v. poglavlje 5.2). Kako za crpljenje argumenata sa semantičkim ulogama grube zrnatosti ne postoji standardna temeljna metoda, u cilju usporedbe uspješnosti modela s nekom drugom metodom kao temeljna metoda odabran je model koji sadrži samo najintuitivnija pravila za prepoznavanje argumenata događaja:

1. Svaki nominalni subjekt predikata koji je sidro događaja proglašava se argumentom tog događaja s ulogom AGENT;
2. Svaki izravni objekt predikata koji je sidro događaja proglašava se argumentom tog događaja s ulogom TARGET;
3. Svaki prijedložni objekt predikata koji je sidro događaja, a koji je dio lokacijskog imenovanog entiteta, proglašava se argumentom tog događaja s ulogom LOCATION;
4. Svaki prijedložni objekt predikata koji je sidro događaja, a koji je dio vremenskog izraza, proglašava se argumentom tog događaja s ulogom TIME.

U tablici 6.5 prikazana je uspješnost modela za crpljenje argumenata događaja (GRAF-ARGUMENTI) zajedno s uspješnošću upravo opisane temeljne metode. Uspješnost je mjerena standardnim mjerama: preciznošću, odzivom i mjerom F_1. Posebno je navedena točnost crpljenja glavnih riječi argumenata (stupci 3–5), a posebno točnost prepoznavanja cjelokupnog prostiranja argumenta (stupci 6–8). Kod vrednovanja prepoznavanja cjelokupnog argumenta, vrednovanje je provedeno *strogo*, što znači da se kao točni vrednuju samo oni argumenti koji se s nekim od ručno označenih argumenata poklapaju u svim pojavnicama. Rezultati sugeriraju da je model za crpljenje argumenata događaja temeljen na sintaktičkim uzorcima vrlo pogodan za prepoznavanje glavnih riječi argumenata, dok je uspješnost prepoznavanja cjelokupnih argumenata temeljena na plitkoj sintaktičkoj analizi nešto manja.

Dva su glavna uzorka nižoj uspješnosti potpuno točnog prepoznavanja prostiranja argumenata događaja. Kao prvo, uspješnost prepoznavanja prostiranja argumenta mjeri se strogim podudaranjem s ljudskim oznakama pa i najmanja odstupanja poput izostavljanja članova (npr. "*the Low Speed Chase*" nasuprot samo "*Low Speed Chase*") u ručnim oznakama uzrokuju "netočna" crpljenja. Kao drugo, ručno označena prostiranja argumenata često su šira od sintaktičkog odsječka kojem pripada glavna riječ argumenta. Tako je, na primjer, u rečenici

(22) *The vessel Low Speed Chase, a 12m racing sailboat with a crew of eight, <u>competed</u> on Saturday in a race around South Farallon Island.*

ručno označeno prostiranje argumenta s ulogom AGENT za događaj "*competed*" izraz "*The vessel Low Speed Chase, a 12m racing sailboat with a crew of eight*", dok je sintaktički odsječak glavne riječi "*Chase*" samo izraz "*The vessel Low Speed Chase*".

Model GRAF-ARGUMENTI postiže značajno bolje rezultate od temeljne metode, kako pri crpljenju glavnih riječi argumenata, tako i pri ekstrakciji cjelokupnih argumenata. Temeljna

Tablica 6.5: Uspješnost modela za crpljenje argumenata događaja

		Glavna riječ			Prostiranje		
	Semantička uloga	P	R	F_1	P	R	F_1
Temeljna metoda	AGENT	93,1	58,5	71,8	72,8	45,7	56,2
	TARGET	88,2	55,5	68,1	63,6	40,0	49,1
	TIME	78,1	23,6	36,2	50,0	15,2	23,3
	LOCATION	79,1	16,7	27,6	62,8	13,4	22,0
	Prosjek	84,6	38,6	53,0	62,3	28,6	38,8
GRAF-ARGUMENTI	AGENT	85,7	79,8	82,6	63,7	59,4	61,5
	TARGET	81,3	74,6	77,8	58,3	54,0	56,1
	TIME	80,0	64,2	71,2	50,0	43,5	46,5
	LOCATION	66,3	56,6	61,1	34,8	34,1	34,5
	Prosjek	78,3	68,8	**73,3**	51,7	47,8	**49,6**

metoda ima bolju preciznost, ali bitno lošiji odziv, što je očekivano s obzirom da su ekstraktivni uzorci dodavani tako da povećaju odziv, a pritom ne naruše značajno preciznost (tj. tako da se maksimizira uspješnost po mjeri F_1). Uspješnost crpljenja argumenata sa semantičkom ulogom LOCATION značajno je niža nego uspješnost crpljenja argumenata s ulogom TIME, što je posljedica činjenice da su argumenti s ulogom TIME u velikoj većini slučajeva vremenski izrazi (prvo pravilo razrješavanja semantičkih uloga), dok su argumenti s ulogom LOCATION puno rjeđe dio lokacijskih imenovanih entiteta, pa je za njihovo prepoznavanje često potrebno osloniti se na semantičku usporedbu glavne riječi argumenta s lokacijskim konceptima iz leksičko-semantičke mreže WordNet što, čini se, nije posebno pouzdan postupak.

Kako se model za crpljenje argumenata događaja temelji na sintaktičkim uzorcima, najveći broj pogrešnih ekstrakcija argumenata događaja vezan je za rečenice koje su pogrešno sintaktički parsane. Svako pogrešno sintaktičko parsanje rečenice najčešće uzrokuje nekoliko pogrešno prepoznatih argumenata. Poboljšanje modela za ovisnosno parsanje, stoga bi svakako dovelo i do poboljšanja predstavljenog modela za crpljenje argumenata događaja.

6.3 Crpljenje vremenskih odnosa među događajima

Većina pristupa za crpljenje vremenskih odnosa između događaja ograničeni su na način da ili (1) razmatraju samo mali podskup parova događaja iz dokumenta između kojih se određuju vremenski odnosi (npr. samo između glavnih događaja susjednih rečenica ili između događaja iste rečenice gdje jedan događaj sintaktički upravlja drugime) (Bethard, 2008; Verhagen i dr., 2010) ili (2) pretpostavljaju parcijalni vremenski uređaj između događaja, pa vremensku strukturu dokumenata predstavljaju stablima ili usmjerenim acikličkim grafovima (Bramsen i dr., 2006; Kolomiyets i dr., 2012). Oba navedena ograničenja mogu dovesti do toga da neki vremenski odnosi, koje je u načelu moguće prepoznati u tekstu, ostanu neotkriveni. Eventualna sintaktička i diskursna ograničenja smanjuju broj parova događaja između kojih se određuju vremenski odnosi, a pretpostavka da je skup događaja s vremenskim odnosima parcijalno uređen skup implicira da je svaki vremenski odnos refleksivna, tranzitivna i antisimetrična relacija. Budući da ova pretpostavka ne vrijedi za sve vremenske odnose koji nas zanimaju (npr. vremenski odnos OVERLAP nije tranzitivna relacija, a vremenski odnos EQUAL simetrična je relacija) podrazumijevanje parcijalnog uređaja među događajima može dovesti do netočnih vremenskih odnosa (npr. netočan odnos OVERLAP uslijed pretpostavljanja tranzitivnosti koja zapravo ne vrijedi).

S druge strane, parove događaja između kojih se izravno određuju vremenski odnosi potrebno je ipak ograničiti na događaje koji su u tekstu relativno blizu jedni drugima. Izravno određivanje vremenskih odnosa između događaja čija su spominjanja udaljena u tekstu (npr. jedno spominjanje na početku, a drugo na kraju velikog novinskog članka) nepraktično je s obzirom na količinu teksta koja se nalazi između spominjanja (tj. količinu informacije koju je potrebno razumjeti). Posljedično, ručno označavanje vremenskih odnosa između svih parova događaja u dokumentima bilo bi preskupo i predugo bi trajalo. Konačno, izgradnja računalnog modela koji bi bio u stanju "razumjeti" značenje teksta između dvaju udaljenih spominjanja događaja zahtjevniji je zadatak od izgradnje modela za otkrivanje vremenskih odnosa između bliskih događaja. U skladu s ovim opažanjima, ali i s teorijom koherentnosti diskursa (engl. *discourse coherence*) (Wolf i Gibson, 2005), prema kojoj se povezani događaji u tekstu spominju blizu jedni drugima, za izgradnju modela za izravno crpljenje vremenskih odnosa između događaja razmatramo sve parove događaja iz iste ili susjednih rečenica, ali ne uvodimo dodatne kriterije na spominjanja događaja (kakav je, na primjer, kriterij konkretnog sintaktičkog odnosa između spominjanja događaja).

Postupak za crpljenje vremenskih odnosa među događajima odvija se u dva koraka. U prvom koraku za zadani par spominjanja događaja potrebno je odrediti je li vremenski odnos među njima moguće odrediti iz teksta. U drugom koraku određujemo vrstu vremenskog odnosa samo za one parove spominjanja događaja za koje je u prethodnom koraku utvrđeno da vremenski odnos postoji.

6.3.1 Modeli i značajke

U prvome koraku radi se o binarnome klasifikacijskom problemu (za par spominjanja događaja na temelju teksta se ili može ili ne može odrediti vremenski odnos), dok se u drugom koraku radi o višerazrednoj klasifikaciji (dodjeljivanje konkretnog vremenskog odnosa parovima događaja za koje je taj odnos odrediv). Jedan par spominjanja događaja predstavlja jedan primjer za učenje, pri čemu se događaji uparuju tako da je prvi događaj para uvijek onaj čije se spominjanje ranije nalazi u tekstu. Oba zadatka rješavaju se modelima nadziranog strojnog učenja, pri čemu se u oba slučaja koristi isti opsežan skup značajki:

- Binarna značajka koja označava jesu li spominjanja događaja u istoj ili susjednoj rečenici (f_1^{VO});
- Binarna značajka koja označava radi li se o susjednim spominjanjima događaja (tj. postoje li u tekstu spominjanja događaja koja se nalaze između promatranih spominjanja događaja (f_2^{VO});
- Udaljenost između sidara spominjanja događaja izražena u broju pojavnica (f_3^{VO});
- Riječ (f_4^{VO}), lema (f_5^{VO}), korijen (f_6^{VO}) i vrsta riječi (f_7^{VO}) sidra prvog spominjanja događaja;
- Riječ (f_8^{VO}), lema (f_9^{VO}), korijen (f_{10}^{VO}) i vrsta riječi (f_{11}^{VO}) sidra drugog spominjanja događaja;
- Binarna značajka koja označava jesu li sidra događaja jednaka (f_{12}^{VO});
- Semantička sličnost sidara događaja izmjerena algoritmom Wua i Palmer (1994) na leksičko-semantičkoj mreži WordNet (f_{13}^{VO});
- Vreća riječi koje se nalaze u tekstu između sidara promatranih spominjanja događaja (f_{14}^{VO});
- Sintaktički put između sidara događaja, tj. konkatenacija ovisnosnih relacija koje se nalaze na putu od sidra prvog događaja do sidra drugog događaja u ovisnosnome sintaktičkom stablu rečenice (f_{15}^{VO}). Ova značajka, kao i sve ostale sintaktičke značajke, računaju se samo za primjere u kojima se oba događaja nalaze u istoj rečenici;
- Binarna značajka koja označava upravlja li sidro prvog događaja sintaktički izravno sidrom drugog događaja (f_{16}^{VO}) i, obratno, značajka koja označava upravlja li sidro drugog događaja sintaktički izravno sidrom prvog događaja (f_{17}^{VO}). Definiramo da riječ w_1 izravno sintaktički upravlja drugom riječi w_2 ukoliko postoji ovisnosna relacija između njih $dep(w_1, w_2)$ u kojoj je w_1 upravljačka riječ, a w_2 zavisna riječ;
- *Priložni modifikator* jednog sidra događaja koji uvodi zavisnu surečenicu u kojoj je drugo sidro događaja predikat često je indikator vremenskog odnosa između sidara, pa je predstavljen zasebnom značajkom (f_{18}^{VO}). U primjeru "*The lightning struck before Stallion rounded South Farallon Island*", prijedlog "*before*" uvodi zavisnu surečenicu "*Stallion rounded South Farallon Island*", koja služi kao vremenska priložna oznaka za predikat

glavne surečenice "*The lightning struck*", pa je "*before*" priložni modifikator za sidro "*struck*";

- Modalni glagoli (npr. *can*, *would*, *may*) koji se odnose na sidro prvog (f_{19}^{VO}) odnosno sidro drugog događaja (f_{20}^{VO}), ukoliko takvi modifikatori postoje uz sidra događaja;
- Pomoćni glagoli (npr. *is*, *are*, *have*) koji se odnose na sidra prvog (f_{21}^{VO}) odnosno drugog (f_{22}^{VO}) događaja, ukoliko takvi pomoćni glagoli postoje;
- Članovi (npr. *a*, *an*, *the*, *each*) pridruženi sidru prvog (f_{23}^{VO}) odnosno sidru drugog (f_{24}^{VO}) događaja;
- Binarna značajka koja označava je li sidro prvog događaja negirano (f_{25}^{VO}) te binarna značajka koja označava je li sidro drugog događaja negirano (f_{26}^{VO}).

Kao i kod modela za crpljenje sidara događaja, većina značajki koje koristimo za crpljenje vremenskih odnosa među događajima kategoričke su značajke s velikim brojem mogućih vrijednosti, što rezultira skupom od oko 200.000 binarnih značajki kojima učimo model strojnog učenja. S obzirom na tako velik broj značajki, kao i kod modela za crpljenje sidara događaja, koristimo logističku regresiju kao vrlo uspješan linearni diskriminativni model. U eksperimentima je također korištena implementacija logističke regresije iz paketa LibLinear (Fan i dr., 2008).

6.3.2 Vrednovanje modela

Modeli za crpljenje vremenskih odnosa među događajima vrednuju se na zbirci TimeBank + AQUIANT koja je korištena i u evaluacijskoj kampanji TempEval-2 (Verhagen i dr., 2010). Zbirka TimeBank + AQUIANT sastoji se od 256 dokumenata u kojima je ukupno označen 3.721 vremenski odnos između događaja, od čega se njih 3.190 odnosi na parove događaja koji se nalaze u istoj rečenici. Od ukupno 256 dokumenata, oznake vremenskih odnosa iz njih 180 korištene su učenje modela, dok je preostalih 76 dokumenata korišteno za vrednovanje modela. Kako bi vrednovanje modela bilo pošteno, pri izgradnji skupova za učenje odnosno ispitivanje posebno se vodilo računa o sljedećem:

1. Svi dokumenti iz skupa za učenje nastali su prije svih dokumenata iz skupa za vrednovanje. Na taj način osigurava se da model koji predviđa vremenske odnose na skupu za vrednovanje prilikom učenja ne koristi znanje koje je nastalo vremenski kasnije (engl. *post-hoc temporal knowledge*);
2. Iz AQUIANT dijela zbirke su uklonjeni odlomci koji su doslovne kopije odlomaka iz ranije nastalih dokumenata, kako se ne bi dogodilo da imamo potpuno iste parove događaja i u skupu za učenje i u skupu za ispitivanje.

U zbirci TimeBank + AQUIANT označeni su svi parovi događaja između kojih je na temelju teksta moguće odrediti vremenski odnos (tj. samo pozitivni primjeri za binarni klasifikator koji određuje postoji li vremenski odnos između dva spominjanja događaja). Stoga je u ci-

Tablica 6.6: Uspješnost modela za crpljenje vremenskih odnosa između događaja

Identifikacija vremenskih odnosa	P	R	F1		
SUSJEDNI-DOGAĐAJI	27.5	58.9	37.5		
GRAF-VREMODN-IDENT	79.3	71.3	**75.1**		
Klasifikacijs vremenskih odnosa	Equals	Overlap	Before	After	Macro F1
NAJČEŠĆI-RAZRED	–	–	51.6	–	12.9
GLAGOLSKO-VRIJEME	11.4	46.2	28.4	7.9	23.5
GRAF-VREMODN-KLAS	67.5	47.0	49.8	57.7	**55.5**

lju izgradnje binarnog klasifikatora koji za par događaja odlučuje je li njihov vremenski odnos odrediv iz teksta prvo potrebno pripremiti negativne primjere, odnosno skup parova događaja između kojih nije moguće odrediti vremenski odnos. Kao negativni primjeri korišteni su svi parovi događaja iz susjednih rečenica za koje u zbirci TimeBank + AQUIANT nije bio označen niti jedan vremenski odnos.

U prvom dijelu tablice 6.6 prikazana je uspješnost modela koji određuje postoji li vremenski odnos između dva spominjanja događaja (GRAF-VREMODN-IDENT) zajedno s rezultatima temeljne metode (SUSJEDNI-DOGAĐAJI), koja vremenski povezuje samo susjedne događaje (tj. predviđa da je vremenski odnos odrediv iz teksta samo za susjedna spominjanja događaja). Model GRAF-VREMODN-IDENT značajno je uspješniji od temeljne metode, pri čemu je odziv nešto manji od preciznosti, ali razlika između preciznosti i odziva modela nije velika. U drugom dijelu tablice 6.6 prikazana je uspješnost modela za klasifikaciju vrste vremenskog odnosa (GRAF-VREMODN-KLAS), uspoređena s rezultatima dvaju temeljnih modela – (1) jednostavnog modela (NAJČEŠĆI-RAZRED), koji svakom paru događaja pridjeljuje onu vrstu vremenskog odnosa koja se najčešće pojavljuje na skupu za učenje (na korištenom skupu za učenje to je vremenski odnos BEFORE) i (2) složenijeg temeljnog modela (GLAGOLSKO-VRIJEME) koji paru događaja dodjeljuje vremenski odnos na temelju glagolskog vremena (engl. *tense*) sidara događaja te međusobne udaljenosti spominjanja događaja. Potonji model funkcionira na sljedeći način:

1. Paru događaja u kojem je sidro prvoga događaja u prošlom vremenu, a sidro drugoga događaja u sadašnjem vremenu, temeljni model pridjeljuje vremenski odnos BEFORE. Obratno, ako je sidro prvoga događaja u sadašnjem vremenu, a sidro drugoga događaja u prošlom, paru događaja pridjeljuje se vremenski odnos AFTER;
2. Paru događaja čija su sidra u istom vremenu (oba u prošlom ili oba u sadašnjem vremenu), a nalaze se u istoj rečenici, temeljna metoda GLAGOLSKO-VRIJEME pridjeljuje

vremenski odnos EQUAL;

3. U svim ostalim slučajevima temeljna metoda paru događaja pridjeljuje vremenski odnos OVERLAP.

Model GRAF-VREMODN-KLAS postiže značajno bolje rezultate od oba temeljna modela. Uspješnost modela za klasifikaciju vremenskih odnosa među događajima naslabija je za razred OVERLAP. Nekoliko je mogućih uzroka tomu. Kao prvo, razred OVERLAP najmanje je brojan razred što znači da je modelu predočeno najmanje primjera za učenje koji pripadaju tome razredu. Nadalje, analiza pogrešaka klasifikacije pokazuje da je najveći broj pogrešaka zabilježen između razreda OVERLAP i BEFORE te OVERLAP i AFTER, što upućuje na zaključak da je vremensko preklapanje (razred OVERLAP) teže prepoznati iz teksta od vremenske podudarnosti (razred EQUAL) ili vremenskog prethođenja (razredi BEFORE i AFTER). Konačno, razred OVERLAP obuhvaća više Allenovih vremenskih odnosa (OVERLAPS, FINISHES, STARTS, DURING), a kako postoji mogućnost da se svaki od tih vremenkih odnosa drugačije manifestira u tekstu, postoji opasnost da je razred OVERLAP, kako je u ovom istraživanju definiran, umjetan koncept. I dok je prva dva potencijalna uzroka lošije klasifikacije razreda OVERLAP moguće ublažiti izgradnjom ujednačenijeg skupa u kojem bi bili označeni samo najistaknutiji primjeri razreda OVERLAP, treća opaska zahtjeva mnogo podrobniju i dugotrajniju analizu (npr. izgradnju zbirke dokumenata označene s primjerima svih Allenovih vremenskih odnosa između događaja uz izgradnju klasifikatora nad takvom zbirkom).

Iako je po apsolutnom iznosu mjere F_1 uspješnost klasifikacije vremenskih odnosa sumjerljiva uspješnosti modela kampanje TempEval-2 (Verhagen i dr., 2010), takva usporedba nije ispravna budući da predstavljeni klasifikacijski modeli GRAF-VREMODN-IDENT i GRAF-VREMODN-KLAS uzimaju u obzir puno veći broj parova događaja (tj. nema ograničenja na primjere koji zadovoljavaju određene sintaktičke uzorke), a model GRAF-VREMODN-KLAS također uzima u obzir i jedan vremenski odnos više (razred EQUAL). Kako bi model za klasifikaciju vremenskih odnosa bilo moguće izravno usporediti s najuspješnijim modelima kampanje TempEval-2, združeni su razredi OVERLAP i EQUAL te su ostavljeni samo oni parovi događaja koji zadovoljavaju kriterije zadataka E i F evaluacijske kampanje TempEval-2 (v. odjeljak 3.2.2), tj. samo parovi glavnih događaja susjednih rečenica i parovi događaja iz iste rečenice u kojima sidro jednog događaja sintaktički upravlja sidrom drugog događaja. Rezultati modela za klasifikaciju događaja uz opisane restrikcije (model GRAF-VREMODN-TIMEBANK) dani su u tablici 6.7 zajedno s rezultatima dvaju najuspješnijih modela iz evaluacijske kampanje TempEval-2 (Llorens i dr., 2010; UzZaman i Allen, 2010). Rezultati potvrđuju pretpostavku da je razvijeni model za klasifikaciju vrste vremenskog odnosa između događaja na danom zadatku jednako uspješan ili uspješniji (razlike u uspješnosti nisu statistički značajne; testirano dvostranim t-testom) od najuspješnijih poznatih modela.

Tablica 6.7: Uspješnost modela za klasifikaciju vremenskih odnosa na zadatcima E i F evaluacijske kampanje TempEval-2

Model	OVERLAP	BEFORE	AFTER	F_1
UzZaman i Allen (2010) (TempEval-2 Task E)	–	–	–	58,0
Llorens i dr. (2010) (TempEval-2 Task E)	–	–	–	55,0
UzZaman i Allen (2010) (TempEval-2 Task F)	–	–	–	60,0
Llorens i dr. (2010) (TempEval-2 Task F)	–	–	–	60,0
GRAF-VREMODN-TIMEBANK	78,9	45,1	56,6	**60,2**

6.4 Razrješavanje koreferencije događaja

Bridovi grafova događaja koji predstavljaju odnos koreferencije između spominjanja događaja uspostavljaju se na temelju modela za prepoznavanje koreferentnih spominjanja događaja unutar istog dokumenta (engl. *within-document event coreference*). Nadalje, kako bi bilo moguće uspoređivati grafove događaja jezgrenim funkcijama (v. poglavlje 4.2), potreban je model za prepoznavanje koreferentnih spominjanja događaja koja se nalaze u različitim dokumentima (engl. *cross-document event coreference*). U okviru doktorske disertacije razvijen je univerzalni model nadziranog strojnog učenja koji na temelju istog skupa značajki prepoznaje koreferentna spominjanja događaja unutar istog dokumenta, ali i koreferentna spominjanja događaja koja dolaze iz različitih dokumenata. Model opisan u nastavku objavljen je u znanstvenom radu (Glavaš i Šnajder, 2013b).

6.4.1 Model i značajke

Model se temelji na značajkama koje uspoređuju odgovarajuće dijelove spominjanja događaja – sidra događaja te argumente događaja s podudarnim semantičkim ulogama. Sve su značajke oblikovane na način da indiciraju visoku sličnost ako su spominjanja događaja koreferentna. Značajke su podijeljenje u tri grupe: (1) značajke koje uspoređuju vreće riječi spominjanja događaja, (2) značajke koje uspoređuju imenovane entitete koji su argumenti događaja te (3) značajke koje uspoređuju sidra događaja te argumente s podudarnim semantičkim ulogama.

Usporedba vreća riječi

Vreća riječi spominjanja događaja skup je koji sadrži sve leme spominjanja događaja – dakle, lemu sidra događaja te sve leme svih argumenata događaja. Promatrajući spominjanja događaja kao vreće riječi, za usporedbu njihove semantičke podudarnosti moguće je koristiti neke od

značajki koje su se pokazale uspješnima na zadatku određivanja semantičke sličnosti kratkih tekstova (Agirre i dr., 2012; Šarić i dr., 2012):

- *Pohlepna usporedba sličnosti vreća riječi* (engl. *greedy bag-of-words overlap*) je značajka koja računa semantičku sličnost vreća riječi dvaju spominjanja događaja (f_1^{KF}). Računanje ove značajke odgovara problemu težinskog uparivanja vrhova bipartitnog grafa (engl. *weighted bipartite graph matching problem*), koji rješavamo iterativno na sljedeći način: u svakoj iteraciji odabiremo par lema (l_1, l_2) (pri čemu je l_1 lema iz vreće riječi spominjanja prvog događaja, a l_2 lema iz vreće riječi spominjanja drugog događaja) koje imaju najveću semantičku sličnost. Semantičku sličnost lema mjerimo algoritmom Wua i Palmer na leksičko-semantičkoj mreži WordNet. Neka je P skup svih parova lema nastalih ovim pohlepnim postupkom uparivanja. Vrijednost značajke za par spominjanja događaja e_1 i e_2 računamo kao težinski zbroj sličnosti svih parova lema iz P koji je normaliziran kardinalnošću veće od vreća riječi:

$$grOv(e_1, e_2) = \frac{\sum_{(l_1,l_2)\in P} w(l_1, l_2) \cdot wupalmer(l_1, l_2)}{\max(|bow(e_1)|, |bow(e_2)|)},$$

  gdje je $w(l_1, l_2)$ težinska funkcija koja rezultira većim od *sadržaja informacije* dviju lema. *Sadržaj informacije* (Resnik, 1999) omogućava da se naglasak stavi na informativnije leme, a za pojedinu lemu l računa se na temelju relativne frekvencije pojavljivanja te leme u nekom velikom korpusu C:

$$ic(l) = -\ln P(l) = \ln \frac{\sum_{l' \in C} freq(l')}{freq(l)}. \tag{6.2}$$

  Za računanje informativnosti lema korištena je velika zbirka Google Book Ngrams (Michel i dr., 2011). Slične strategije uparivanja temeljene na semantičkoj sličnosti lema korištene su pri određivanju semantičke sličnosti kratkih tekstova (Šarić i dr., 2012) te za vrednovanje sustava za strojno prevođenje (Lavie i Denkowski, 2009);

- *Težinski presjek lema* (engl. *weighted lemma overlap*) računa se kao zbroj sadržaja informacije lema koje se nalaze u vrećama riječi obaju spominjanja događaja (f_2^{KF}). Pokrivenost sadržaja informacije događaja e_1 lemama događaja e_2 računa se kao:

$$wlc(e_1, e_2) = \frac{\sum_{l \in bow(e_1) \cap bow(e_1)} ic(l)}{\sum_{l' \in bow(e_2)} ic(l')}. \tag{6.3}$$

  Konačna vrijednost značajke težinskog prosjeka lema računa se kao harmonijska sredina pokrivenosti sadržaja prvog spominjanja događaja (e_1) lemama drugog spominjanja događaja (e_2) i pokrivenosti drugog spominjanja događaja lemama prvog spominjanja

događaja:

$$wlo(e_1,e_2) = 2 \cdot \frac{wlc(e_1,e_2) \cdot wlc(e_2,e_1)}{wlc(e_1,e_2) + wlc(e_2,e_1)}. \tag{6.4}$$

- *Presjek n-grama sadržajnih riječi* (engl. *content ngram overlap*) broji ngrame sadržajnih riječi (imenice, glagoli, pridjevi i prilozi) koji se nalaze u vrećama riječi obaju spominjanja događaja. Presjek računamo za unigrame (f_3^{KF}), bigrame (f_4^{KF}) i trigrame (f_5^{KF}). Neka je $bow(e,n)$ skup svih ngrama duljine n dobivenih iz vreće riječi spominjanja događaja e. Pokrivenost jednog spominjanja događaja n-gramima sadržajnih riječi duljine n drugog događaja računa se na sljedeći način:

$$cngc(e_1,e_2,n) = \frac{|bow(e_1,n) \cap bow(e_2,n)|}{|bow(e_1,n)|}.$$

  Konačna vrijednost značajke presjeka ngrama sadržajnih riječi računa se kao harmonijska sredina pokrivenosti prvog spominjanja događaja (e_1) ngramima (duljine n) sadržajnih riječi drugog spominjanja događaja (e_2) i pokrivenosti drugog spominjanja događaja ngramima sadržajnih riječi prvog spominjanja događaja:

$$cngo(e_1,e_2,n) = 2 \cdot \frac{cngc(e_1,e_2,n) \cdot cngc(e_2,e_1,n)}{cngc(e_1,e_2,n) + cngc(e_2,e_1,n)}. \tag{6.5}$$

Podudarnost imenovanih entiteta

Značajke za mjerenje podudarnosti imenovanih entiteta broje imenovane entitete koji se nalaze kao argumenti obaju spominjanja događaja. Pri tome se podudaranje imenovanih entiteta broji neovisno o podudarnosti semantičkih uloga argumenata koje ti imenovani entiteti popunjavaju:

- *Podudarnost riječi pisanih velikim početnim slovom* (engl. *capital letter overlap*) predstavlja broj riječi pisanih velikih početnim slovom (kao rudimentarni način prepoznavanja imenovanih entiteta) koje pronalazimo u okviru obaju spominjanja događaja (f_6^{KF}). Broj podudarnih riječi pisanih velikim početnim slovom normalizira se redom s ukupnim brojem riječi pisanih velikim početnim slovom u prvom odnosno drugom spominjanju događaja, a konačna vrijednost značajke računa se kao harmonijska sredina tih dviju normaliziranih vrijednosti;
- *Podudarnost imenovanih entiteta* (engl. *named entity overlap*) mjerimo usporedbama po vrstama imenovanih entiteta: *Person* (f_7^{KF}), *Organization* (f_8^{KF}), *Location* (f_9^{KF}) te po vremenskim izrazima (f_{10}^{KF}), i to kao broj podudarnih imenovanih entiteta određene vrste (npr. *Person*), normaliziran s ukupnim brojem imenovanih entiteta te vrste u prvom odnosno drugom spominjanju događaja. Konačna vrijednost svake od ovih značajki jest harmonijska sredina normaliziranih preklapanja za odgovarajuću vrstu imenovanog entiteta. Za svaku od četiri vrste imenovanih entiteta model koristi i po jednu binarnu značajku

(f_{11}^{KF} – f_{14}^{KF}) koja označava ima li barem jedno od spominjanja događaja koja uspoređujemo barem jedan imenovani entitet odgovarajuće vrste (tj. ima li usporedba imenovanih entiteta smisla). Tim značajkama razlikujemo slučajeve kada niti jedno od spominjanja događaja nema niti jedan imenovani entitet nekog tipa (što ne znači da spominjanja događaja nisu koreferentna) od slučajeva kad spominjanja događaja imaju entitete nekog tipa, ali ti entiteti nisu podudarni (što znači da spominjanja događaja vjerojatno nisu koreferentna);

- *Podudarnost brojeva* (engl. *numbers overlap*) mjeri broj numeričkih izraza koji postoje u oba spominjanja događaja (f_{15}^{KF}). Numerički izrazi pojavljuju se kao dio datuma, vremenskih izraza, novčanih iznosa ili postotaka, a njihova nepodudarnost između spominjanja događaja u pravilu sugerira da spominjanja događaja nisu koreferentna. Za primjere "*Australians won three gold medal*" i "*Australians won seven gold medal*" zaključujemo da se ne radi o istom natjecanju budući su Australci osvojili različit broj medalja. Ukupan broj numeričkih izraza u oba spominjanja događaja koristi se kao dodatna značajka koja služi tome da razlikujemo slučajeve kad niti jedno spominjanje događaja ne sadrži numeričke izraze od slučajeva u kojima ih sadrže, ali ti izrazi nisu podudarni.

Podudarnost sidara i argumenata događaja

Kako sidra događaja određuju osnovno značenje samog događaja, koreferentna spominjanja događaja imaju semantički bliska sidra. Slično vrijedi i za argumente događaja – koreferentna spominjanja događaja trebala bi imati podudarne argumente. Drugim riječima, ako se dva spominjanja događaja odnose na isti događaj stvarnog svijeta, onda bi argument s ulogom AGENT, primjerice, trebao biti isti za oba spominjanja događaja. Značajke temeljene na vrećama riječi i podudarnosti imenovanih entiteta indiciraju semantičku sličnost dvaju spominjanja događaja, ali ne nužno i koreferenciju. Razmotrimo sljedeća dva spominjanja događaja:

(23) *The dog bit a boy.*

(24) *The boy bit a dog.*

Gledano kroz značajke usporedbe vreća riječi te značajke podudarnosti imenovanih entiteta, ova su dva spominjanja događaja potpuno jednaka, iako ne opisuju isti događaj stvarnog svijeta. Stoga su potrebne značajke koje uspoređuju argumente događaja s istim semantičkim ulogama kako bi model mogao utvrditi da se u ovakvim slučajevima ne radi o koreferenciji događaja. U prethodnom primjeru argument s ulogom AGENT u prvom spominjanju događaja je "*The dog*", a u drugome spominjaju događaja to je "*The boy*". Nepostojanje koreferencije moguće je zaključiti na temelju semantičke različitosti koncepata "*dog*" i "*boy*". Za usporedbu sidara i argumenata događaja koristimo sljedeće značajke:

- Semantička sličnost sidara događaja izmjerena algoritmom Wua i Palmer na leksičko-semantičkoj mreži WordNet (f_{16}^{KF});
- Semantička sličnost između glavnih riječi argumenata događaja s istom semantičkom ulogom za sve četiri semantičke uloge argumenata – AGENT (f_{17}^{KF}), TARGET (f_{18}^{KF}), TIME (f_{19}^{KF}), LOCATION (f_{20}^{KF}). Kao i semantičku sličnost sidara, sličnost između glavnih riječi argumenata mjerimo algoritmom Wua i Palmer na leksičko-semantičkoj mreži WordNet;
- Semantička sličnost sintaktičkih odsječaka argumenata događaja s istom semantičkom ulogom za sve četiri semantičke uloge argumenata – AGENT (f_{21}^{KF}), TARGET (f_{22}^{KF}), TIME (f_{23}^{KF}), LOCATION (f_{24}^{KF}). Sličnost sintaktičkih odsječaka događaja računa se kao prosječna semantička sličnost svih parova lema između tih odsječaka (pri čemu se sličnost dviju lema također računa algoritmom Wua i Palmer na mreži WordNet);
- Za svaku od četiri semantičke uloge argumenata koristi se i binarna značajka koja označava ima li usporedba argumenata s tom ulogom smisla (f_{25}^{KF} – f_{28}^{KF}). Usporedba argumenata neke semantičke uloge nema smisla kada niti jedno od spominjanja događaja nema niti jedan argument te semantičke uloge. Korištenjem ovih značajki model je u stanju razlikovati parove spominjanja događaja gdje niti jedno spominjanje nema argument neke semantičke uloge (što ne mora značiti da spominjanja događaja nisu koreferentna) od parova spominjanja događaja koja imaju argumente određene semantičke uloge, ali ti argumenti nisu podudarni (što znači da spominjanja događaja vjerojatno nisu koreferentna);
- Razlika u ukupnom broju argumenata između spominjanja događaja (f_{29}^{KF});
- Ukupan broj podudarnih argumenata između spominjanja događaja (f_{30}^{KF});
- Broj nepodudarnih argumenata za svako od dva spominjanja događaja (f_{31}^{KF} i f_{32}^{KF});
- Konkatenacija oznaka vrsta riječi sidara dvaju spominjanja događaja (npr. VV, VN, NN, VA) (f_{33}^{KF}).

Model za prepoznavanje koreferentnih spominjanja unutar dokumenta, potreban za izgradnju dijela bridova u grafovima događaja, dodatno koristi binarne značajke koje označavaju nalaze li se spominjanja događaja u istoj rečenici (f_{34}^{KF}) te nalaze li se spominjanja događaja u susjednim rečenicama (f_{35}^{KF}).

Kao klasifikacijski model za prepoznavanje parova koreferentnih spominjanja događaja koristi se stroj potpornih vektora (SVM) s radijalnim baznim funkcijama (engl. *radial-basis function kernel*, RBF). Odabir modela motiviran je činjenicom da je, uslijed većinski binarnih i numeričkih značajki (jedino je značajka f_{33}^{KF} kategorička značajka), ukupan broj značajki bitno manji od broja primjera u skupu za učenje. U eksperimentima je korištena implementacija algoritma SVM iz bibliotoeke LibSVM (Chang i Lin, 2011).

Tablica 6.8: Uspješnost modela za prepoznavanje koreferentnih spominjanja događaja

	Unutar dokumenta			Između dokumenata		
	P	R	F1	P	R	F1
Temeljna metoda	93,1	28,3	43,4	90,7	28,6	43,3
Model Bejana i Harabagiu (2010)	83,0	34,4	48,6	37,5	85,6	52,1
GRAF-KOREF	81,6	41,8	**55,3**	79,0	57,0	**66,2**

6.4.2 Vrednovanje modela

Model za prepoznavanje koreferentnih spominjanja događaja (GRAF-KOREF) vrednujemo na zbirci EventCorefBank (Bejan i Harabagiu, 2008), ručno označenoj lancima koreferentnih spominjanja događaja, koju smo podijelili na skupove za učenje i ispitivanje u omjeru 80%–20%. Za usporedbu koristimo dvije temeljne metode. Prva je standardna temeljna metoda koja par spominjanja događaja smatra koreferentnima ukoliko su im leme sidara jednake ili je jedna lema nominalizacija druge (Ahn, 2006; Bejan i Harabagiu, 2010). Druga metoda za usporedbu jest neparametarski Bayesovski model za razrješavanje koreferencije događaja koji spada u nenadzirane modele strojnog učenja (Bejan i Harabagiu, 2010). U tablici 6.8 prikazana je uspješnost tri modela (model GRAF-KOREF, temeljna metoda i model nenandziranog strojnog učenja Bejana i Harabagiu (2010)) na skupu za ispitivanje zbirke EventCorefBank, i to zasebno za parove događaja unutar istog dokumenta (stupci 2–4) te posebno za parove događaja iz različitih dokumenata (stupci 5–7). Model GRAF-KOREF temeljen na nadziranom strojnom učenju bitno je uspješniji i od temeljne metode i od naprednog modela nenadziranog strojnog učenja (Bejan i Harabagiu, 2010). Uspješnost modela značajno je veća na parovima spominjanja događaja iz različitih dokumenata, što sugerira da je razrješavanje koreferencije unutar istog dokumenta zahtjevniji problem. Ovo je u prvom redu posljedica činjenice da unutar istog dokumenta naredna spominjanja događaja koji je prethodno već spomenut u tekstu često nemaju eksplicitno navedene argumente. Razmotrimo spominjanja događaja u sljedećem primjeru:

(25) *Jackson was <u>arrested</u> on DUI charge early yesterday. General Manager A.J. Smith issued the following statement on the <u>arrest</u>...*

Prvo spominjanje događaja "*arrested*" ima argumente "*Jackson*" s ulogom TARGET i "*early yesterday*" s ulogom TIME dok su isti argumenti implicitni za njegovo koreferentno spominjanje "*arrest*". Budući da se predstavljeni model prvenstveno temelji na usporedbi sidara i argumenata spominjanja događaja, očekivano više griješi na primjerima gdje jedno od spominjanja događaja ima implicitne argumente. Drugi čest uzrok lažnih negativnih primjera jesu pogreške modela za razrješavanje koreferencije entiteta (Lee i dr., 2011). Na primjer, u odsječku teksta

(26) *Matt Smith has been cast as the next incarnation of the Doctor. Even the skeptics agreed that the right guy was chosen...*

spominjanja događaja "*cast*" i "*chosen*" kao argumente s ulogom TARGET imaju koreferentna spominjanja entiteta "*Math Smith*" i "*the right guy*". Model za razrješavanje koreferencije entiteta, međutim, ne prepoznaje ta dva izraza kao spominjanja istog entiteta, pa posljedično ni model za razrješavanje koreferencije događaja ne prepoznaje da spominjanja događaja "*cast*" i "*chosen*" imaju podudarne argumente s ulogom TARGET.

Poglavlje 7

Vrednovanje grafova događaja

Vrednovanje uspješnosti pojedinačnih modela od kojih se sastoji postupak izgradnje grafova događaja (npr. model za crpljenje sidara činjeničnih spominjanja događaja) provodi se uobičajenim mjerama poput preciznosti, odziva i mjere F_1 (van Rijsbergen, 1979). Uspješnost pojedinačnih modela, međutim, ne odražava ukupnu kvalitetu grafova događaja budući da se greške ranijih modela u slijedu propagiraju te uzrokuju pogreške modela koji ovise o njihovom izlazu. Na primjer, ako model za crpljenje sidara događaja preskoči jedno sidro događaja, tada model za crpljenje vremenskih odnosa između događaja neće razmatrati vremenske odnose između tog događaja i drugih događaja.

Na ovom su mjestu predstavljene dvije nove mjere za vrednovanje uspješnosti cjelokupnog postupka izgradnje grafova događaja ili, ekvivalentno, za mjerenje ukupne kvalitete izgrađenih grafova događaja. Predložene mjere temelje se na podudaranju informacija između automatski izgrađenih grafova događaja i referentnih grafova događaja koje su izgradili ljudi (v. poglavlje 5.3). Mjere uzimaju u obzir sve informacije sadržane u grafovima događaja (sidra, argumente, vremenske odnose i koreferenciju), a pokazuje se da imaju poželjna svojstva kakva imaju i standardne mjere preciznosti i odziva.

7.1 Mjere temeljene na tenzorskom umnošku grafova

Za obje mjere koristimo tenzorski umnožak grafova (v. poglavlje 4.2.3) kao sredstvo za određivanje preklapanja između automatski izgrađenog grafa događaja G_C i referentnog grafa G_R. Korištenje tenzorskog umnoška kao osnove za računanje mjera vrednovanja ima smisla budući da će graf koji nastaje tenzorskim umnoškom grafova G_C i G_R biti to veći što je automatski izgrađeni graf G_C sličniji referentnom grafu događaja G_R. Međutim, budući da u ovom slučaju ne mjerimo sličnost grafova događaja izgrađenih iz različitih dokumenata (što je u odjeljku 4.2 bilo predstavljeno kao osnovna motivacija za upotrebu jezgrenih funkcija nad grafovima), već vrednujemo točnost crpljenja informacija o događajima usporedbom automatski izgrađenog

grafa s odgovarajućim referentnim grafom, kao predikat za podudarnost oznaka vrhova grafova događaja ne koristimo koreferenciju spominjanja događaja već potpuno podudaranje sidara i svih argumenata događaja. Drugim riječima, predikat $\delta(v_r, v_c)$, koji uspoređuje vrhove $v_r \in G_R$ i $v_c \in G_C$, bit će zadovoljen ako i samo ako vrijedi sljedeće:

1. Sidro spominjanja događaja $m_r(v_r)$ i sidro spominjanja događaja $m_c(v_c)$ predstavljaju istu pojavnicu u tekstu;
2. Spominjanja događaja $m_r(v_r)$ i $m_c(v_c)$ pripadaju istom semantičkom razredu događaja (npr. PERCEPTION);
3. Skupovi argumenata spominjanja događaja $m_r(v_r)$ i spominjanja događaja $m_c(v_c)$ moraju biti identični, što znači da automatski ekstrahirano spominjanje događaja $m_c(v_c)$ ne smije imati niti jedan argument viška niti manjka u odnosu na ručno označeno spominjanje događaja $m_r(v_r)$. Svi argumenti referentnog i ekstrahiranog spominjanja događaja dodatno se moraju poklapati i po semantičkim ulogama.

Dvije predložene mjere uspješnosti izgradnje cjelokupnih grafova događaja jesu mjera *relativne veličina preklapanja* (engl. *relative overlap size*, ROS) i mjera podudaranja *najveće povezane komponente* (engl. *largest connected component*, LCC). Mjera relativne veličine preklapanja mjeri sposobnost sustava da prepozna sve informacije vezane za događaje, bez obzira pripadaju li te informacije glavnoj priči (engl. *main narrative*) teksta ili nekoj od sporednih priča (engl. *side narrative*). S druge strane, mjera podudaranja najveće povezane komponente mjeri sposobnost sustava da prepozna informacije o događajima vezanima samo za glavnu priču teksta.

Mjere za vrednovanje ukupne kakvoće izgrađenih grafova događaja moraju zadovoljavati dva važna kriterija:

1. Vrijednosti koje mjere pridjeljuju grafovima događaja doista odražavaju točnost ekstrahiranih informacija koje se nalaze u automatski izgrađenim grafovima događaja. Jednostavnije rečeno, mjere moraju pridjeljivati veće vrijednosti što su grafovi događaja točniji;
2. Rezultate mjera potrebno je moći lako tumačiti odnosno povezati sa standardnih mjerama preciznosti i odziva.

Obje predložene mjere zadovoljavaju oba prethodno navedena kriterija koja ih čine prikladnima za vrednovanje automatski izgrađenih grafova događaja. Kako bi zadovoljile navedene kriterije, mjere su osmišljene tako da kažnjavaju izostanak informacija iz referentnog grafa G_R u automatski izgrađenom grafu G_C, kao i suvišne informacije u automatski izgrađenom grafu G_C koje ne postoje u referentnom grafu G_R.

7.1.1 Relativna veličina preklapanja (mjera ROS)

Mjera relativne veličine preklapanja (ROS) uspoređuje veličinu grafa koji nastaje tenzorskim umnoškom automatski izgrađenog grafa G_C i referentnog grafa G_R s veličinama tih grafova.

Veličinu grafa G računamo kao zbroj broja njegovih vrhova i broja njegovih bridova, $|V(G)| + |E(G)|$. Mjeru ROS računamo iz komponenti koje odgovaraju preciznosti i odzivu:

$$ROS_P(G_C, G_R) = \frac{|V(G_C \times G_R)| + |E(G_C \times G_R)|}{|V(G_C)| + |E(G_C)|}, \tag{7.1}$$

$$ROS_R(G_C, G_R) = \frac{|V(G_C \times G_R)| + |E(G_C \times G_R)|}{|V(G_R)| + |E(G_R)|}, \tag{7.2}$$

gdje je $G_C \times G_R$ tenzorski umnožak grafova G_C i G_R. Vrijednost mjere ROS računamo kao harmonijsku sredinu komponenti preciznosti i odziva:

$$ROS(G_C, G_R) = 2 \cdot \frac{ROS_P(G_C, G_R) \cdot ROS_R(G_C, G_R)}{ROS_P(G_C, G_R) + ROS_R(G_C, G_R)}. \tag{7.3}$$

Mjera ROS odražava ukupnu količinu informacije koju automatski izgrađeni graf događaja G_C i referentni graf događaja G_R dijele. Međutim, ta mjera ni na koji način ne razmatra niti ocjenjuje strukturu preklapanja između izgrađenog i referentnog grafa događaja. Istom vrijednošću mjere ROS može, primjerice, biti ocijenjen automatski izgrađen graf događaja koji se s referentnim grafom preklapa u jednoj velikoj povezanoj komponenti (cjelovito podudaranje) kao i automatski izgrađen graf koji s referentnim grafom dijeli više manjih međusobno nepovezanih komponenti (fragmentirano podudaranje).

7.1.2 Podudaranje najveće povezane komponente (mjera LCC)

Mjera podudaranja najveće povezane komponente (LCC), za razliku od mjere ROS, donekle uzima u obzir i strukturu preklapanja između automatski izgrađenih i referentnih grafova događaja. Pri izračunu vrijednosti mjere LCC prvo se računa tenzorski umnožak između najveće slabo povezane komponente (engl. *largest weakly connected component*) prvog grafa događaja (npr. G_R kod izračuna komponente odziva mjere LCC) i drugog grafa događaja (npr. G_R kod izračuna komponente odziva mjere LCC), a potom se uspoređuje veličina najveće slabo povezane komponente tako dobivenog umnoška s veličinom najveće slabo povezane komponente prvog grafa (npr. G_R kod izračuna komponente odziva mjere LCC). Neka je $lcc(G)$ najveća slabo povezana komponenta grafa G. Mjeru LCC računamo iz komponenti koje odgovaraju preciznosti i odzivu:

$$LCC_P(G_C, G_R) = \frac{\left|V\left(lcc(G_R \times lcc(G_C))\right)\right|}{|V(lcc(G_C))|}, \tag{7.4}$$

$$LCC_R(G_C, G_R) = \frac{\left|V\left(lcc(G_C \times lcc(G_R))\right)\right|}{|V(lcc(G_R))|}. \tag{7.5}$$

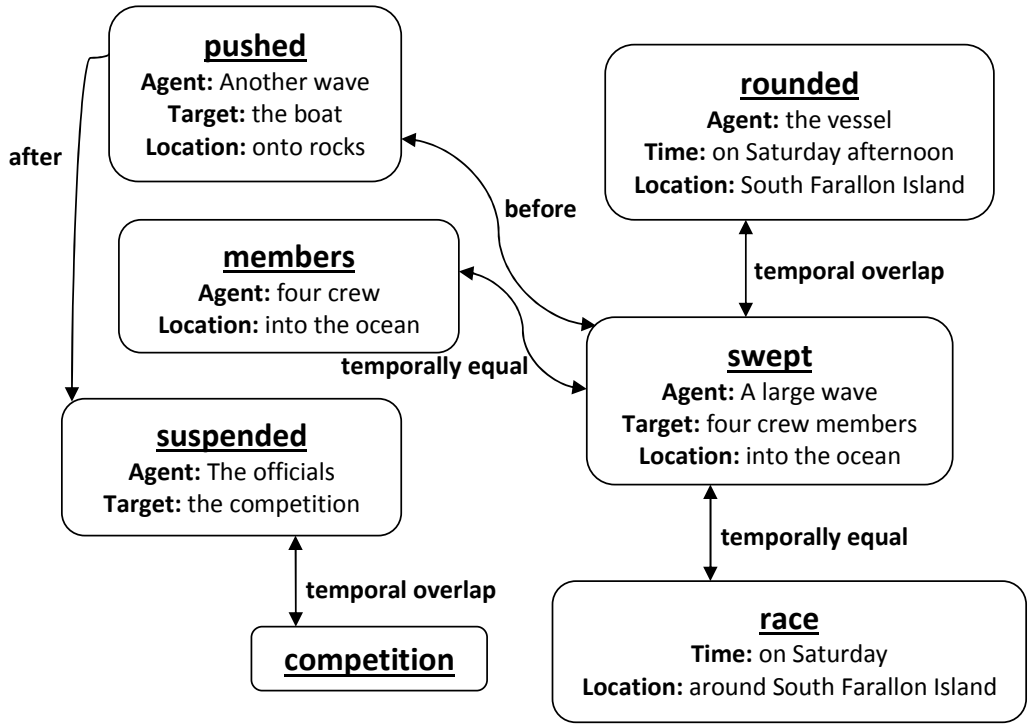
Vrijednost mjere LCC računamo kao harmonijsku sredinu komponenti preciznosti i odziva:

$$LCC(G_C, G_R) = 2 \cdot \frac{LCC_P(G_C, G_R) \cdot LCC_R(G_C, G_R)}{LCC_P(G_C, G_R) + LCC_R(G_C, G_R)}. \tag{7.6}$$

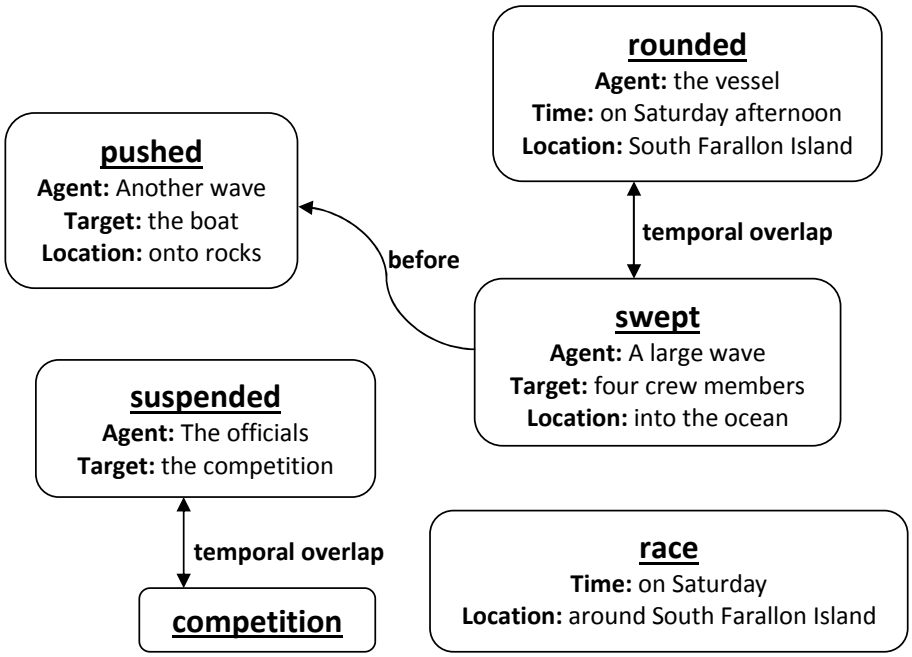
Mjera LCC temelji se na pretpostavci da su događaji koji pripadaju istoj priči predstavljeni na informacijski koherentan način te da će sukladno tome biti predstavljeni povezanom komponentom grafa događaja. Tada će glavna priča teksta (npr. novinskog članka) odgovarati najvećoj povezanoj komponenti grafa događaja, a sporedne priče manjim povezanim komponentama koje u pravilu neće biti spojene s komponentom koja odgovara glavnoj priči jer će zbog udaljenosti u tekstu biti nemoguće odrediti vremenske odnose između sidara događaja koji pripadaju različitim pričama. Pretpostavku da je povezana komponenta koja odgovara glavnoj priči u tekstu značajno veća od ostalih povezanih komponenti (koje odgovaraju sporednim pričama) empirijski potvrđuju ručno označeni grafovi događaja, kao i automatski izgrađeni gafovi događaja za 105 novinskih članaka iz zbirke EVEXTRA (v. odjeljak 5.3), gdje prosječni omjer veličine najveće povezane komponente (po broju vrhova) u odnosu na ukupnu veličinu grafa iznosi 0,83. Uspoređujući preklapanje između jednog grafa i najveće povezane komponente drugoga, mjera LCC odražava podudarnost glavne novinske priče prepoznate od strane automatskog postupka i glavne priče prepoznate od strane ljudi, zanemarujući pri tome utjecaj sporednih priča u tekstovima. S druge strane, mjera LCC manje je prikladna za grafove događaja izgrađene nad tekstovima koji nemaju jednu glavnu priču već više ravnopravnih priča (npr. novinski članak koji daje pregled bitnih događanja u svijetu u jednoj godini). Graf događaja tada sadrži više povezanih komponenti približno jednake veličine. Tada bi se promatranjem samo najveće povezane komponente pri vrednovanju automatski izgrađenog grafa događaja zapravo zanemarila točnost crpljenja informacija za vrlo velik dio teksta (tj. za sve dijelove teksta koji odgovaraju svim ostalim povezanim komponentama). U slučajevima kada automatski postupak izgradnje grafa događaja rezultira izrazito fragmentiranim grafovima događaja, mjera ROS bolje odražava kakvoću izgrađenih grafova događaja od mjere LCC.

7.1.3 Primjer računanja mjera ROS i LCC

Pogledajmo sad primjer izračuna mjera ROS i LCC za jedan par referentnog grafa događaja i automatski izgrađenog grafa događaja. Neka je referentni graf događaja G_R onaj sa slike 4.1 iz poglavlja 4. Na istome je tekstu automatski izgrađen graf događaja G_C koji je prikazan na slici 7.1. U ovom slučaju, model za prepoznavanje sidara događaja napravio je dvije pogreške – nije prepoznao sidro događaja "*competing*" te je pogrešno prepoznao pojavnicu "*members*" kao sidro događaja. Tenzorski umnožak tih dvaju grafova prikazan je na slici 7.2. Tenzorski umnožak sadrži ukupno šest vrhova koji odgovaraju spominjanjima događaja čije je informacije (sidra i argumente) postupak za crpljenje grafova događaja potpuno točno ekstrahirao. Tenzor-



Slika 7.1: Primjer automatski izgrađenog grafa događaja



Slika 7.2: Tenzorski umnožak referentnog grafa događaja (slika 4.1) i automatski izgrađenog grafa događaja (slika 7.1)

ski umnožak sadrži i tri brida koji odgovaraju točno ekstrahiranim vremenskim odnosima među točno ekstrahiranim spominjanjima događaja. Valja primijetiti kao su i referentni graf događaja G_R i automatski izgrađeni graf G_C slabo povezani grafovi, pa njihove najveće slabo povezane komponente odgovaraju njima samima, tj. $lcc(G_C) = G_C$ i $lcc(G_R) = G_R$. Stoga je tenzorski umnožak koji se računa za komponentu preciznosti mjere LCC jednak tenzorskom umnošku koji se računa za komponentu odziva te mjere, tj. $lcc(G_C) \times G_R = lcc(G_R) \times G_C = G_R \times G_C$. Veličine ovih triju grafova i njihovih najvećih slabo povezanih komponenti jesu:

$$
\begin{aligned}
|V(G_R)| &= 7 & |V(G_C)| &= 7 & |V(G_C \times G_R)| &= 6 \\
|E(G_R)| &= 8 & |E(G_C)| &= 6 & |E(G_C \times G_R)| &= 3 \\
|V(lcc(G_R))| &= 7 & |V(lcc(G_C))| &= 7 & |V(lcc(G_C \times G_R))| &= 3
\end{aligned}
$$

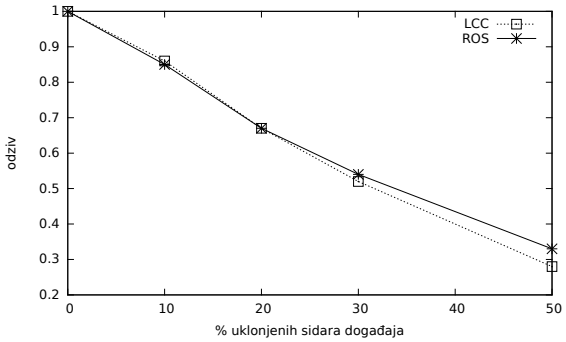
Uvrštavajući ove vrijednosti u jednadžbe (7.1)–(7.6) dobivamo iznose mjera ROS i LCC koje odražavaju kvalitetu automatski izgrađenog grafa G_R:

$$
\begin{aligned}
ROS_P(G_C, G_R) &\approx 0.69 & LCC_P(G_C, G_R) &\approx 0.43 \\
ROS_R(G_C, G_R) &\approx 0.60 & LCC_R(G_C, G_R) &\approx 0.43 \\
ROS(G_C, G_R) &\approx 0.64 & LCC(G_C, G_R) &\approx 0.43
\end{aligned}
$$

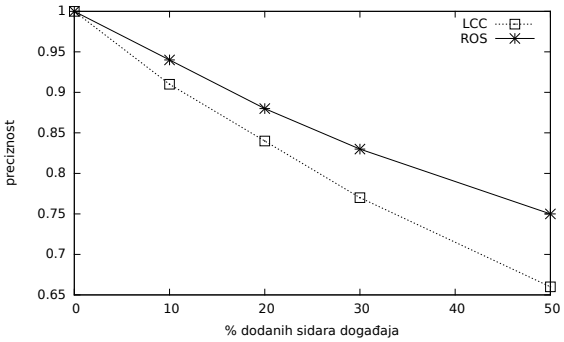
7.2 Provjera mjera ROS i LCC

Da bi bile korisne, mjere ROS i LCC moraju uistinu odražavati kakvoću automatski izgrađenih grafova događaja. Drugim riječima, iznosi mjera ROS i LCC trebaju biti to manji što je više informacija koje postoje u referentnom grafu događaja, a ne postoje u automatski izgrađenom grafu događaja. Iznosi mjera ROS i LCC također trebaju biti to manji što je u automatski izgrađenom grafu događaja više informacija koje ne postoje u referentnom grafu događaja. Obje mjere harmonijske su sredine komponenti koje se mogu povezati s mjerama preciznosti i odziva koje se uobičajeno koriste za vrednovanje u zadatcima obrade prirodnog jezika.

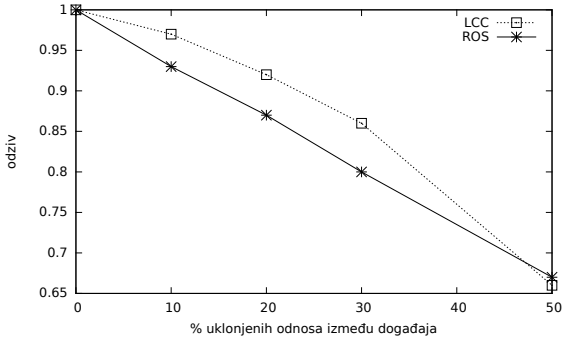
Empirijskom provjerom mjera ROS i LCC utvrđuje se imaju li te mjere doista prethodno navedena poželjna svojstva. Slijedeći ideju iz (UzZaman i Allen, 2011), "automatski izgrađeni" graf događaja G_C dobiva se dodavanjem informacija (sidra, argumenti, vremenski odnosi) u referentni graf G_R, odnosno uklanjanjem informacija iz referentnog grafa G_R. Drugim riječima, automatska izgradnja grafa događaja simulira se manipulacijama nad referentnim grafom događaja. Za tako dobivene grafove G_C računaju se mjere ROS i LCC te se analizira kretanje iznosa tih mjera ovisno o količini suvišnih ili nedostajućih informacija u odnosu na početni referentni graf G_R. Iznosi mjera LCC i ROS računaju se na 105 referentnih grafova ručno označenih na podskupu zbirke EVEXTRA (v. poglavlje 5.3), a prosječni rezultati prikazani su na slici 7.3.



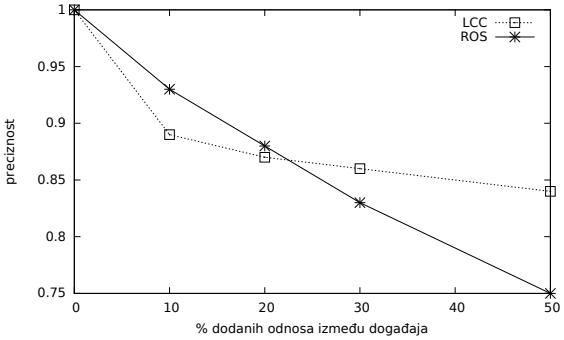
(a) Uklonjeni vrhovi događaja



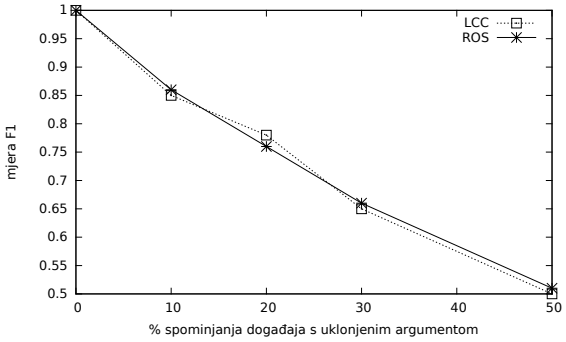
(b) Lažni vrhovi događaja



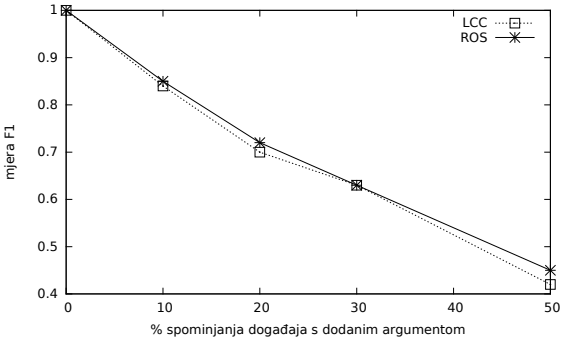
(c) Uklonjeni odnosi između događaja



(d) Lažni odnosi između događaja



(e) Uklonjeni argumenti događaja



(f) Lažni argumenti događaja

Slika 7.3: Provjera mjera ROS i LCC: ispitivanje ponašanja mjera pri uklanjanju stvarnih informacija (sidra (a), odnosi (c) i argumenti (e)) i dodavanju lažnih informacija (sidra (b), odnosi (d) i argumenti (f))

Prvo se uklanjaju vrhovi (i pripadni im bridovi) iz referentnog grafa, kako bi se simulirao slučaj u kojem neka sidra događaja nisu ekstrahirana (lažni negativni primjeri). Pri tome je vidljivo (slika 7.3a) kako komponente odziva za obje mjere opadaju proporcionalno broju vrhova uklonjenih iz referentnog grafa. U sljedećem eksperimentu, dodavanjem vrhova u referentni graf simulira se pojava lažnih pozitivnih primjera pri ekstrakciji sidara događaja, pri čemu je uočljivo da komponente preciznosti za obje mjere opadaju proporcionalno broju vrhova dodanih u referentni graf (slika 7.3b). Sličan pad preciznosti odnosno odziva može se primijetiti i kada se umjesto vrhova u referentni graf dodaju, odnosno iz njega uklanjaju bridovi (slike 7.3c i 7.3d). Pad u preciznosti uzrokovan dodavanjem bridova (tj. odnosa među događajima) za mjeru LCC, međutim, postaje zanemariv nakon dodavanja svega nekoliko bridova. Ovo je posljedica činjenice da su referentni grafovi skoro pa povezani grafovi (s vrlo izraženom najvećom povezanom komponentom), pa ih je dodavanjem svega nekoliko bridova moguće pretvoriti u povezane grafove. Kako dodavanje bridova u povezan graf ne mijenja veličinu najveće povezane komponente grafa (koja je u tom slučaju jednaka cijelom grafu), daljnje dodavanje bridova ne dovodi do daljnjeg pada komponente preciznosti mjere LCC. Ipak, stvarni automatski izgrađeni grafovi puno su fragmentiraniji od ručno označenih grafova, pa neka lažno ekstrahirana vremenska relacija može povezati dvije komponente grafa i stvoriti novu, mnogo veću najveću slabo povezanu komponentu grafa događaja te time značajno narušiti komponentu preciznosti mjere LCC. Konačno, uklanjanje postojećih argumenata događaja iz vrhova referentnog grafa, kao i dodavanje umjetnih argumenata spominjanjima događaja, smanjuje iznose mjera ROS i LCC (slike 7.3e i 7.3f). Budući da se vrh tenzorskog umnoška stvara samo za vrhove automatski izgrađenog događaja koji se s nekim od vrhova referentnog grafa podudaraju u svim argumentima, i dodavanje i uklanjanje argumenata događaja uzrokuju pad, kako komponente preciznosti, tako i komponente odziva za obje mjere. To je posljedica činjenice da se i uklanjanjem i dodavanjem argumenata smanjuje broj vrhova u tenzorskom umnošku grafova G_C i G_R.

7.3 Vrednovanje grafova događaja

Koristeći mjere ROS i LCC vrednujemo ukupnu kvalitetu automatski izgrađenih grafova događaja. Postupak za izgradnju grafova događaja sastoji se od više međusobno zavisnih komponenti. Kod složenih sustava gdje postoji interakcija više pojedinačnih komponenti korisno je vidjeti kako performanse pojedine komponente sustava utječu na performanse cijelog sustava. Stoga su u ovom poglavlju prikazani i rezultati analize prostora za poboljšanja (engl. *headroom analysis*), koja otkriva koliko bi se podigla ukupna kakvoća sustava kada bi neka od komponenti radila savršeno.

Tablica 7.1: Uspješnost postupka automatske izgradnje grafova događaja

Model	LCC	ROS
Temeljna metoda	0.175 ± 0.103	0.184 ± 0.091
GRAF-IZGRADNJA	$\mathbf{0.225} \pm \mathbf{0.098}$	$\mathbf{0.287} \pm \mathbf{0.063}$

7.3.1 Kakvoća automatski izgrađenih grafova događaja

U cilju smislene usporedbe rezultata sustava za automatsku izgradnju grafova događaja, kao temeljnu metodu koristimo kombinaciju temeljnih metoda korištenih kod vrednovanja pojedinačnih modela u izgradnji grafova događaja:

1. Temeljna metoda MEMORIZE za crpljenje sidara događaja te određivanje njihovih semantičkih razreda (v. odjeljak 6.1);
2. Temeljna metoda s osnovnim pravilima za crpljenje argumenata događaja (v. odjeljak 6.2);
3. Temeljna metoda SUSJEDNI-DOGAĐAJI za određivanje parova događaja između kojih postoji vremenski odnos i temeljna metoda GLAGOLSKO-VRIJEME za klasifikaciju vrste vremenskog odnosa među događajima (v. odjeljak 6.3);
4. Temeljna metoda za prepoznavanje koreferentnih spominjanja događaja temeljem podudarnosti lema sidara događaja (v. odjeljak 6.4).

Uspješnost automatskog postupka za izgradnju grafova događaja (GRAF-IZGRADNJA) vrednovana mjerama ROS i LCC zajedno s uspješnošću temeljne metode za izgradnju grafova događaja dana je u tablici 7.1. U tablici su dane srednje vrijednosti i standardne devijacije mjere ROS i LCC za grafove događaja izgrađene za 80 novinskih članaka (v. odjeljak 5.3).

Rezultati prikazani u tablici 7.1 dobiveni su bez primjenjivanja tranzitivnog zatvaranja.[1] Uslijed nemalog broja pogrešno klasificiranih vremenskih odnosa ($F_1 = 55\%$, v. odjeljak 6.3) primjena pravila za tranzitivnost dovodi do stvaranja još većeg broja netočnih vremenskih odnosa te konzistentno rezultira manje kvalitetnim grafovima događaja prema mjerama ROS i LCC.

Postupak automatske izgradnje grafova događaja (GRAF-IZGRADNJA) značajno je uspješniji od temeljne metode izgradnje grafova događaja, pri čemu je razlika u uspješnosti statistički značajna uz $p < 0.05$ (dvostrani t-test) za mjeru LCC te uz $p < 0.01$ za mjeru ROS. Kako mjera LCC za oba postupka za sve grafove događaja daje konzistentno niže vrijednosti od mjere ROS, zaključujemo da mjera LCC daje konzervativniju procjenu kvalitete automatski izgrađenih gra-

[1] Postupak tranzitivnog zatvaranja, na temelju svojstva tranzitivnosti određenih relacija, automatski dodaje nove odnose u graf događaja. Primjerice, na temelju postojećih odnosa *A prije B* i *B prije C*, tranzitivnim bi zatvaranjem u graf događaja bio dodan i odnos *A prije C*.

Tablica 7.2: Rezultati analize prostora za poboljšanja po komponentama

Komponenta	LCC	ΔLCC	ROS	ΔROS
Crpljenje sidara događaja	0,269 ± 0,127	0,044	0,371 ± 0,084	0,084
Crpljenje argumenata događaja	0,245 ± 0,095	0,020	0,345 ± 0,082	0,058
Crpljenje vremenskih odnosa	**0,477 ± 0,170**	**0,252**	**0,437 ± 0,102**	**0,150**
Prepoznavanje koreferencije događaja	0,249 ± 0,106	0,024	0,361 ± 0,074	0,074

fova događaja od mjere ROS.

7.3.2 Analiza prostora za poboljšanja

U okruženju gdje su sve komponente nesavršene, a resursi (ljudski resursi, novac, vrijeme) ograničeni, provođenje analize prostora za poboljšanja za svaku od komponenti sustava otrkiva koju se komponentu najviše isplati unaprjeđivati. Analiza prostora za poboljšanja provedena je za sve četiri glavne komponente sustava za izgradnju grafova događaja: (1) model za crpljenje sidara činjeničnih spominjanja događaja, (2) model za crpljenje argumenata događaja, (3) model za crpljenje vremenskih odnosa među događajima i (4) model za prepoznavanje koreferencije između spominjanja događaja. Pri analizi prostora za poboljšanja za neku komponentu koristimo ljudske oznake za dotičnu komponentu te automatske modele za sve ostale komponente sustava. Primjerice, pri analizi prostora za poboljšanja za komponentu koja crpi vremenske odnose između događaja, pri izgradnji grafova događaja automatski će se crpiti sidra i argumenti događaja te automatski odrediti koreferentna spominjanja događaja, a potom preuzeti ljudske oznake vremenskih odnosa između prethodno ekstrahiranih događaja. U tablici 7.2 prikazani su rezultati analize prostora za poboljšanje za sve komponente sustava za izgradnju grafova događaja, izmjereni mjerama LCC i ROS. Stupci ΔLCC i ΔROS predstavljaju prosječnu razliku u kakvoći grafova događaja u odnosu na potpuno automatizirani postupak. Analiza prostora za poboljšanja pokazuje da se najviše isplati poboljšanje modela za crpljenje vremenskih odnosa između događaja, tj. da poboljšanje točnosti ekstrakcije vremenskih odnosa najviše doprinosi ukupnoj kakvoći izgrađenih grafova događaja. Iako je uspješnost modela za crpljenje vremenskih odnosa po apsolutnom iznosu podjednaka uspješnosti modela za prepoznavanje koreferentnih spominjanja događaja, uspješnost prepoznavanja vremenskih odnosa ima veću težinu jer je broj vremenskih odnosa u referentnim grafovima događaja u prosjeku znatno veći od broja parova koreferentnih spominjanja događaja. Povećanje uspješnosti izgradnje grafova događaja uz referentne vremenske odnose veće je po mjeri LCC (0.477 nasuprot 0.225) nego po mjeri ROS (0.437 nasuprot 0.287). Ovo je posljedica činjenice da samo jedna točno prepoznata vremen-

ska relacija (odnosno brid u grafu događaja) može značajno povećati najveću slabo povezanu komponentu, a time posljedično i iznos mjere LCC.

Dio III

Primjene grafova događaja

Poglavlje 8

Pretraživanje informacija temeljem grafova događaja

Potreba za informacijama jedna je od temeljnih ljudskih potreba. Potreba za učinkovitim i računalno potpomognutim pretraživanjem informacija javlja se uslijed povećanja broja izvora informacija i nemogućnosti jednostavnog pronalaženja informacija koje su od interesa za pojedinca. Pretraživanje informacija (engl. *information retrieval*) područje je koje se bavi učinkovitim pronalaženjem dokumenata koji odgovaraju informacijskoj potrebi korisnika unutar velikih zbirki dokumenata (Manning i dr., 2008). Informacijska potreba korisnika definira se kao svjesna ili nesvjesna potreba da se pronađe i dohvati informacija koja ostvaruje neku želju ili namjeru korisnika (Taylor, 1962). Iako se danas područje pretraživanja informacija najčešće povezuje s internetom i svjetskom mrežom (engl. *world wide web*) kao nedvojbeno najvećim izvorom sadržaja, potreba za učinkovitim pronalaženjem korisnih informacija u obilju informacijskih izvora u mnogim je djelatnostima postojala i prije pojave interneta (npr. zbirke znanstvenih publikacija, pravni propisi, zbirke medicinskih dokumenata).

U suvremenim sustavima za pretraživanje informacija korisnici svoje informacijske potrebe formuliraju u obliku *upita* (engl. *queries*). U većini sustava za pretraživanje informacija danas su pretežno podržani tzv. upiti temeljeni na ključnim riječima (engl. *keyword-based queries*), kod kojih korisnik svoju informacijsku potrebu sažima u svega nekoliko ključnih riječi (npr. "*Bush in Argentina*"). Korisnici, međutim, imaju i informacijske potrebe koje podrazumijevaju strukturu (npr. sintaktičku ili semantičku povezanost između dijelova upita) kakvu nije moguće izraziti samo ključnim riječima, kao što je to slučaj u sljedećem primjeru:

(27) *What are the countries that President Bush <u>visited</u> and in which his <u>visit</u> triggered <u>protests</u>?*

Kauzalnost između posjeta predsjednika Busha nekoj državi i prosvjeda uzrokovanih tim posjetom naprosto nije moguće izraziti upitom temeljenim na ključnim riječima. Nadalje, postoje situacije u kojima nositelj informacijske potrebe kreće od jednog (ili nekoliko) postojećih do-

kumenata te unutar velike zbirke pokušava pronaći dokumente koji su (po nekom unaprijed definiranom kriteriju) najsličniji polaznom dokumentu, kao što je to slučaj u zadatku otkrivanja i praćenja novinskih tema (v. poglavlje 3.5).

Primjer 27 predstavlja informacijsku potrebu određenu događajima. S jedne strane, brojne su informacijske potrebe korisnika koje su određene događajima, dok s druge strane postoji mnoštvo izvora informacija koji u prvom redu opisuju događaje iz stvarnog svijeta (npr. novinski članci, policijski izvještaji, biografije, burzovna izvješća). Stoga se prirodno nameće ideja da bi oslanjanje na informacije o događajima moglo biti vrlo korisno za učinkovito pretraživanje informacija nad zbirkama dokumenata usmjerenih na događaje. Tradicionalni modeli za pretraživanje informacija poput modela vektorskog prostora (engl. *vector space model*, VSM) (Salton i dr., 1975), jezičnog modela (engl. *language model*, LM) (Ponte i Croft, 1998) ili vjerojatnosnog modela (engl. *probabilistic model*, PM) (Robertson i Jones, 1976) dokumente predstavljaju kao vreće riječi, zanemarujući informacije koje proizlaze iz sintaktičke, semantičke i diskursne strukture tih dokumenata.

U ovom poglavlju predstavljen je model za pretraživanje informacija koji se temelji na usporedbi strukturiranih informacija o događajima. Drugim riječima, ovo poglavlje pokazuje kako je korištenjem jezgrenih funkcija za mjerenje sličnosti grafova događaja moguće ostvariti modele za učinkovito pretraživanje informacija. Učinkovitost modela temeljenih na grafovima događaja pokazujemo na dvama različitim zadatcima. Prvo u poglavlju 8.2 grafove događaja primjenjujemo na zadatku pronalaženja dokumenata koji opisuju iste događaje iz stvarnog svijeta (Glavaš i Šnajder, 2013), što je jedan od temeljnih zadataka u području otkrivanja i praćenja tema, a potom u poglavlju 8.3 na zadatku klasičnog pretraživanja informacija, gdje upiti odražavaju informacijske potrebe korisnika koje su usmjerene na događaje (Glavaš i Šnajder, 2013a; Glavaš i Šnajder, 2014b). Odjeljak 8.1 daje pregled srodnih istraživanja u području pretraživanja informacija koja koriste događaje i postupke strukturiranja informacija o događajima.

8.1 Srodna istraživanja

Iako današnji sustavi za pretraživanje informacija i dalje pretežno implementiraju inačice nekog od tradicionalnih modela pretraživanja informacija (Salton i dr., 1975; Ponte i Croft, 1998; Robertson i Jones, 1976), u novije vrijeme predloženi su i modeli koji se oslanjaju na izvjesnu strukturu među riječima upita i dokumenata. Drugim riječima, pojavljuju se proširenja tradicionalnih modela koja koriste podatke o supojavljivanju riječi ili udaljenosti među riječima (Gao i dr., 2004; Metzler i Croft, 2005; Lee i dr., 2006) kao i informacije o sintaktičkim ili semantičkim zavisnostima među riječima (Maisonnasse i dr., 2007; Park i dr., 2011; Shinzato i dr., 2012). Gao i suradnici (2004) proširuju jezični model tako što pretpostavljaju da su i upiti i dokumenti generirani na temelju latentnog planarnog acikličkog grafa, koji modelira jezik kojim

su pisani dokumenti i upiti, a u kojem bridovi označavaju supojavljivanja riječi u dokumentu. Relevantnost pojedinog dokumenta za zadani upit mjeri se kao vjerojatnost da graf izgrađen na temelju dokumenta generira upit, i to ne samo riječi upita već i zavisnosti među njima. Park i suradnici (2011) također proširuju jezični model za pretraživanje informacija, ali prilikom izgradnje modela dokumenta ne koriste supojavljivanja između riječi već ovisnosne sintaktičke relacije. Model dokumenta sadrži ovisnosna sintaktička stabla svih rečenica dokumenta. Relevantnost dokumenta za upit određuje se na temelju preklapanja između sintaktičkih stabala rečenica dokumenta i sintaktičkih ovisnosti između riječi upita, pri čemu se traži približna, a ne potpuna podudarnost grafova (engl. *inexact graph matching*).

Pristupi za pretraživanje informacija koji se temelje na grafovima u načelu se mogu podijeliti u dvije osnovne skupine:

1. Pristupi kod kojih se cijela zbirka dokumenata prikazuje kao graf u kojem svaki vrh predstavlja pojedinačni dokument. Upiti se kod takvih pristupa umeću u graf kao dodatni vrhovi, a pri mjerenju relevantnosti dokumenata za upit koriste se različiti algoritmi nad grafovima (Mihalcea i Tarau, 2004; Kurland i Lee, 2010);
2. Pristupi kod kojih se pojedinačni dokumenti i upiti predstavljaju grafovima koncepata (gdje su koncepti predstavljeni riječima), pri čemu se relevantnost dokumenta za upit određuje usporedbom grafa upita s grafom dokumenta (Montes-y Gómez i dr., 2000; Park i dr., 2011).

Pristup pretraživanju informacija temeljen na grafovima događaja, koji će biti opisan u nastavku, pripada potonjoj skupini jer se temelji na usporedbi grafova, uz bitnu razliku da se radi o semantički bogatijim grafovima (vrhovi su događaji, a bridovi vremenski odnosi među događajima) u odnosu na postojeće pristupe u kojima su grafovi određeni na temelju supojavljivanja riječi ili na temelju sintaktičkih odnosa među riječima.

S obzirom na količinu tekstnih izvora koji su usmjereni na događaje, nezanemarive informacijske potrebe određene događajima te izniman napredak u području crpljenja informacija o događajima ostvaren u proteklom desetljeću, broj pristupa pretraživanju informacija usmjerenih na semantiku događaja vrlo je malen. Lin i suradnici (2007) pristupaju pretraživanju informacija uspoređujući strukturu predikata i argumenata (engl. *predicate-argument structure*) upita s predikatima i pripadnim argumentima dokumenta. Predikat i argumenti iz upita ne crpe se, međutim, automatski, već korisnik upit treba ručno rastaviti na riječ predikata i argumente odgovarajućih semantičkih uloga (npr. LOKACIJA). Kawahara i suradnici (2013) predložili su sličan pristup koji se temelji na usporedbi predikata i argumenata semantičkih uloga, pri čemu argumente u upitu prepoznaju automatskim postupkom. Nadalje, u tom radu autori demonstriraju kako je pretraživanje temeljeno na usporedbi predikata i njihovih semantičkih argumenata učinkovitije od pretraživanja informacija temeljenog na usporedbi sintaktičkih odnosa među riječima. Oba navedena modela pretraživanja informacija prilagođena su isključivo upitima koji

sadrže jedan predikat te ne uzimaju u obzir upite koji uključuju semantičke (npr. vremenske) odnose između dva ili više događaja.

8.2 Prepoznavanje dokumenata koji opisuju iste događaje

Prepoznavanje dokumenata koji govore o istim događajima važan je zadatak u otkrivanju i praćenju novinskih tema (v. poglavlje 3.5), gdje je cilj otkriti dokumente koji govore o istim događajima iz stvarnog svijeta i pratiti dinamiku tih događaja kroz vrijeme. Kako bi se odredilo opisuju li i u kojoj mjeri dva dokumenta iste događaje iz stvarnog svijeta, potrebno je usporediti informacije o događajima sadržane u tim dokumentima. Budući da grafovi događaja u idealnom slučaju sadrže sve relevantne informacije o svim spominjanjima događaja u dokumentu, kao metodu za mjerenje podudarnosti informacija usmjerenih na događaje u novinskim člancima predlaže se uspoređivanje grafova događaja jezgrenim funkcijama nad grafovima (v. poglavlje 4.2). Učinkovitost predložene metode demonstrirana je na dvama povezanim, ali ipak međusobno komplementarnim zadatcima: (1) na zadatku određivanja parova novinskih članaka koji opisuju iste događaje iz stvarnog svijeta i (2) na zadatku rangiranja parova dokumenata prema sličnosti događaja koje opisuju.

8.2.1 Određivanje novinskih članaka koji opisuju iste događaje

Prvi je zadatak definiran kao binarni klasifikacijski zadatak – za svaki par novinskih članaka potrebno je odlučiti opisuju li iste događaje iz stvarnog svijeta (Glavaš i Šnajder, 2013). Za vrednovanje uspješnosti predložene metode temeljene na grafovima događaja i jezgrenim funkcijama nad grafovima potrebna je zbirka novinskih članaka s označenim parovima dokumenata koji opisuju iste događaje.

Skup podataka

Zbirke dokumenata korištene u kampanjama TDT (Wayne, 2000) u ovom zadatku nisu izravno iskoristive budući da za parove dokumenata ne sadrže informacije o podudarnosti događaja koje dokumenti opisuju. Stoga je za potrebe vrednovanja pristupa temeljenog na grafovima događaja izgrađena zbirka EVKERNELS[1] i to na način opisan u nastavku. Koristeći internetsku uslugu EMM News Brief (Steinberger i dr., 2009), koja grupira novinske objave po tematskoj sličnosti, prikupljeno je deset slučajno odabranih grupa tematski povezanih dokumenata s ukupno 64 novinska članka. Budući da ulsuga EMM NewsBrief članke grupira automatski, bilo je potrebno dodatno ručno pročistiti grupe kako bi se osiguralo da:

[1] Zbirka EVKERNELS dostupna je na internetskoj adresi `http://takelab.fer.hr/evkernels`

Vijest 1
"*Just days before crucial talks with world powers on its disputed nuclear programme, Iran was presenting a defiant face on Wednesday, announcing the halt of oil exports to EU nations and warning the West to drop its "language of force." At the same time, chief nuclear negotiator Saeed Jalili was promising to lay out "new initiatives" at the talks due to take place in Istanbul on Saturday – as long as the nations on the other side of the table employed a "constructive approach."...*"
Vijest 2
"*Iran has halted oil exports to Germany, according to Iranian television, in an apparent effort to strengthen its position ahead of a fresh round of nuclear talks with Western powers in Istanbul on Saturday. It stopped crude exports to Spain and Greece on Tuesday to preempt the EU's embargo on oil imports, which come into force in July. Iran halted oil exports to Germany on Wednesday, a day after it stopped crude exports to Spain and Greece, according to Iran's official Press TV news network....*"
Vijest 3
"*Iran has cut oil exports to Germany one day after halting crude sales to Spain as part of its countersanctions against the European Union (EU), Iranian Press TV reported. Tehran has already stopped oil exports to France, Britain, and Greece and is now considering halting crude sales to Italy. Iran's decision to cut crude exports to six European countries - including the Netherlands, Spain, Italy, France, Greece and Portugal - was made after the EU foreign ministers agreed on January 23 to ban oil imports from Iran and freeze the assets of the country's Central Bank across the EU....*"

Tablica 8.1: Primjer dokumenata koji opisuju iste događaje (skup podataka za zadatak pronalaženja dokumenata koji opisuju iste događaje)

1. Svi parovi novinskih članaka koji pripadaju istoj grupi doista opisuju iste događaje iz stvarnog svijeta;
2. Ne postoji niti jedan par dokumenata koji opisuju iste događaje iz stvarnog svijeta, a nalaze se u različitim grupama.

Skup parova dokumenata koji služe kao zlatni standard (engl. *gold standard*, *ground truth*) izgrađen je uparivanjem svakog dokumenta sa svim drugim dokumentima zbirke. Na taj je način dobiven konačan skup podataka koji se sastoji od 2016 parova novinskih članaka, od čega je 195 pozitivnih parova (parovi dokumenata iste grupe, tj. parovi dokumenata koji opisuju iste događaje) i 1821 negativnih parova (parovi dokumenata iz različitih grupa koji ne opisuju iste događaje). Primjeri odsječaka dokumenata jedne grupe dani su u tablici 8.1.

Modeli

Podudarnost informacija o događajima između dva dokumenta mjeri se na način da se za svaki od dokumenata izgradi graf događaja na način opisan u poglavlju 6, a zatim se određuje podudarnost između tih grafova nekom od jezgrenih funkcija koje su opisane u odjeljku 4.2. Ukoliko je sličnost dvaju grafova događaja određena iznosom jezgrene funkcije veća od nekog praga α, klasifikacijski model donosi odluku da zadani dokumenti opisuju iste događaje.

Korištena su tri različita modela koji se razlikuju po jezgrenoj funkciji koju upotrebljavaju (v. odjeljak 4.2): (1) model s jezgrenom funkcijom temeljenom na tenzorskom umnošku grafova (PGK-TENZORSKI), (2) model s jezgrenom funkcijom temeljenom na konormalnom umnošku grafova (PGK-KONORMALNI) te (3) model s jezgrenom funkcijom težinske dekompozicije grafa (WDK). Kao temeljnu metodu u ovom eksperimentu koristimo model vektorskog prostora (VSM), pri čemu su težine pojedinih riječi izračunate po shemi TF-IDF (Salton i Buckley, 1988). Ovdje je važno naglasiti da internetska usluga EMM NewsBrief pri grupiranju dokumenata koristi algoritam čija je jedna komponenta upravo model vektorskog prostora, stoga imajući na umu da su referentne iznake izvedene na temelju EMM NewsBrief grupa, VSM na ovome zadatku predstavlja izuzetno kompetitivnu temeljnu metodu.

Osim modela koji koriste pojedinačne jezgrene funkcije nad grafovima događaja, isprobani su i modeli koji izlaze pojedinačnih jezgrenih funkcija kombiniraju na način da ih koriste kao značajke za model nadziranog strojnog učenja, zajedno s drugim značajkama koje opisuju veličinu grafova događaja koji se uspoređuju. Dva su razloga za isprobavanje takvih kombiniranih modela. Kao prvo, budući da jezgrene funkcije temeljene na umnošku grafova mjere broj podudarnih šetnji između dva grafa, a jezgrena funkcija težinske dekompozicije broj podudarnih podgrafova, postoji mogućnost da različite jezgrene funkcije mjere različite aspekte sličnosti grafova događaja, pa temeljem te potencijalne komplementarnosti njihovim kombiniranjem možemo dobiti robusniju procjenu sličnosti grafova događaja. Kao drugo, iznosi jezgrenih funkcija izravno ovise o veličini grafova događaja, pa tako usporedbom dugačkih dokumenata koji ne govore o istim događajima ponekad možemo dobiti veće iznose jezgrenih funkcija nego usporedbom kratkih dokumenata koji govore o istim događajima. Stoga je iznose jezgrenih funkcija potrebno na neki način normalizirati veličinama ulaznih grafova. Nije, međutim, očigledno na koji način međusobno kombinirati iznose jezgrenih funkcija niti kako treba izgledati funkcija njihove normalizacije veličinom grafova. Stoga sve iznose jezgrenih funkcija i podatke o veličinama grafova događaja (broj vrhova i bridova) predajemo kao ulazne značajke algoritmu SVM. Model SVM s jezgrenom funkcijom RBF (SVM-GRAF) koristi, dakle, sljedeće značajke:

- Iznos jezgrene funkcije temeljene na tenzorskom umnošku grafova događaja (f_1^{TDT});
- Iznos jezgrene funkcije temeljene na konormalnom umnošku grafova događaja (f_2^{TDT});
- Iznos jezgrene funkcije težinske dekompozicije grafova događaja (f_3^{TDT});
- Broj vrhova prvoga ulaznog grafa (f_4^{TDT});

Tablica 8.2: Uspješnost prepoznavanja parova dokumenata koji opisuju iste događaje

Model	P	R	F_1
PGK-TENZORSKI	89,7	82,3	85,8
PGK-KONORMALNI	89,3	77,8	83,2
WDK	88,6	73,7	80,5
SVM-GRAF	91,1	87,6	89,3
SVM-GRAF+VSM	**93,8**	**96,2**	**95,0**
VSM (temeljna metoda)	90,9	82,9	86,7

- Broj vrhova drugoga ulaznog grafa (f_5^{TDT});
- Broj bridova prvoga ulaznog grafa (f_6^{TDT});
- Broj bridova drugoga ulaznog grafa (f_7^{TDT});
- Veličina najveće slabo povezane komponente prvoga ulaznog grafa (f_8^{TDT});
- Veličina najveće slabo povezane komponente drugoga ulaznog grafa (f_9^{TDT}).

Model vektorskog prostora, koji na ovom zadatku koristimo kao temeljnu metodu, moguće je kao mjeru sličnosti dokumenata kombinirati s jezgrenim funkcijama nad grafovima događaja te na taj način dobiti hibridnu mjeru podudarnosti događaja opisanih u dokumentima. VSM se kombinira s jezgrenim funkcijama nad grafovima na način da sličnost dokumenata izračunatu putem kosinusa kuta između vektora dokumenata u vektorskom prostoru dodamo kao dodatnu značajku (f_{10}^{TDT}) u model nadziranog strojnog učenja (SVM-GRAF+VSM).

Rezultati

Skup podataka od 2016 parova dokumenata podijeljen je na skup za učenje i skup za ispitivanje u omjeru 70%–30%. Za svaki od tri modela temeljena na grafovima događaja kao i za temeljnu metodu VSM optimalna vrijednost praga α određena je na skupu za učenje, a potom je model s optimalnom vrijednošću parametra α vrednovan na skupu za ispitivanje. Za modele SVM-GRAF i SVM-GRAF+VSM optimalne vrijednosti hiperparametara algoritma SVM određene su pretraživanjem po rešetci putem deseterostruke unakrsne provjere na skupu za učenje, a potom su modeli s optimalnim vrijednostima hiperparametara vrednovani na skupu za ispitivanje. Rezultati vrednovanja na skupu za ispitivanje dani su u Tablici 8.2. Temeljna metoda i sva tri modela temeljena na pojedinačnim jezgrenim funkcijama imaju bolju preciznost nego odziv, što znači da postoje parovi dokumenata koji opisuju iste događaje, a koje ti modeli nisu u stanju prepoznati kao takve (takvi parovi predstavljaju lažne negativne primjere za klasifikator).

Visoka je preciznost modela očekivana jer su negativni parovi stvoreni na temelju dokumenata koji dolaze iz tematski vrlo različitih grupa (npr. tema "Sirijski građanski rat" nasuprot teme "Olimpijske igre u Londonu"), pa teško dolazi do pojave lažnih pozitivnih primjera. Modeli PGK-TENZORSKI i PGK-KONORMALNI temeljeni na jezgrenoj funkciji umnoška grafova uspješniji su od modela WDK temeljenog na jezgrenoj funkciji težinske dekompozicije grafova, što sugerira da je preklapanje šetnji u grafovima događaja bolji indikator podudarnosti događaja koje dokumenti opisuju od preklapanja podgrafova.

Nijedan od tri modela temeljena na pojedinačnim jezgrenim funkcijama nad grafovima događaja nije uspješniji od temeljnog VSM modela. Međutim, model SVM-GRAF koji na nelinearan način kombinira izlaze jezgrenih funkcija značajno je uspješniji od sva tri modela koji koriste pojedinačne jezgrene funkcije kao i od temeljnog modela VSM ($p < 0.05$; dvostrani t-test), iz čega zaključujemo da je pretpostavka komplementarnosti aspekata sličnosti koje pojedine jezgrene funkcije mjere bila ispravna. Nadalje, model SVM-GRAF+VSM, koji kombinira sličnost dokumenata određenu jezgrenim funkcijama nad grafovima događaja sa sličnošću dokumenata koju mjeri VSM, značajno je uspješniji od svih modela koji koriste isključivo jezgrene funkcije kao i od temeljnog modela VSM ($p < 0.01$; dvostrani t-test), što implicira da su mjere sličnosti temeljene na grafovima događaja (kao mjere usporedbe nad strukturiranim reprezentacijama dokumenata) komplementarne mjerama sličnosti koje dokumente promatraju kao vreće riječi odnosno koje zanemaruju strukturu dokumenata. Štoviše, model SVM-GRAF+VSM jedini je model među vrednovanima za koji je odziv veći od preciznosti, što znači da se parovi dokumenata koji su lažni negativni primjeri za modele temeljene na jezgrenim funkcijama nad grafovima događaja razlikuju od parova dokumenata koji su lažni negativni primjeri za temeljnu metodu VSM.

8.2.2 Rangiranje parova dokumenata prema podudarnosti događaja

Drugi zadatak kojim mjerimo sposobnost modela da prepoznaju dokumente koji opisuju iste događaje iz stvarnog svijeta usmjeren je na rangiranje parova dokumenata na temelju sličnosti koja odražava stupanj do kojeg su događaji koje ti dokumenti opisuju podudarni (Glavaš i Šnajder, 2013). Poseban naglasak u ovom zadatku stavljen je na razlikovanje između parova dokumenata koji opisuju iste događaje i parova dokumenata koji su tematski slični, ali ne opisuju iste događaje (npr. dokument koji opisuje "građanski rat u Sudanu" nasuprot dokumenta koji opisuje "građanski rat u Siriji".

Skup podataka

Za potrebe ovog zadatka pripremljen je skup parafraza novinskih tekstova koji je izgrađen na sljedeći način. Prvo je slučajno odabrano deset novinskih članaka s novinskog servisa EMM

NewsBrief. Potom su za svaki od odabranih novinskih članaka stvorene dvije različite *pozitivne parafraze* i dvije različite *negativne parafraze*. Pozitivna parafraza dobiva se parafraziranjem teksta na način da izvorno značenje ostane u potpunosti očuvano. U kontekstu događaja, pozitivna parafraza mora opisivati isti događaj kao i izvorni tekst. Negativna parafraza dobiva se parafraziranjem izvornog teksta na način da se bitno promijeni značenje. U kontekstu događaja, negativna parafraza tematski je slična izvornome tekstu, ali ne opisuje isti događaj iz stvarnog svijeta kao i izvorni tekst. Primjer odlomka novinskog članka s pripadnima mu pozitivnom i negativnom parafrazom dan je u tablici 8.3.

Pozitivne parafraze dobivene su primjenom postupka *kružnog prevođenja* (engl. *round-trip translation*). Kružno prevođenje postupak je automatskog parafraziranja teksta strojnim prevođenjem teksta u neki drugi jezik (ili nizom prevođenja kroz više jezika) te povratnim prevođenjem u izvorišni jezik. Kružno je prevođenje provedeno preko dvaju parova proizvoljno odabranih jezika: (1) hrvatskog i mađarskog te (2) danskog i finskog. U konkretnom slučaju za svaki su novinski tekst (pisan engleskim jezikom) stvorene dvije pozitivne parafraze novinskih tekstova i to:

1. Prevođenjem teksta s engleskog jezika na hrvatski, potom s hrvatskog jezika na mađarski te konačno s mađarskog jezika na engleski jezik;
2. Prevođenjem teksta s engleskog jezika na danski, potom s danskog jezika na finski te konačno s finskog jezika na engleski jezik.

Za prevođenje je korišten javno dostupni alat Google Translate[2]. Leksičko-semantičke i sintaktičke pogreške uzrokovane nesavršenim postupkom strojnog prevođenja ručno su ispravljene od strane označivača.

Za stvaranje negativnih parafraza angažirani su označivači kojima je dan naputak da originalni tekst izmijene tako da opisuje neki drugi događaj u odnosu na događaj koji tekst izvorno opisuje. Odabir promjena nad tekstom kojima će se to ostvariti prepušten je u potpunosti označivačima. Drugim riječima, nikakve konkretne transformacije teksta označivačima nisu bile predložene. Kako su za svaki izvorni dokument stvorene dvije negativne i dvije pozitivne parafraze, konačan skup podataka sastoji se od 60 parova dokumenata, od čega $10 \cdot \binom{3}{2} = 30$ pozitivnih parova i $10 \cdot \binom{3}{2} = 30$ negativnih primjera. Pozitivni parovi dokumenata nastaju uparivanjem pozitivnih parafraza s izvornim dokumentima te uparivanjem pozitivnih parafraza međusobno. Negativni parovi dokumenata nastaju uparivanjem negativnih parafraza s izvornim dokumentima te međusobnim uparivanjem negativnih parafraza, pri čemu je ručno provjereno da dvije različite negativne parafraze ne opisuju iste događaje (što je, iako malo vjerojatno, teoretski moguće jer su negativne parafraze istog dokumenta međusobno nezavisno sastavili različiti označivači).

[2] `http://translate.google.com`

Tablica 8.3: Parafraziranje događaja u novinskim tekstovima

Izvorni tekst
"*Europe's naval force patrolling off the coast East Africa said on Tuesday it had attacked Somali pirate installations on land, the first time it had conducted such an action since extending its remit strictly from sea-based initiatives. Initial reports indicated no casualties during the operation, which happened earlier on Tuesday. According to the European Union Naval Force (Somalia) Operation Atalanta's website, suspects are detained awaiting a decision of whether they face prosecution.*"
Pozitivna parafraza
"*European navy patrolling the coast of East Africa said on Tuesday it had attacked the Somali pirate equipment in the country. It had conducted such an operation by extending its mandate from sea-based operations only. Preliminary reports indicated that there were no casualties during the operation, which took place early Tuesday (Naval force website).*"
Negativna parafraza
"*Europe's naval forces detained nine suspected pirates and seized two skiffs near the Somali coast Tuesday after a merchant vessel issued a call of suspicious activity, EU officials said. Initial reports indicated that there were no casualties during the operation. According to the European Union Naval Force (Somalia) Operation Atalanta's website, suspects are detained awaiting a decision of whether they face prosecution.*"

Modeli i mjere za vrednovanje

Kao i na prethodnome zadatku, i na ovome zadatku uspoređujemo tri modela koji koriste različite jezgrene funkcije nad grafovima događaja: (1) model s jezgrenom funkcijom temeljenom na tenzorskom umnošku grafova (PGK-TENZORSKI), (2) model s jezgrenom funkcijom temeljenom na konormalnom umnošku grafova (PGK-KONORMALNI) i (3) model s jezgrenom funkcijom težinske dekompozicije grafa (WDK). Za razliku od prethodnog klasifikacijskog zadatka na kojemu je za svaku od jezgrenih funkcija trebalo odrediti prag na temelju kojeg su se dokumenti proglašavali podudarnima odnosno nepodudarnima, ovdje se rezultati jezgrenih funkcija izravno koriste za rangiranje parova dokumenata prema sličnosti događaja koje opisuju. I u ovome zadatku model vektorskog prostora (VSM) koristi se kao temeljna metoda.

Idealan model bi sve one parove dokumenata koji opisuju iste događaje rangirao iznad svih parova dokumenata koji opisuju različite događaje. Stoga za određivanje uspješnosti modela na ovom zadatku koristimo mjere vrednovanja koje ocjenjuju kvalitetu poretka, a uobičajeno se koriste u području pretraživanja informacija: (1) *preciznost na rangu r* (engl. *r-precision, precision at rank r*) i (2) *prosječnu preciznost* (engl. *average precision*, AP). Preciznost na rangu r iskazuje koliki je relativni udio dokumenata d u zbirci D koji su relevantni za upit q unutar prvih r dokumenata dohvaćenih od strane sustava za pretraživanje informacija:

$$P_r(D,q) = \frac{|\{d \in D_r \mid relevant(d,q)\}|}{r}, \tag{8.1}$$

gdje je D_r skup koji se sastoji od prvih r dokumenata dohvaćenih od strane sustava za pretraživanje informacija, a $relevant(d,q)$ predikat koji je zadovoljen ako i samo ako je dokument d relevantan za upit q. Prosječna preciznost AP računa se kao prosjek preciznosti na svim rangovima na kojima se nalazi neki dokument d relevantan za upit q:

$$AP(D,q) = \frac{\sum_{r \in R} P_r(D,q)}{|R|}, \tag{8.2}$$

gdje je R skup svih rangova na kojima se nalaze dokumenti relevantni za upit. U konkretnom zadatku rangiranja parova dokumenata, možemo zamisliti da se "upit" odnosi na dohvaćanje parova dokumenata koji opisuju iste događaje, pa su samo pozitivni parovi dokumenata relevantni.

Rezultati

Za svaki od tri modela temeljena na usporedbi grafova događaja jezgrenim funkcijama i temeljni model VSM parovi su dokumenata poredani silazno prema vrijednostima sličnosti izračunatima tim modelima. Uspješnost svih modela, izmjerena preciznošću na rangu 30 (budući da imamo 30 pozitivnih parova dokumenata) te prosječnom preciznošću, prikazana je u tablici

Tablica 8.4: Uspješnost rangiranja parova dokumenata na temelju sličnosti događaja koje opisuju

Model	Preciznost na rangu 30	Prosječna preciznost (AP)
PGK-TENZORSKI	86,7	96,8
PGK-KONORMALNI	**93,3**	**97,5**
WDK	86,7	95,7
VSM (temeljna metoda)	80,0	77,1

8.4. Svi modeli temeljeni na jezgrenim funkcijama značajno su uspješniji od temeljnog modela VSM po mjeri prosječne preciznosti ($p < 0.01$), dok je po mjeri preciznosti na rangu 30 jedino model PGK-KONORMALNI značajno uspješniji od temeljne metode ($p < 0.05$). Značajnost razlika u rezultatima testirana je neparametarskim statističkim testom stratificiranog miješanja (engl. *stratified shuffling test*) (Yeh, 2000). Model PGK-KONORMALNI neznačajno je uspješniji od modela PGK-TENZORSKI i WDK. Činjenica da jezgrene funkcije nad grafovima događaja negativnim parovima dokumenata dodjeljuju malene vrijednosti sugerira da je metoda temeljena na jezgrenim funkcijama posebno prikladna za razlikovanje dokumenata koji sadržajno pripadaju istoj temi, ali ne opisuju iste događaje iz stvarnog svijeta.

8.3 Pretraživanje informacija usmjerenih na događaje

Upiti usmjereni na događaje su upiti koji u sebi sadrže spominjanja događaja, poput upita

(28) *The party <u>won</u> the <u>elections</u> causing large <u>demonstrations</u>.*

U takvim upitima ključna informacija odgovara upravo spominjanjima događaja. Stoga model za pretraživanje informacija koji se ovdje predlaže počiva na dvjema idejama:

1. **Filtriranje informacija:** iz upita kao i iz dokumenata u zbirci nad kojom se radi pretraživanje izdvojiti samo informacije koje odgovaraju činjeničnim spominjanja događaja;
2. **Podudarnost strukture:** relevantnost dokumenta za upit određivati isključivo na temelju podudarnosti informacija koje odgovaraju činjeničnim spominjanjima događaja i vremenskoj strukturi među njima.

Sukladno navedenome, upite i dokumente predstavljamo kao grafove događaja, a relevantnost pojedinog dokumenta za zadani upit odgovara podudarnosti između grafa događaja upita i grafa događaja dokumenta, koju pak mjerimo jezgrenim funkcijama nad grafovima. Korištenjem jezgrenih funkcija nad grafovima događaja, moguće je prepoznati podudarnost pojedinih spominjanja događaja između upita i dokumenata, ali i podudarnost njihovih međusobnih vremenskih

odnosa.

8.3.1 Ispitne zbirke i označavanje relevantnosti

Standardna paradigma vrednovanja modela i sustava za pretraživanje informacija (tzv. Cranfield paradigma) (Voorhees, 2002) podrazumijeva postojanje ispitne zbirke koja se sastoji od dokumenata, upita i ocjena relevantnosti dokumenata za upite. Kako za informacijske potrebe usmjerene na događaje ne postoji standardna ispitna zbirka, za potrebe vrednovanja uspješnosti modela pretraživanja informacija temeljenih na grafovima događaja bilo je potrebno izgraditi odgovarajuće ispitne zbirke. Izgrađene su dvije ispitne zbirke od kojih svaka sadrži 50 upita. Prva zbirka, nazvana MIXEDTOPIC, sadrži 25.948 novinskih članaka, prikupljenih preko servisa EMM News Brief, koji se odnose na mnoštvo različitih tema. Druga zbirka, naziva ONETOPIC, sadrži 1.387 novinskih članaka koji su svi vezani za temu građanskog rata u Siriji. Druga zbirka nastala je odabiranjem iz prve zbirke samo onih dokumenata koji sadrže riječ "*Syria*" ili neku od njenih izvedenica (npr. "*Syrian*").[3] Za svaku od zbirki jedan je označivač izgradio 50 upita, i to na temelju sljedećih uputa:

1. Nasumično odabrati jedan dokument zbirke;
2. Pročitati dokument temeljito i s razumijevanjem;
3. Osmisliti upit koji sadržava barem dva spominjanja događaja tako da je odabrani dokument relevantan za osmišljeni upit.

Drugim riječima, označivač je za neki nasumično odabran dokument, odnosno za neki informacijski istaknuti dio dokumenta, trebao stvoriti vrlo kratak apstraktivni sažetak (engl. *abstractive summary*).[4] Nekoliko primjera upita dobivenih opisanim postupkom, zajedno s odlomcima izvornih dokumenata iz kojih su izvedeni, prikazano je u tablici 8.5.

[3]Zbirke MIXEDTOPIC i ONETOPIC dostupne su na adresi `http://takelab.fer.hr/data/retrographs`

[4]*Apstraktivni sažetak* nekog teksta jest sažetak koji ne sadrži iste fraze ili rečenice koje se pojavljuju u izvornom tekstu, već sažeto parafrazira ključne informacije izvornog teksta. Nasuprot toga, *ekstraktivni sažetak* (engl. *extractive summary*) sadrži informacijski najistaknutije fraze ili rečenice izvornog teksta.

Tablica 8.5: Upiti usmjereni na događaje s odlomcima novinskih članaka iz kojih su izvedeni.

Zbirka	Upit	Odlomak novinskog članka
MIXEDTOPIC	*United Russia won the elections triggering large demonstrations*	*The pro-Kremlin United Russia party won less than 50 per cent of the vote, a steep fall from its earlier majority, according to preliminary results. But opposition parties and international observers said the vote was marred by widespread reports of vote-rigging. Thousands of security forces were out in the Russian capital and helicopters roamed the sky Wednesday, a show of force following protests over scandal-marred elections that saw Prime Minister Vladimir Putin's party struggle to keep a majority.*
	The bank lost billions in trading and its stocks fell after the loss has been announced.	*A surprise $2 billion trading loss by a division of JPMorgan Chase triggered calls for tougher regulation of banks. Stock in the bank, the largest in the United States, lost 8 percent of its value on Wall Street immediately after the loss announcement, and other American and British banks suffered heavy losses as well.*
ONETOPIC	*Dozens of bodies were ditched in the streets after vindicatory attacks against Sunnis*	*Dozens of bodies were dumped in the streets of a Syrian city at the heart of the country's nearly 9-month-old uprising, a grim sign that sectarian bloodshed is escalating as the country descends further toward civil war. Up to 50 people were killed in Homs on Monday, but details about what happened in Syria's third-largest city only came to light Tuesday with reports of retaliatory attacks pitting members of the Alawite sect against Sunnis.*
	The minister met with the Syrian opposition and advocated them to unite.	*British Foreign Secretary William Hague on Monday called on Syrian opposition groups to "unite" against Syrian President Bashar al-Assad. Hague made the statement after meeting with Syrian opposition representatives in London on Monday.*

Potpuno označavanje relevantnosti dokumenata za upite podrazumijeva da se za svaki par dokument–upit označi je li dokument relevantan za dotični upit, tj. zadovoljava li dokument informacijsku potrebu korisnika izraženu upitom. Za iznimno velike zbirke dokumenata i veliki broj upita, međutim, potpuno označavanje relevantnosti nije izvedivo zbog velikog broja parova dokument–upit koje je potrebno označiti. Primjerice, za zbirku MIXEDTOPIC, koja sadrži 25.948 dokumenata i 50 upita, bilo bi potrebno označiti relevantnost za 1.297.400 parova dokument–upit što je, naravno, nemoguće izvesti s obzirom na vremenska i financijska ograničenja. Stoga se u takvim slučajevima standardno koristi metoda *probiranja dokumenata* (engl. *document pooling*) (Zobel, 1998; Voorhees, 2002), kojom se za pojedini upit značajno smanjuje broj dokumenata za koje je potrebno označiti relevantnost. Kod probiranja, skup dokumenata D za koje je potrebno označiti relevantnost za zadani upit q jest unija prvih N dokumenata dohvaćenih od svakog od modela za pretraživanje informacija koji sudjeluje u vrednovanju. Neka, primjerice, raspolažemo sa K modela za pretraživanje informacija (M_1, M_2, $\ldots$, M_K). Neka je D_{M_i} skup od N dokumenata koje je, za upit q, model M_i najviše rangirao. Tada je skup D dokumenata za koje se označava relevanost u odnosu na upit q određen unijom skupova D_{M_i}, tj. $D = D_{M_1} \cup D_{M_2} \cup \cdots \cup D_{M_K}$. U ovom vrednovanju, za probiranje dokumenata za označavanje relevantnosti korištena su dva modela: model vektorskog prostora s TF-IDF težinama (VSM) te jezični model (LM) temeljen na unigramima. Ovi modeli ujedno su korišteni kao temeljni modeli za usporedbu s modelima temeljenim na grafovima događaja. Modeli temeljeni na grafovima događaja nisu korišteni za probiranje dokumenata iz praktičnih razloga – izgradnja grafova događaja za sve dokumente u velikim zbirkama (poput zbirke MIXEDTOPIC) vremenski je i računalno zahtjevan postupak. Važno je, međutim, napomenuti kako izostavljanje modela temeljenih na grafovima događaja prilikom probiranja dokumenata za označavanje relevanosti samo ide na ruku temeljnim modelima VSM i LM, budući da je dokazano kako vrednovanje na zadatcima pretraživanja informacija zasnovano na probiranju dokumenata pogoduje upravo onim modelima koji sudjeluju u probiranju (Büttcher i dr., 2007).

Za zbirke MIXEDTOPIC i ONETOPIC za svaki je upit svakom od dvije temeljne metode probrano 75 dokumenata (tj. $N = 75$) jer (1) većina grupa preuzetih sa servisa EMM sadrži manje od 50 dokumenata, što znači da za većinu upita postoji najviše 50 relevantnih dokumenata u zbirci i jer (2) želimo vrlo dobru procjenu odziva čak i za upite čiji je izvor u nekoj od rijetkih grupa koje sadrže više od 50 dokumenata. Prosječna veličina skupa probranih dokumenata za koje je potrebno označiti relevantnost jest 95 dokumenata (što znači da između 75 dokumenata probranih modelom vektorskog prostora i 75 dokumenata probranih jezučnim modelom postoji značajno preklapanje).

Jedan je označivač označio relevatnost za sve probrane dokumente za svaki od 50 upita zbirke MIXEDTOPIC i svaki od 50 upita zbirke ONETOPIC. U početku označavanja drugi je označivač označio relevanosti probranih dokumenata za dva upita, pri čemu je uočeno potpuno

slaganje označivača u oznakama relevanosti što je omogućilo da daljnje označavanje provedemo samo jedan označivač. Savršeno slaganje označivača potvrdilo je intuiciju da je jednostavnije odrediti relevantnost dokumenata za upite koji opisuju događaje nego za upite temeljene na ključnim riječima, gdje tipično postoji izvjesno neslaganje među označivačima (Voorhees, 2002). U zbirci MIXEDTOPIC postoji prosječno 12 relevatnih dokumenata po upitu, dok u zbirci ONETOPIC prosječno po upitu postoji 8 relevatnih dokumenata.

8.3.2 Modeli pretraživanja informacija

Kao i na zadatcima TDT u prethodnom odjeljku, i ovdje koristimo tri različita modela temeljena na grafovima događaja, koji se međusobno razlikuju po jezgrenim funkcijama kojima mjere podudarnost grafa događaja upita s grafovima događaja dokumenata: (1) model s jezgrenom funkcijom temeljenom na tenzorskom umnošku grafova (PGK-TENZORSKI), (2) model s jezgrenom funkcijom temeljenom na konormalnom umnošku grafova (PGK-KONORMALNI) te (3) model s jezgrenom funkcijom težinske dekompozicije grafa (WDK). Kako bi rezultate koje ostvaruju modeli temeljeni na grafovima događaja stavili u odgovarajući kontekst, na zbirkama MIXEDTOPIC i ONETOPIC vrednuju se najuspješniji modeli triju tradicionalnih paradigmi pretraživanja informacija: modeli vektorskog prostora, jezični modeli i vjerojatnosni modeli). Konkretno, vrednuju se (1) model vektorskog prostora s težinskom shemom TF-IDF i kosinusom kao mjerom sličnosti vektora, (2) Hiemstrin (2001) jezični model te (3) dva modela (In_expC2 i DFR_BM25) iz radnog okvira divergencije od slučajnosti (engl. *divergence from randomness framework*, DFR) (Amati, 2003; Ounis i dr., 2006). Kao temeljne modele u eksperimentima koristimo implementaciju navedenih modela u platformi za pretraživanje informacija Terrier.[5]

Uspješnost modela za pretraživanje informacija temeljenih na grafovima događaja kao i svih prethodno navedenih temeljnih modela određujemo standardno korištenom mjerom *srednje prosječne preciznosti* (engl. *mean average precision*, MAP). Neka je D skup svih dokumenata, a Q skup svih upita neke ispitne zbirke te neka je, u skladu s izrazom (8.2), $AP(D,q)$ prosječna preciznost nekog modela na zbirci dokumenata D za upit q. Srednja prosječna preciznost tog modela na zbirci dokumenata D za skup upita Q tada nije ništa drugo do srednja vrijednost prosječnih preciznosti za sve upite:

$$MAP(D,Q) = \frac{\sum_{q \in Q} AP(D,q)}{|Q|}. \tag{8.3}$$

8.3.3 Rasprava rezultata

Uspješnost svih modela temeljenih na grafovima događaja i svih temeljnih modela određena mjerom MAP na zbirkama MIXEDTOPIC i ONETOPIC prikazana je u tablici 8.6. Rezultati u

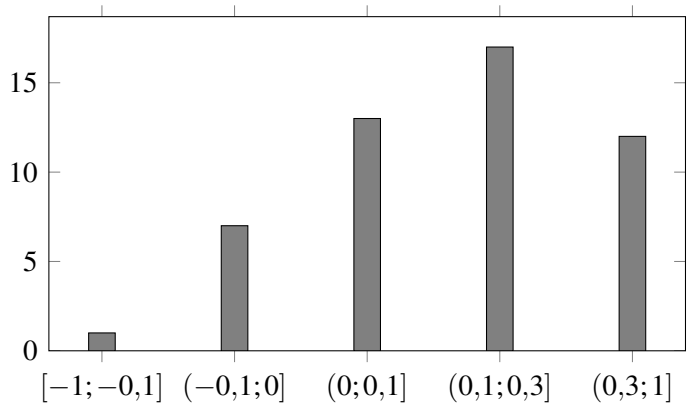
[5] `http://terrier.org`

Tablica 8.6: Uspješnost modela za pretraživanje informacija temeljenih na jezgrenim funkcijama nad grafovima događaja

		Ispitna zbirka	
	Model	MIXEDTOPIC	ONETOPIC
Temeljni modeli	TF-IDF VSM	0,335	**0,199**
	Hiemstra LM	0,300	0,175
	In_expC2	**0,341**	0,188
	DFR_BM25	0,332	0,192
Grafovi događaja	PGK-TENZORSKI	**0,502**	**0,407**
	PGK-KONORMALNI	0,434	0,359
	WDK	0,449	0,358
	NOSTRUCT	0,374	0,303

tablici 8.6 jasno pokazuju kako su svi modeli temeljeni na grafovima događaja značajno uspješniji od svih modela koji pripadaju tradicionalnim paradigmama pretraživanja informacija. Razlike u uspješnosti u odnosu na temeljne modele za model PGK-TENZORSKI značajne su uz $p < 0.01$, a za modele PGK-KONORMALNI i WDK uz $p < 0.05$ na obje ispitne zbirke (za testiranje je korišten dvostrani t-test). Ovakvi rezultati potvrđuju inicijalnu pretpostavku da je model pretraživanja informacija temeljen na strukturiranom predstavljanju informacija o događajima prikladniji za upite koji opisuju događaje iz stvarnog svijeta od tradicionalnih modela pretraživanja informacija. Iako je model PGK-TENZORSKI nešto uspješniji od modela PGK-KONORMALNI i WDK, razlike u rezultatima nisu statistički značajne. Međutim, sama činjenica da model PGK-KONORMALNI nije uspješniji od modela PGK-TENZORSKI sugerira da konormalni umnožak između grafa događaja upita i grafa događaja dokumenta češće uvodi lažne vremenske odnose u rezultantni graf nego što nadoknađuje nedostajuće vremenske odnose u nekom od ulaznih grafova događaja (u grafu dokumenta ili u grafu upita).

Uspješnost na zbirci ONETOPIC dosljedno je manja od uspješnosti na zbirci MIXEDTOPIC za sve modele, što ne iznenađuje, budući da su dokumenti zbirke ONETOPIC u prosjeku međusobno mnogo sličniji (jer pripadaju istoj temi građanskog rata u Siriji) nego dokumenti zbirke MIXEDTOPIC. Stoga na zbirci ONETOPIC modeli teže razlikuju relevantne od nerelevantnih dokumenata za pojedine upite. Važno je, međutim, primijetiti da je pad uspješnosti na zbirci ONETOPIC u odnosu na uspješnost na zbirci MIXEDTOPIC bitno manji za modele temeljene na grafovima događaja (17% za model PGK-KONORMALNI, 19% za model PGK-TENZORSKI i



Slika 8.1: Histogram razlika u iznosima prosječnih preciznosti
Histogram razlika u iznosima prosječnih preciznosti na upitima zbirke MIXEDTOPIC između najuspješnijeg modela temeljenog na grafovima događaja (PGK-TENZORSKI) i najuspješnijeg temeljnog modela (In_expC2)

20% za WDK) nego za temeljne modele (npr. 42% za DFR_BM25), na temelju čega zaključujemo da su modeli pretraživanja informacija temeljeni na grafovima događaja posebno pogodni za pretraživanje informacija u tematski ograničenim zbirkama dokumenata.

Mjera srednje prosječne preciznosti ne otkriva na koliko su upita modeli temeljeni na grafovima događaja doista uspješniji od temeljnih modela. Moguće je zamisliti dva kvalitativno vrlo različita modela pretraživanja informacija koji na svim pojedinačnim upitima daju vrlo različite vrijednosti prosječne preciznosti, ali u prosjeku približno istu vrijednost MAP. Kako bi se dobio bolji uvid u kvalitativna svojstva predloženih modela pretraživanja informacija, napravljena je usporedba razlika u iznosima prosječnih preciznosti (AP) po pojedinačnim upitima između modela PGK-TENZORSKI (najuspješniji model među modelima temeljenim na grafovima događaja) i modela In_ExpC2 (najuspješniji model među temeljnim modelima). Na slici 8.1 prikazan je histogram razlika u vrijednostima AP između ta dva modela za 50 upita zbirke MIXEDTOPIC. Iz histograma se vidi da je model PGK-TENZORSKI uspješniji od temeljnog modela In_expC2 za 42 od ukupno 50 upita zbirke MIXEDTOPIC. Razlika u vrijednostima AP modela najčešće (za 17 upita) se nalazi u rasponu od 0,1 do 0,3 u korist modela PGK-TENZORSKI, dok je za čak 12 upita ta razlika i veća od 0,3. Podrobnijom analizom za osam upita na kojima je vjerojatnosni model In_expC2 uspješniji od modela PGK-TENZORSKI utvrđeno je kako se radi o slučajevima gdje ili (1) važno spominjanje događaja nije ekstrahirano iz upita (dva takva slučaja) ili (2) model za razrješavanje koreferencije spominjanja događaja nije prepoznao koreferenciju između važnog spominjanja događaja iz upita i spominjanja događaja iz dokumenta (šest takvih slučajeva).

Iz informacijske perspektive, korištenjem grafova događaja za pretraživanje informacija u odnosu na tradicionalne modele pretraživanja informacija (1) filtriramo samo informacije koje se odnose na činjenične događaje i (2) uvodimo vremensku strukturu između događaja. Stoga

je važno analizirati koliko svaki od ta dva aspekta doprinosi uspješnosti modela za pretraživanje informacija temeljenih na grafovima događaja. U tu svrhu, ispitana je uspješnost modela koji koristi samo spominjanja događaja, ali ne i vremenske odnose među njima. Ovaj model, nazvan NOSTRUCT, rangira dokumente isključivo prema broju spominjanja događaja koja su podudarna sa spominjanjima događaja iz upita. Drugim riječima, dokumenti se rangiraju samo prema broju vrhova tenzorskog umnoška grafa događaja dokumenta i grafa događaja upita. Uspješnost modela NOSTRUCT dana je u posljednjem retku tablice 8.6. Svi modeli temeljeni na jezgrenim funkcijama nad grafovima događaja koji dodatno koriste i strukturu grafova događaja uspješniji su od modela NOSTRUCT (pri čemu je razlika statistički značajna jedino za model PGK-TENZORSKI uz $p < 0.05$), dok je, s druge strane, model NOSTRUCT uspješniji od svih temeljnih modela (pri čemu su razlike na ispitnoj zbirci ONETOPIC statistički značajne uz $p < 0.05$). Budući da već i filtriranje samo informacija koje opisuju događaje (model NOSTRUCT) daje rezultate bolje od temeljnih modela, a dodavanjem vremenske strukture (modeli PGK-TENZORSKI, PGK-KONORMALNI i WDK) uspješnost pretraživanja informacija još dodatno raste, zaključujemo kako i filtriranje informacija i uvođenje strukture značajno doprinose uspješnosti predloženih modela za pretraživanje informacija temeljenih na jezgrenim funkcijama primijenjenima nad grafovima događaja.

Poglavlje 9

Sažimanje i pojednostavljivanje teksta temeljem događaja

Tekstovi koji opisuju događaje iz stvarnog svijeta, poput novinskih članaka, obiluju informacijama o događajima – opisima samih radnji te opisima sudionika događaja i okolnosti kao što su vrijeme i mjesto odvijanja događaja. Ipak, takvi tekstovi često sadrže i značajnu količinu teksta koji podupire opise samih događaja, a koji se odnosi na pojašnjenja ili pozadinske činjenice. Pozadinske činjenice i opisi, koji imaju svrhu pojašnjavanja, nisu dio spominjanja događaja i stoga ne čine okosnicu priče. Razmotrimo odlomak novinskog članka koji opisuje pomorski incident između Kine i Filipina:

(29) *The Philippines vowed Thursday to defend what is theirs as part of a stand-off over a Chinese warship circling South China Sea. South China Sea is a home to a myriad of conflicting territorial claims.*

Posljednja rečenica u prethodnom primjeru može se smatrati pozadinskom činjenicom u kontekstu opisa recentnog sukoba.

U vrijeme obilja informacija kada informacije o istim događajima iz stvarnog svijeta pristižu iz mnoštva različitih tekstnih izvora vrlo je važno moći učinkovito razlučiti ključne informacije (informacije o događajima) od manje bitnih informacija (pozadinskih činjenica ili pojašnjenja). Sažimanje teksta (engl. *text summarization*) upravo ima za cilj izgraditi sažetak koji sadrži samo ključne informacije iz jednog ili više tematski povezanih tekstnih izvora, pri čemu je jednako važno da u sažetku nema informacijske zalihosti.

Osim što pored glavne priče u pravilu sadrže i pojašnjenja odnosno pozadinske činjenice, novinski su članci često pisani sintaktički vrlo složenim i stilski urešenim rečenicama. Zbog takvih rečenica ljudi s poteškoćama u čitanju imaju problema s razumijevanjem novinskih tekstova. Pojednostavljivanje teksta (engl. *text simplification*) ima za cilj učiniti tekstove razumljivijima nekoj ciljanoj skupini ljudi (npr. djeci ili ljudima s poteškoćama u čitanju).

Potreba za uklanjanjem informacijski manje važnih dijelova teksta postoji i kod sažimanja i kod pojednostavljivanja teksta, ali je drukčije motivirana. Dok kod sažimanja teksta nebitne informacije želimo ukloniti jer smo ograničeni veličinom sažetka, pa želimo zadržati samo najbitnije informacije, kod pojednostavljivanja teksta uklanjanje suvišnih informacija dovodi do povećanja razumljivosti teksta za određene skupine ljudi. Međutim, za razliku od pojednostavljenog teksta, gdje je važno da rečenice budu sintaktički i informacijski jednostavne, za sažetak je jedino bitno da sadrži ključne informacije, dok rečenice mogu ostati sintaktički složene i stilski ukrašene. Stoga, iako su u nekim aspektima slični, sažimanje i pojednostavljivanje teksta dva su različita zadatka obrade prirodnog jezika.

U ovom poglavlju opisujemo algoritme za automatsko sažimanje i pojednostavljivanje novinskih tekstova temeljene na informacijama o događajima ekstrahiranima iz teksta. Algoritam za sažimanje grupa tematski povezanih dokumenata (opisan u poglavlju 9.1) određuje važnost pojedinačnih spominjanja događaja na temelju grafova događaja (Glavaš i Šnajder, 2014b), dok algoritam za pojednostavljivanje teksta (opisan u poglavlju 9.2) zadržava samo dijelove teksta vezane za spominjanja događaja, pri čemu svako spominjanje događaja predstavlja vlastitom sintaktički jednostavnom rečenicom (Glavaš i Štajner, 2013).

9.1 Sažimanje grupa dokumenata temeljem grafova događaja

Iako je algoritam za sažimanje temeljen na grafovima događaja primjenjiv i na pojedinačne dokumente, u ovoj je disertaciji razmatran zahtjevniji zadatak – sažimanje grupa tematski povezanih dokumenata (engl. *multi-document summarization*). Cilj sažimanja grupa dokumenata jest crpljenje najinformativnijeg sadržaja iz izvornih dokumenata te prezentacija tog sadržaja u sažetom obliku na način prikladan nekoj zadanoj primjeni (Mani, 2001). U kontekstu događaja, sažimanje teksta na temelju više povezanih dokumenata omogućava povezivanje bitnih informacija o događajima koje su raspršene kroz više dokumenata te, posljedično, prikaz svih ključnih informacija o događaju na jednome mjestu.

Sažetak koji nastaje kao rezultat postupka sažimanja ograničen je veličinom, a mora zadovoljiti tri osnovna kriterija: informativnost, koherentnost i gramatičku ispravnost. Dobar sažetak također ne smije biti zalihosan (tj. ne smije biti ponavljanja informacija), a zadovoljavanje tog kriterija pogotovo je izazov kod automatskog sažimanja grupa tematski povezanih dokumenata, gdje je informacijsko preklapanje između dokumenata grupe veliko. Pristupi sažimanju grupa dokumenata mogu biti ekstraktivni i apstraktivni. Apstraktivni pristupi rezultiraju sažecima koji ne sadrže fraze ili rečenice iz izvornih dokumenata, dok ekstraktivni sažeci sadrže rečenice (ili dijelove rečenica) iz izvornih dokumenata koje su ocijenjene kao najinformativnije. Nadalje, sažimanje teksta može biti usmjereno i neusmjereno. Neusmjereno sažimanje teksta (engl. *non-focused summarization*) jest sažimanje koje nije vođeno nekom specifičnom informacijskom

potrebom izraženom u obliku upita. Nasuprot tome, usmjereno sažimanje (engl. *focused summarization*) crpi informacije koje su najvažnije u kontekstu neke unaprijed zadane informacijske potrebe dane u obliku upita.

U tekstovima koji opisuju događaje iz stvarnog svijeta kao što su novinski članci, najvažnija je informacija pohranjena u spominjanjima događaja. U skladu s tim, postupak za sažimanje grupa dokumenata trebao bi biti prvenstveno usmjeren na spominjanja događaja. Stoga je razvijen algoritam za ekstraktivno sažimanje grupa dokumenata usmjerenih na događaje koji koristi informacije o događajima koje su automatski ekstrahirane iz teksta u obliku grafova događaja. U nastavku je detaljno opisan algoritam za automatsko sažimanje grupa dokumenata temeljen na grafovima događaja (poglavlje 9.1.2) kao i eksperimenti kojima je vrednovana uspješnost algoritma (poglavlje 9.1.3). U nastavku je najprije dan pregled srodnih istraživanja koja se bave sažimanjem dokumenata koristeći informacije o događajima.

9.1.1 Pristupi sažimanju teksta temeljeni na događajima

Pristupi sažimanju grupa dokumenata opisani u literaturi u pravilu su usmjereni na novinske tekstove, koji u prvom redu sadrže opise događaja iz stvarnog svijeta. Međutim, unatoč činjenici da je praćenje događaja kroz niz novinskih objava prototipni primjer primjene postupka automatskog sažimanja grupa dokumenata (Barzilay i dr., 1999), broj pristupa koji koriste informacije o događajima za sažimanje teksta iznimno je malen.

Korištenje događaja za sažimanje grupa dokumenata prvi su predložili Daniel i suradnici (2003). Pridržavajući se definicije događaja TDT (v. poglavlje 3.5), rečenice koje ulaze u sažetak odabiru prema zastupljenosti poddogađaja (engl. *sub-events*) glavnog događaja koji dokumenti grupe opisuju. Označivači su za svaki događaj ručno odredili poddogađaje te svakoj rečenici za svaki poddogađaj pridijelili vrijednost koja označava koliko je poddogađaj zastupljen u rečenici. Algoritam potom za sažetak odabire one rečenice koje imaju najveći zbroj vrijednosti po svim relevantnim poddogađajima. Ovdje se, međutim, ne radi o potpuno automatiziranom pristupu sažimanju teksta budući da je potrebna ljudska intervencija za prepoznavanje poddogađaja te određivanje zastupljenosti svih poddogađaja po svim rečenicama.

Za razliku od prethodnog pristupa, koji događaje promatra na razini dokumenata, Filatova i Hatzivassiloglou (2004) promatraju događaje kroz spominjanja na rečeničnoj razini. Spominjanje događaja, međutim, definiraju neizravno, kao supojavljivanje dvaju imenovanih entiteta između kojih postoji glagol ili glagolska imenica. Dva su temeljna nedostatka ovakvog pristupa. Prvi je da nisu svi glagoli i glagolske imenice sidra spominjanja događaja. Drugi je da su nerijetka informacijski vrlo važna spominjanja događaja koja nemaju nemaju niti jedan imenovani entitet među svojim argumentima, poput spominjanja "*detonated*" i "*wounding*" u sljedećem primjeru:

(30) *The rebels then detonated the bomb, wounding at least dozen people.*

Li i suradnici (2006) proširuju pristup Filatove i Hatzivassilogloua (2004) na način da grade graf supojavljivanja između imenovanih entiteta i sidara događaja. Nakon pridijeljivanja početnih informacijskih težina imenovanim entitetima i sidrima događaja na temelju učestalosti njihovog spominjanja u grupi dokumenata, informacijske vrijednosti propagiraju kroz graf supojavljivanja algoritmom PageRank (Page i dr., 1999), čime dobivaju zaglađene vrijednosti relevantnosti pojedinih sidara događaja i imenovanih entiteta. Konačno, informacijsku vrijednost rečenice računaju kao zbroj informacijskih vrijednosti sidara događaja i imenovanih entiteta koji se u toj rečenici nalaze. Crpljenje sidara događaja, međutim, ne obavljaju automatski, a uz to zanemaruju međusobne semantičke odnose sidara događaja (npr. vremenski odnosi ili odnosi uzroka i učinka), kao i odnose između sidara događaja i imenovanih entiteta (npr. semantička uloga imenovanog entiteta). Slično kao i u radu Filatove i Hatzivassilogloua (2004), rečenice koje ne sadrže imenovane entitete nemaju mogućnost dobiti visoku informacijsku vrijednost, čak niti onda kada su potencijalno vrlo informativne.

Istraživanja u području sažimanja temeljem ažuriranja informacija (engl. *update summarization*) (Dang i Owczarzak, 2008; Du i dr., 2010; Yan i dr., 2011; He i dr., 2012) usmjerena su na izdvajanje novih informacija koji kontinuirano pristižu u toku dokumenata (engl. *on-line document stream*). Nove informacije koje je potrebno izdvojiti iz nekog dokumenta su one informacije koje nisu postojale u prethodnim dokumentima toka. Takav oblik sažimanja potreban je u okruženjima gdje je korisnik upoznat sa sadržajem prethodno pristiglih dokumenata u trenutku kad pristiže novi dokument. Nisu poznati pristupi koji bi koristili spominjanja događaja na rečeničnoj razini za postupke sažimanja temeljem ažuriranja informacija.

Algoritam za sažimanje grupa dokumenata predstavljen u ovome poglavlju podrazumijeva da korisnik nije prethodno upoznat sa sadržajem dokumenata grupe. Algoritam obavlja ekstraktivno i neusmjereno sažimanje grupa dokumenata (engl. *extractive non-focused multi-document summarization*), koristeći pritom informacije o događajima strukturirane u obliku grafova događaja. Algoritam prvo procjenjuje informativnost pojedinačnih spominjanja događaja, a potom dodatno rafinira procjene informativnosti spominjanja događaja na temelju njihovih međusobnih vremenskih odnosa, gradeći na zrenju da događaj ima veću važnost ukoliko se dogodio neposredno prije ili neposredno poslije nekog drugog važnog događaja.

9.1.2 Algoritam sažimanja teksta temeljen na grafovima događaja

Algoritam za sažimanje grupa dokumenata ekstraktivni je algoritam koji za sažetak odabire najinformativnije rečenice iz izvornih dokumenata. Rad algoritma temelji se na dvjema pretpostavkama:

1. Značajnost rečenice proporcionalna je značajnosti spominjanja događaja koja se nalaze u

Algoritam 1: Sažimanje(D, l)

1: **Ulaz**: Grupa dokumenata D; duljina sažetka l
2: **Inicijalizacija**:
3: G_d – graf događaja za dokument d iz grupe D
4: $E(D)$ – skup svih spominjanja događaja svih dokumenata grupe D
5: $E(d)$ – skup svih spominjanja događaja dokumenta d
6: $Sent(D)$ – skup svih rečenica svih dokumenata grupe D
7: **Sažmi**:
8: $P_{scores} \leftarrow \emptyset$; $I_{scores} \leftarrow \emptyset$;
9: **za svako** spominjanje događaja e iz $E(D)$ **učini**
10: $P_{scores}[e] \leftarrow S_P(e)$
11: $I_{scores}[e] \leftarrow S_I(e)$
12: **za svaki** dokument $d \in D$ **učini**
13: $P_{scores} \leftarrow$ PageRank(G_d, $E(d)$, P_{scores})
14: $I_{scores} \leftarrow$ PageRank(G_d, $E(d)$, I_{scores})
15: $R_P \leftarrow$ sortiraj($E(D)$, P_{scores})
16: $R_I \leftarrow$ sortiraj($E(D)$, I_{scores})
17: $S_{scores} \leftarrow \emptyset$
18: **za svako** spominjanje događaja e iz $E(D)$ **učini**
19: $S_{scores}[e] \leftarrow$ rang(e, R_P) + rang(e, R_I)
20: $Sent_{scores} \leftarrow \emptyset$
21: **za svaki** dokument d iz D **učini**
22: **za svaku** rečenicu s iz d **učini**
23: $Sent_{scores}[s] \leftarrow 0$
24: **za svako** spominjanje događaja e iz s **učini**
25: $Sent_{scores}[s] = Sent_{scores}[s] + S_{scores}[e]$
26: $C \leftarrow$ grupiraj_recenice($Sent(D)$)
27: $CR \leftarrow$ predstavnici_grupa(C, $Sent_{scores}$)
28: $CR \leftarrow$ sortiraj_silazno(CR, $Sent_{scores}$)
29: $saetak \leftarrow \emptyset$
30: **učini**
31: $saetak \leftarrow saetak \cup$ uzmi_prvu(CR)
32: **dok vrijedi** duljina($saetak$) $\leq l$
33: **Izlaz**: $saetak$

toj rečenici;

2. Značajnost spominjanja događaja moguće je pouzdano procijeniti na temelju informacija o događajima koje su ekstrahirane iz teksta, a koje su sadržane u grafovima događaja pojedinačnih dokumenata grupe.

Algoritam za sažimanje grupa dokumenata temeljem grafova događaja pregledno je prikazan u pseudokodu Algoritam 1. Početni je korak izgradnja grafova događaja za sve dokumente u grupi (redak 3 u pseudokodu algoritma). Nakon toga se za sve dokumente informacijska zna-

čajnost svakog spominjanja događaja određuje na temelju triju kriterija: (1) važnosti sudionika događaja, (2) informativnosti riječi koje čine događaj te (3) vremenskih odnosa promatranog događaja s drugim događajima. Određivanje informacijske značajnosti na temelju navedenih kriterija provodi se u dva koraka. Prvo se za svako spominjanje događaja računaju vrijednost važnosti njegovih sudionika (redak 10) i vrijednost informativnosti riječi događaja (redak 11). U drugom se koraku obje prethodno izračunate vrijednosti zaglađuju algoritmom PageRank na cjelokupnom grafu događaja. Drugim riječima, zaglađivanje vrijednosti dodijeljenih pojedinačnim spominjanjima događaja obavlja se na temelju vremenskih odnosa među spominjanjima događaja. Nakon izvršavanja algoritma PageRank za obje vrijednosti (važnost sudionika događaja i informativnost riječi događaja), spominjanja događaja rangiramo prema svakoj od te dvije vrijednosti (retci 13 i 14). Konačna značajnost spominjanja događaja odgovara zbroju rangova tog spominjanja u objema rangiranim listama (retci 18 i 19). Nakon što smo izračunali konačne vrijednosti značajnosti za svako spominjanje događaja, značajnost rečenica određujemo zbrajanjem značajnosti svih spominjanja događaja koji se nalaze u tim rečenicama (retci 20–25). Kako bismo osigurali da sažetak ne sadrži zalihosne informacije, rečenice grupiramo prema semantičkoj sličnosti (redak 26). Iz svake grupe semantički sličnih rečenica potom odabiremo onu rečenicu koja ima najveću dodijeljenu vrijednost značajnosti (redak 27) nakon čega odabrane rečenice koje predstavljaju grupe sortiramo silazno prema dodijeljenoj im vrijednosti značajnosti (redak 28) i dodajemo tim redoslijedom u sažetak dok god ne premašimo ograničenje veličine sažetka (retci 29–32). U nastavku slijedi detaljan opis svakog od ključnih koraka algoritma.

Važnost sudionika događaja

Prva vrijednost na temelju koje procjenjujemo značajnost spominjanja događaja jest važnost njegovih sudionika. Sudionicima događaja smatramo imenovane entitete koji su u postupku crpljenja informacija o događajima prepoznati kao argumenti događaja sa semantičkom ulogom AGENT ili TARGET. Intuitivno, sudionici koji se često pojavljuju u dokumentima grupe važniji su za temu koju grupa određuje. S druge strane, sudionici koji se pojavljuju u samo manjem broju dokumenata grupe (npr. samo u jednom dokumentu), pa makar i često, manje su važni za cijelu temu od sudionika koji se pojavljuju u većini ili u svim dokumentima grupe. U skladu s navedenim, važnost sudionika za temu određujemo na temelju ukupne frekvencije pojavljivanja tog imenovanog entiteta u svim dokumentima grupe te na temelju broja dokumenata grupe u kojima se taj imenovani entitet pojavljuje barem jednom.

Neka je D grupa dokumenata za koju treba izgraditi sažetak, neka je d jedan dokument grupe D te neka je e jedno spominjanje događaja unutar dokumenta d. Važnost sudionika spominjanja

događaja e računa se na sljedeći način:

$$S_P(e) = \sum_{p \in P(e)} \sum_{d \in D} \#(p,d) \cdot \frac{\left|\{d \in D \mid \#(p,d) > 0\}\right|}{|D|} \tag{9.1}$$

gdje je $P(e)$ skup svih sudionika događaja e, p pojedinačni sudionik događaja e, a $\#(p,d)$ je broj pojavljivanja sudionika p u dokumentu d.

Važno je naglasiti kako postoje pristupi koji koriste slične mjere informativnosti imenovanih entiteta za određivanje informacijske važnosti rečenica (Ge i dr., 2003; Saggion i Gaizauskas, 2004; Ouyang i dr., 2011). Glavna razlika takvih pristupa u odnosu na ovdje predloženi pristup jest u tome da ovdje predložena mjera važnosti sudionika događaja razmatra isključivo važnost imenovanih entiteta koji se pojavljuju kao sudionici događaja, čime se sprječava da se u sažetak uvrste pozadinske rečenice koje sadrže važne imenovane entitete koji, međutim, nisu vezani niti za jedan konkretan događaj, poput sljedeće rečenice:

(31) *China and Philippines have many unresolved territorial issues in the South China Sea.*

Informativnost riječi događaja

Važnost sudionika događaja za temu u mnogim je slučajevima vrlo indikativna za informacijsku važnost događaja. Međutim, nerijetki su primjeri vrlo informativnih događaja koji uopće ne sadrže imenovane entitete, poput primjera:

(32) *Insurgents launched a massive attack early in the morning.*

S druge strane, postoje i primjeri događaja koji kao svoje argumente imaju važne sudionike, ali ipak nisu važni za temu. Na primjer, rečenica

(33) *Obama and Putin took the photograph together at the Lough Erne Resort.*

nije informacijski posebno važna u kontekstu, primjerice, teme sastanka predstavnika zemalja članica G8. Iz ovih je razloga informativnost događaja potrebno moći procijeniti mjerom koja je na neki način komplementarna važnosti sudionika događaja. U tu se svrhu predlaže mjera koja računa informativnost riječi koje čine spominjanje događaja – informativnost sidra događaja i svih riječi koje čine bilo koji od argumenata događaja. Informativnost pojedine riječi koja tvori spominjanje događaja računamo kao razliku relativne učestalosti pojavljivanja te riječi unutar grupe dokumenata koju treba sažeti i relativne učestalosti pojavljivanja te riječi u nekoj općenitoj i tematski neograničenoj zbirci dokumenata. Dvije su pretpostavke koje opravdavaju ovakav način računanja informativnosti spominjanja događaja:

1. Riječi čija je relativna učestalost unutar dokumenata teme bitno veća od relativne frekvencije unutar neke općenite zbirke značajne su za zadanu temu;

2. Spominjanja događaja koja se sastoje od riječi koje su važne za neku temu i sama su važna za tu temu.

Neka je D skup tematski povezanih dokumenata za koje treba izgraditi sažetak te neka je C velika tematski neograničena zbirka dokumenata. Mjeru informativnosti riječi spominjanja događaja e računamo kako slijedi:

$$S_I(e) = \sum_{w \in W(e)} \left(\frac{\#(w,D)}{\sum_{w' \in W(D)} \#(w',D)} - \frac{\#(w,C)}{\sum_{w' \in W(C)} \#(w',C)} \right), \tag{9.2}$$

gdje je $W(C)$ skup svih različitih riječi koje se pojavljuju u zbirci C, $W(e)$ skup svih riječi koje čine spominjanje događaja e (tj. sidro događaja e i sve riječi koje su dio bilo kojeg argumenta događaja e), a $\#(w,C)$ je broj pojavljivanja riječi w u zbirci C. Kao velika zbirka tematski različitih dokumenata C u eksperimentima je korištena zbirka Google Books Ngrams (Michel i dr., 2011).

Slične mjere koje se temelje na informativnosti riječi korištene su za određivanje informativnosti rečenica u dokumentima (Lin i Hovy, 2000; Conroy i dr., 2004; Nenkova i Vanderwende, 2005). Ključna je razlika, međutim, u tome što se ovdje predloženi pristup oslanja na informativnost samo onih riječi koje su sastavni dio spominjanja događaja, a ne svih riječi u rečenici. Riječi koje nisu dio spominjanja događaja često su vrlo neinformativne (otprilike jednake relativne učestalosti u svim zbirkama) i stoga pribrajanje njihove informativnosti može značajno narušiti procjenu informativnosti dužih rečenica. Razmatrajući samo riječi koje pripadaju spominjanjima događaja, taj se problem izbjegava.

Zaglađivanje temeljem vremenske strukture

Osim intrinzične značajnosti spominjanja događaja koja proizlazi iz prethodno opisanih mjera važnosti sudionika i informativnosti riječi, značajnost događaja za temu ovisi i o njegovim vremenskim odnosima s drugim događajima. Drugim riječima, spominjanje događaja smatramo značajnijim za temu ukoliko je u izravnom vremenskom odnosu s nekim drugim značajnim događajem. Neka se, primjerice, događaj e_1 dogodio neposredno nakon događaja e_2 koji je vrlo značajan za temu predstavljenu grupom dokumenata. Tada pretpostavljamo da je vjerojatno da je i događaj e_2 značajan za temu te da mu treba povećati iznos mjere značajnosti. Ovakvo rafiniranje značajnosti spominjanja događaja na temelju vremenskih odnosa s drugim događajima nazivamo *zaglađivanjem po vremenskoj strukturi dokumenta* (engl. *smoothing over temporal structure*). Zaglađivanje vrijednosti značajnosti spominjanja događaja provedeno je algoritmom PageRank (Page i dr., 1999) nad strukturom grafa događaja.

Algoritam PageRank prvotno je osmišljen za rangiranje internetskih stranica prema njihovoj važnosti. Osnovna je ideja algoritma u tome da je vrh u grafu to značajniji što je povezaniji s

drugim visoko značajnim vrhovima grafa. Značajnost nekog vrha trebala bi biti još veća ukoliko njegovi značajni susjedni vrhovi nemaju puno drugih susjeda osim njega. Neka je $\mathbf{W}$ matrica susjedstva grafa događaja G koja je normalizirana po retcima. Algoritam PageRank iterativno prilagođava vektor značajnosti vrhova grafa $\mathbf{a}$ na sljedeći način:

$$\mathbf{a}^{(\mathbf{k})} = \alpha \mathbf{a}^{(\mathbf{k-1})} \mathbf{W} + (\mathbf{1} - \alpha)\mathbf{s} \tag{9.3}$$

gdje je α faktor prigušenja (engl. *damping factor*) algoritma PageRank. Vektor $\mathbf{s}$ predstavlja normalizirani izvor vrijednosti za sve vrhove, pa zbroj elemenata tog vektora iznosi 1. Kao izvore vrijednosti vrhova (tj. kao elemente vektora $\mathbf{s}$) koristimo normalizirane vrijednosti važnosti sudionika događaja (S_P) i informativnosti riječi događaja (S_I). Drugim riječima, kada se zaglađivanje algoritmom PageRank vrši po vrijednostima važnosti sudionika događaja, onda element s_i vektora $\mathbf{s}$ iznosi $S_P(e_i)/\sum_{j=1}^{n} S_P(e_j)$, a kada se zaglađivanje obavlja po vrijednostima informativnosti riječi događaja tada s_i iznosi $S_I(e_i)/\sum_{j=1}^{n} S_I(e_j)$. Algoritam PageRank, dakle, na svakom grafu događaja izvodimo dva puta: jednom za vrijednosti važnosti sudionika događaja, a drugi put za vrijednosti informativnosti riječi događaja.

Sprječavanje zalihosti

Grupiranje semantički podudarnih (ili približno podudarnih) rečenica izvornih dokumenata uobičajen je korak kojim se pokušava sprječiti zalihost informacija u sažetku (Saggion i Gaizauskas, 2004; Christensen i dr., 2013). U ovom algoritmu za sažimanje grupa dokumenata semantička sličnost rečenica mjeri se modelom TakeLab-STS[1] (Šarić i dr., 2012), jednim od najuspješnijih modela za računanje semantičke sličnosti kratkih tekstova (Agirre i dr., 2012). Grupiranje se provodi algoritmom aglomerativnog grupiranja s jednostrukom povezanošću (engl. *single-linkage agglomerative clustering*) (Gower i Ross, 1969) na temelju semantičke sličnosti između rečenica koje daje model TakeLab-STS. Algoritam aglomerativnog grupiranja početno svaku rečenicu svrstava u zasebnu grupu, da bi se potom u svakom koraku dvije najsličnije grupe stopile u jednu grupu. Sličnost između grupa kod algoritma s jednostrukom povezanošću odgovara najvećoj sličnosti koja postoji između neke rečenice prve grupe i neke rečenice druge grupe. Stapanje grupa zaustavlja se kada više ne postoji niti jedan par grupa čija je sličnost veća od nekog unaprijed zadanog praga t.

Zalihost informacija u sažetku sprječava se tako da se iz svake grupe semantički sličnih rečenica odabere samo jedna reprezentativna rečenica. Iz svake se grupe kao predstavnik odabire ona rečenica koja ima najveću vrijednost značajnosti za temu (tj. najveći zbroj značajnosti spominjanja događaja koje sadrži). Konačno, odabrane reprezentativne rečenice grupa rangiraju se po značajnosti za temu (od najznačajnijih prema najmanje značajnima) i redom dodaju u

[1] `http://takelab.fer.hr/sts`

sažetak sve dok se ne dosegne ograničenje duljine sažetka.

9.1.3 Eksperimentalno vrednovanje

Kako je rečeno ranije, opisanim algoritmom za sažimanje grupa tematski povezanih dokumenata obavlja se ekstraktivno neusmjereno sažimanje grupa dokumenata pisanih engleskim jezikom. Stoga su za vrednovanje uspješnosti algoritma potrebni skupovi podataka koji zadovoljavaju iste kriterije, tj. zbirke koje sadrže sažetke koje su napisali ljudi odabirući najinformativnije rečenice izvornih tekstova i bez namjere da zadovolje neke specifične informacijske potrebe.

Zbirke tekstova i mjere za vrednovanje

Zbirke tekstova koje su postale *de facto* standard za vrednovanje automatskih postupaka za sažimanje grupa dokumenata jesu zbirke nastale u okviru kampanja vrednovanja DUC (Document Understanding Conference), koje su se održavale od 2001. do 2007. godine. Dva su zajednička zadatka s konferencija DUC bila usmjerena na ekstraktivno neusmjereno sažimanje grupa dokumenata: (1) zadatak ekstraktivnog sažimanja iz kampanje DUC-2002, gdje je za grupe tematski povezanih dokumenata trebalo izgraditi sažetke duljine do 200 pojavnica (Over i Liggett, 2002) i (2) zadatak 2 iz kampanje DUC-2004, gdje je za grupe tematski povezanih dokumenata trebalo izgraditi sažetke duljine do 100 pojavnica (Over i Yen, 2004). Zbirka DUC-2002 sastoji se od 59 grupa koje sadrže između 5 i 15 tematski povezanih novinskih članaka, dok se zbirka DUC-2004 sastoji od 50 grupa od kojih svaka sadrži 10 tematski povezanih novinskih članaka.

Uspješnost automatskog postupka za sažimanje teksta mjeri se na način da se automatski izgrađeni sažeci uspoređuju s referentnim sažecima koje su sastavili ljudi. Predloženo je više mjera za automatsko uspoređivanje sažetaka izgrađenih automatskim postupkom s referentnim sažecima, od kojih su mjere $ROUGE_1$ i $ROUGE_2$ (Lin, 2004) među najčešće korištenima u literaturi. Neka je s_c automatski izgrađen sažetak, neka je s_r referentni sažetak te neka je $W_N(s)$ skup svih ngrama duljine N u sažetku s. Mjera $ROUGE_N$ računa udio ngrama duljine N referentnog sažetka s_r koje možemo pronaći u automatski izgrađenom sažetku s_c:

$$ROUGE_N(s_c, s_r) = \frac{|\{W_N(s_c) \cap W_N(s_r)\}|}{|W_N(s_r)|} \tag{9.4}$$

U skladu s ovom formulom, mjera $ROUGE_1$ broji riječi referentnog sažetka koje pronalazimo u automatski izgrađenom sažetku, dok mjera $ROUGE_2$ mjeri udio bigrama (dvije uzastopne riječi) referentnog sažetka koje pronalazimo u automatski izgrađenom sažetku. Često je slučaj da na raspolaganju imamo više referentnih sažetaka za istu grupu dokumenata. U tom slučaju računamo vrijednosti $ROUGE_N$ između automatski izgrađenog sažetka i svakog od referentnih sažetaka te odabiremo najveću od dobivenih vrijednosti. Neka je S_r skup koji sadrži više od

jednog referentnog sažetka, a s_c automatski izgrađen sažetak. Kada imamo više referentnih sažetaka, vrijednost $ROUGE_N$ za automatski izgrađen sažetak računamo na sljedeći način:

$$ROUGE_N(s_c, S_r) = \max_{s_r \in S_r} ROUGE_N(s_c, s_r) \tag{9.5}$$

Kako su mjere $ROUGE_N$ za mjerenje kvalitete automatski izgrađenih sažetaka predložene 2004. godine, one još nisu bile korištene u kampanji DUC-2002. Za potrebe usporedbe rezultata sa sustavima za sažimanje koji su sudjelovali u kampanji DUC-2002, sažetci koji su rezultati tih sustava također su vrednovani mjerama $ROUGE_1$ i $ROUGE_2$.

Modeli

Model za sažimanje grupa dokumenata temeljen na grafovima događaja ima dva parametra čije je vrijednosti potrebno optimirati: (1) prag sličnosti rečenica t pri kojem se zaustavlja grupiranje te (2) faktor prigušenja α algoritma PageRank. Zbirka DUC-2002, koja se sastoji od 59 grupa dokumenata koje treba sažeti, podijeljena je stoga na dva dijela – skup za učenje, koji se sastoji od 30 nasumično odabranih grupa, te skup za ispitivanje u kojem se nalazi preostalih 29 grupa. Vrijednosti parametara t i α optimirane su na skupu za učenje (optimalne vrijednosti su $t = 0,3$ i $\alpha = 0,15$), da bi potom model s optimalnim vrijednostima parametara bio vrednovan na ispitnom dijelu zbirke DUC-2002 te na cjelovitoj zbirci DUC-2004.

Vrednovane su četiri različite inačice modela za sažimanje grupa dokumenata temeljem grafova događaja:

1. Model koji značajnost spominjanja događaja e ocjenjuje samo na temelju važnosti sudionika događaja $S_P(e)$ (model DOGAĐAJI-VSD);
2. Model koji značajnost spominjanja događaja e ocjenjuje samo na temelju informativnosti riječi spominjanja događaja $S_I(e)$ (model DOGAĐAJI-IRD);
3. Model koji značajnost spominjanja događaja e određuje kombinirajući važnost sudionika događaja $S_P(e)$ i informativnost riječi događaja $S_I(e)$ (model DOGAĐAJI-VSD + IRD);
4. Potpuni model koji koristi zaglađivanje algoritmom PageRank na temelju strukture grafa događaja, kako za važnost sudionika događaja, tako i za informativnost riječi događaja (model DOGAĐAJI- VSD + IRD + PAGERANK).

Modele temeljene na događajima uspoređujemo s najuspješnijim modelima na zadatcima kampanja DUC-2002 i DUC-2004 kao i s prosječnim slaganjem između ljudskih označivača na tim zadatcima. Slaganje između označivača, naime, dobiva se računanjem mjera $ROUGE_N$ za par referentnih sažetaka.

Tablica 9.1: Uspješnost modela za automatsko sažimanje grupa dokumenata na ispitnom dijelu zbirke DUC-2002

Model	$ROUGE_1$	$ROUGE_2$
DUC-2002 najbolji (Sustav 21)	0,395	**0,103**
DUC-2002 medijan (Sustav 20)	0,365	0,086
DUC-2002 slaganje označivača	**0,418**	0,102
DOGAĐAJI-VSD	0,397	0,099
DOGAĐAJI-IRD	0,353	0,085
DOGAĐAJI-VSD + IRD	0,407	0,114
DOGAĐAJI-VSD + IRD + PAGERANK	**0,415**	**0,116**

Rasprava rezultata

Rezultati vrednovanja na ispitnom dijelu zbirke DUC-2002 dani su u tablici 9.1, dok su rezultati vrednovanja na zbirci DUC-2004 prikazani u tablici 9.2. Rezultati pokazuju da su modeli temeljeni na grafovima događaja uspješniji od najuspješnijih modela koji su sudjelovali u kampanjama vrednovanja DUC-2002 (sustav pod rednim brojem 21) i DUC-2004 (Conroy i dr., 2004) (iznimka je model DOGAĐAJI-IRD na ispitnom dijelu zbirke DUC-2002). Na ispitnom dijelu zbirke DUC-2002 model DOGAĐAJI-VSD, koji se oslanja isključivo na važnost sudionika događaja uspješniji je od modela DOGAĐAJI-IRD, koji se oslanja isključivo na informativnost riječi događaja. S druge strane, na zbirci DUC-2004 ova dva modela imaju gotovo istu uspješnost, zbog čega je teško dati konačan sud o tome koja je od dviju mjera značajnosti spominjanja događaja bolja. Međutim, model DOGAĐAJI-VSD + IRD koji kombinira te dvije mjere, značajno je uspješniji ($p < 0.05$; dvostrani t-test) od obaju modela koji koriste samo jednu od njih (DOGAĐAJI-VSD i DOGAĐAJI-IRD), što ukazuje na opravdanost kombiniranja tih mjera značajnosti spominjanja događaja. Konačno, potpuni model sažimanja DOGAĐAJI-VSD + IRD + PAGERANK, koji koristi zaglađivanje temeljem strukture grafova događaja numerički je uspješniji od modela DOGAĐAJI-VSD + IRD, koji ne koristi strukturno zaglađivanje algoritmom PageRank (razlika u uspješnosti značajna je uz $p < 0.05$ samo za mjeru $ROUGE_1$ na zbirci DUC-2004), što potvrđuje pretpostavku da se svojstvo značajnosti događaja za temu može propagirati na temelju vremenskih odnosa među događajima. Zanimljivo je primijetiti da je uspješnost potpunog modela vrlo blizu slaganja ljudskih označivača (pogotovo za ispitni dio zbirke DUC-2002), iz čega zaključujemo da nema puno prostora za daljnja poboljšanja predstavljenog modela za sažimanje grupa dokumenata.

U tablici 9.3 prikazani su sažeci izgrađeni automatski potpunim modelom DOGAĐAJI-VSD

Tablica 9.2: Uspješnost modela za automatsko sažimanje grupa dokumenata na zbirci DUC-2004

Model	$ROUGE_1$	$ROUGE_2$
DUC-2004 najbolji (Conroy i dr., 2004)	0,382	0,092
DUC-2004 temeljna metoda	0,324	0,064
DUC-2004 slaganje označivača	**0,440**	**0,134**
DOGAĐAJI-VSD	0,385	0,096
DOGAĐAJI-IRD	0,386	0,099
DOGAĐAJI-VSD + IRD	0,395	0,103
DOGAĐAJI-VSD + IRD + PAGERANK	**0,405**	**0,107**

+ IRD + PAGERANK zajedno s pripadnim referentnim sažecima za jednu grupu zbirke DUC-2002 i jednu grupu zbirke DUC-2004. Usporedbom automatski izgrađenih sažetaka s pripadnim referentnim sažecima otkrivamo kako među njima postoji značajna semantička podudarnost.

Tablica 9.3: Primjeri automatski izgrađenih sažetaka za grupe tematski povezanih dokumenata iz zbirki DUC-2002 i DUC-2004

Grupa	**Sažetak** (model DOGAĐAJI-VSD + IRD + PAGERANK)	**Referentni sažetak**
DUC-2002 d068f (200 pojavnica)	*Checkpoint Charlie, the Berlin Wall border post that symbolized the Cold War, was hoisted into history today. Checkpoint Charlie went up in 1961 in the middle of the Friedrichstrasse boulevard after Communist East Germany erected the Berlin Wall to choke off a flood of refugees to the enclave of West Berlin. Berlin has been split into Soviet, U.S., British and France sectors and West Berlin remains technically separate from West Germany. With the wall being dismantled daily in anticipation of German unification, U.S. officials decided to remove Checkpoint Charlie with a grand flourish. The Soviet Union said today that a united Germany can join NATO after a five-year transition period during which all Soviet and U.S. troops would leave the country. Since East Germany overthrew its Communist government last fall and the German borders were opened, Checkpoint Charlie has become as superfluous as the crumbling Berlin Wall. Baker, Soviet Foreign Minister Eduard Shevardnadze and the foreign ministers from France, Britain and the two Germanys each heralded the end of the checkpoint as a symbol of change. The German and Allied foreign ministers are meeting in East Berlin to discuss the status of the city and a united Germany's role in world affairs.*	*Checkpoint Charlie, the most famous symbol of the Cold War was lifted into history today. At an invitation only ceremony attended by the foreign ministers of the four World War II allies, a U.S. army band played an old Berlin song as the checkpoint was hoisted onto a waiting flatbed truck, to be taken to a museum. Speaking at the ceremony, Soviet Foreign Minister Eduard Shevardnadze, the first Soviet foreign minister to visit Berlin, called for the removal of Berlin's status as an allied controlled city and the withdrawal of troops six months after the two German states unify. Secretary of State James Baker, citing Checkpoint Charlie's status as a Cold War symbol, expressed the hope that its removal would bury the conflicts it created. The decision to remove Checkpoint Charlie came as large sections of the wall were being dismantled and the opening of the border between the two Berlins made the checkpoint superfluous. Its removal was timed to coincide with a meeting of German and Allied foreign ministers in East Berlin to discuss the status of Berlin and a united Germany. Among the invited guests was West Germany's Chancellor Willy Brandt who was mayor of Berlin in the 1960s.*
DUC-2004 d30003t (100 pojavnica)	*The Spanish and British governments appeared Wednesday to be seeking shelter from the political storm brewing over the possible extradition of former Chilean dictator Augusto Pinochet to Spain. British police arrested Pinochet in his bed Friday at a private London hospital in response to a request from Spain, which wants to question Pinochet about allegations of murder during the decade after he seized power in 1973. A delegation of Chilean legislators lobbying against the possible extradition of Augusto Pinochet to Spain to face trial, warned Thursday that Chile was on the brink of political turmoil.*	*Former Chilean dictator Augusto Pinochet has been arrested in London at the request of the Spanish government. Pinochet, in London for back surgery, was arrested in his hospital room. Spain is seeking extradition of Pinochet from London to Spain to face charges of murder in the deaths of Spanish citizens in Chile under Pinochet's rule in the 1970s and 80s. The arrest raised confusion in the international community as the legality of the move is debated. Pinochet supporters say that Pinochet's arrest is illegal, claiming he has diplomatic immunity. The final outcome of the extradition request lies with the Spanish courts.*

9.2 Pojednostavljivanje novinskih članaka temeljem događaja

Kao što je već spomenuto, novinski tekstovi osim događaja sadrže i rečenice koje opisuju pozadinu događaja, odnose među sudionicima i sl. Takve rečenice u pravilu nisu dio glavne informacije koju novinski članak prenosi (odnosno informacije koja je motivirala pisanje članka). Nadalje, neovisno o tome radi li se o rečenicama koje opisuju glavne događaje ili o rečenicama kojima se iznose pozadinske informacije, novinski je tekst često sintaktički vrlo složen i stilski obojen (Bell, 1991). Zbog svega navedenoga, mnogi ljudi (npr. djeca, neizvorni govornici, osobe s poteškoćama u čitanju) imaju poteškoća s razumijevanjem novinskog teksta.

Provedena su brojna istraživanja koja pokazuju da ljudima s niskom razinom pismenosti kao i ljudima s intelektualnim poteškoćama i ljudima s poteškoćama u čitanju (autistični ljudi, afazični ljudi, ljudi s urođenom gluhoćom i dr.) razumijevanje novinskih tekstova predstavlja poseban izazov (Carroll i dr., 1999; Devlin, 1999; Feng, 2009; Štajner i dr., 2012). S druge strane, sposobnost poimanja događanja u svijetu ili vlastitoj okolini, osobama s takvim poteškoćama važno je za integraciju u društvo (Freyhoff i dr., 1998). Stoga je vrlo važno učiniti novinske tekstove jednako dostupnima ljudima s kognitivnim poteškoćama i poteškoćama u čitanju. Razlikovanje između važnih i nevažnih informacija u novinskim člancima predstavlja poseban problem za osobe s kognitivnim poteškoćama (Pimperton i Nation, 2010). Nemogućnosti razlikovanja bitnoga od nebitnoga posebno doprinose složene rečenice u kojima postoje semantičke ovisnosti između podrečenica (Feng, 2009; Carretti i dr., 2010). Tako, primjerice, u rečenici

(34) *Philippines and China diplomatically resolved a tense naval standoff, the most dangerous confrontation between the two sides in recent years.*

cijela zavisna podrečenica "*the most dangerous confrontation between the two sides in recent years*" ima semantičku ulogu pojašnjenja sukoba te puno veću stilsku nego informacijsku vrijednost. Ključna informacija u prethodnom primjeru jest, naime, spominjanje događaja razrješavanja pomorskog sukoba između Kine i Filipina, sadržana u glavnoj surečenici "*Philippines and China diplomatically resolved a tense naval standoff*".

Zbog svega navedenoga, novinski tekst potrebno je pojednostaviti tako da se: (1) uklone rečenice koje ne nose ključne informacije, (2) rečenice pojednostave na način da svaka rečenica predstavlja isključivo jednu misao (npr. razbijanje zavisno složenih rečenica) te (3) uklone stilske figure i složeni vokabular. U ovom poglavlju predstavljamo modele za pojednostavljivanje teksta temeljem događaja koji ciljaju na prva dva načina smanjivanja složenosti: rečenice koje ne sadrže spominjanja događaja se izbacuju, a svako se spominjanje događaja predstavlja zasebnom rečenicom u pojednostavljenom tekstu. Kakvoću automatski pojednostavljenog teksta vrednujemo automatski mjerama čitljivosti teksta (engl. *readability metrics*) te ljudskim ocjenama kvalitete pojednostavljenjog teksta (gramatičnost, nepromijenjenost značenja, smanjenje suvišne informacije).

U nastavku su opisani modeli za automatizirano pojednostavljivanje teksta temeljeni na događajima (poglavlje 9.2.2) te postupci vrednovanja njihove uspješnosti (poglavlje 9.2.3). Ovi su modeli (i postupci njihova vrednovanja razvijeni) u suradnji sa Sveučilištem u Wolverhamptonu[2] te su objavljeni u radu (Glavaš i Štajner, 2013). Poglavlje započinje sažetim pregledom istraživanja u području pojednostavljivanja teksta.

9.2.1 Istraživanja u području pojednostavljivanja teksta

Nekoliko je važnih projekata bilo usmjereno na pojednostavljivanje tekstova za pojedine skupine ljudi s posebnim potrebama – za ljude s aleksijom[3] (Carroll i dr., 1999; Devlin i Unthank, 2006), ljude s kognitivnim poteškoćama (Saggion i dr., 2011), autistične ljude (Orasan i dr., 2013), ljude s urođenom gluhoćom (Inui i dr., 2003) te ljude s niskom razinom pismenosti (Aluísio i dr., 2008). Većina pristupa za automatizirano pojednostavljivanje teksta koji su razvijenih u okviru navedenih projekata temelje se na pravilima za leksičko i sintaktičko pojednostavljivanje rečenica. Sintaktičko pojednostavljivanje rečenica uobičajeno se provodi rekurzivnom primjenom skupa pravila (npr. rekurzivnim razlaganjem složenih rečenica na jednostavnije), često zanemarujući pritom semantičku interakciju koja prelazi granice rečenica. Leksičko pojednostavljivanje u pravilu se odnosi se na zamjenu "sofisticiranih" riječi jednostavnijim sinonimima (Lal i Ruger, 2002; Burstein i dr., 2007).

S obzirom na količinu pozadinskih informacija prisutnih u novinskim tekstovima, iznenađuje nedostatak radova usmjerenih na automatizirano pojednostavljivanje tekstova redukcijom nevažnog sadržaja. Iznimka je rad Drndarević i suradnika (2013), koji automatski uklanjaju izraze u zagradama. U istom je radu nemogućnost redukcije nebitnog sadržaja prepoznata kao glavni nedostatak automatiziranih pristupa pojednostavljivanju teksta i glavni razlog zbog kojeg je kakvoća automatski pojednostavljenih tekstova daleko od kakvoće ručno pojednostavljenih tekstova.

U ovom poglavlju predstavljeni su modeli koji tekst pojednostavljuju upravo uklanjajući informacijski manje bitan sadržaj. U skladu s hipotezom da događaji predstavljaju najvažniju informaciju u novinskim tekstovima (na kojoj je temeljen i model za sažimanje grupa dokumenata), predloženi modeli uklanjaju dijelove teksta koji nisu dio spominjanja događaja. Jedini iz literature poznat model koji koristi događaje za pojednostavljivanje teksta jest rad Barlacchija i Tonelli (2013), gdje se dječje priče na talijanskom jeziku pojednostavljuju na način da se uklanjaju argumenti događaja koji nisu nužni za gramatičku i semantičku ispravnost rečenica. Dječje su priče, međutim, početno već mnogo jednostavnije od novinskih tekstova.

[2] `http://clg.wlv.ac.uk/`

[3] Aleksija je neurološki poremećaj koji je karakteriziran nemogućnošću čitanja ili razumijevanja pisanih riječi.

9.2.2 Modeli za pojednostavljivanje teksta na temelju događaja

Osnovna je zamisao modela za pojednostavljivanje teksta temeljem događaja jednostavna: ukloniti sav tekstni sadržaj koji ne pripada spominjanjima događaja, bilo sidru ili bilo kojem od argumenata događaja. Na taj način (1) uklanjanjem nebitnog sadržaja smanjujemo složenost novinskog teksta i (2) skraćivanjem rečenica povećavamo čitljivost i razumljivost teksta. Pojednostavljivanje novinskih tekstova temeljimo na dvjema shemama: (1) pojednostavljivanju na razini rečenice te (2) pojednostavljivanju po događajima. Za razliku od algoritma za sažimanje grupa dokumenata, algoritmi za pojednostavljivanje teksta ne koriste vremensku strukturu grafova događaja, već samo pojedinačna spominjanja događaja.

Pojednostavljivanje na razini rečenice

Pojednostavljivanje na razini rečenice (engl. *sentence-wise simplification*) uklanja iz rečenice sve riječi koje nisu dio nekog spominjanja događaja (sidro ili dio nekog argumenta). Svakoj rečenici izvornog teksta koja sadrži barem jedno činjenično spominjanje događaja odgovara točno jedna rečenica u pojednostavljenom tekstu. Rečenice izvornog teksta u kojima nema niti jednog spominjanja događaja (npr. "*What a shame!*") nisu uključene u pojednostavljeni tekst. Pesudokod algoritma za pojedinačno pojednostavljivanje rečenica izvornog teksta prikazan je u Algoritmu 2. Prvo se iz rečenice crpe sva spominjanja događaja (redak 3 u pseudokodu algoritma), a potom se rečenica rastavlja na pojavnice (redak 6). Za svaku se pojavnicu provjerava je li dio nekog od spominjanja događaja pronađenih u rečenici (retci 7–10). Ukoliko pojavnica tvori neko spominjanje događaja, dodaje se u skup pojavnica pojednostavljene rečenice (redak 11).

Algoritam 2: Simpl-Sentence(s)

1: **Ulaz**: rečenica s
2: **Inicijalizacija**:
3: M – skup spominjanja događaja u rečenici s
4: **Pojednostavi**:
5: $S \leftarrow \emptyset$;
6: $T \leftarrow tokeniziraj(s)$
7: **za svaku** pojavnicu t iz T **učini**
8: **za svako** spominjanje događaja m iz M **učini**
9: $A \leftarrow sve\_pojavnice(m)$
10: **ako** $t \in A$ **učini**
11: $S \leftarrow S \cup t$
12: **Izlaz**: pojednostavljena rečenica S

Shema pojednostavljivanja na razini rečenice nema poseban mehanizam kojim se osigurava gramatičnost pojednostavljene rečenice. Stoga su pojednostavljene rečenice nastale uporabom

ove sheme nerijetko negramatične, najčešće uslijed izostavljanja veznika ili prijedloga, koji u pravilu nisu dio niti jednog događaja. Razmotrimo, primjerice, rečenicu:

(35) *China sent in its fleet and provoked Philippines.*

Pojednostavljivanjem na razini rečenice dobije se rečenica koja je negramatična uslijed izostavljanja veznika "*and*":

(36) *China sent in its fleet provoked Philippines.*

Pojednostavljivanje po događajima

Pojednostavljivanje po događajima (engl. *event-wise simplification*) pretvara svako činjenično spominjanje događaja izvornog teksta u zasebnu rečenicu pojednostavljenog teksta. Kako izrazi koji tvore argumente spominjanja događaja mogu istovremeno tvoriti više različitih spominjanja događaja (jer pojedini izraz može biti prepoznat kao argument za više sidara događaja; v. poglavlje 6.2), pojavnice takvih izraza pojavljivat će se u dvije ili više rečenica pojednostavljenog teksta. Tako će se, na primjer, pojavnica "*China*" u rečenici

(37) *China sent in its fleet and provoked Philippines.*

pojaviti u dvije rečenice pojednostavljenog teksta

(38) *China sent in its fleet. China provoked Philippines.*

budući da je "*China*" argument s ulogom AGENT i za događaj "*sent*" i za događaj "*provoked*". Budući da u rečenicama koje sadrže više spominjanja događaja nužno dolazi do usitnjavanja na više rečenica u pojednostavljenom tekstu, uvode se tri dodatna pravila koja služe očuvanju gramatičke ispravnosti pojednostavljenog teksta. Kao prvo, uklanjaju se spominjanja događaja semantičkog razreda REPORTING jer u pravilu ne mogu samostalno tvoriti gramatički ispravne rečenice (npr. "*Obama said*"). Kao drugo, spominjanja događaja čija su sidra imenice ne izdvajamo u posebne rečenice budući da u pravilu imaju malen broj argumenata te su često i sami argumenti drugih spominjanja događaja. Na primjer, u rečenici

(39) *China and Philippines resolved a naval standoff.*

imeničko spominjanje događaja "*standoff*" argument je s ulogom TARGET za spominjanje "*resolved*". Konačno, sidra događaja u gerundu koja su predikati zavisnih surečenica pretvaramo u prošlo vrijeme (engl. *past simple*). Tako će, primjerice, rečenica

(40) *Philippines disputed China's territorial claims, triggering the naval confrontation.*

u pojednostavljenom tekstu biti razložena na dvije rečenice:

(41) *Philippines disputed China's territorial claims. Philippines triggered the naval confrontation.*

pri čemu je izvorno sidro događaja "*triggering*" iz gerunda prebačeno u prošlo vrijeme ("*triggered*"). Pseudokod algoritma za pojednostavljivanje po događajima dan je u Algoritmu 3. Kao i kod algoritma pojednostavljivanja na razini rečenice, prvo se ekstrahiraju spominjanja događaja (redak 3 u pseudokodu) nakon čega se rečenica rastavlja na pojavnice (redak 6). Potom se za svaku pojavnicu t i svako spominjanje događaja m provjerava tvori li t spominjanje m (retci 9–13). Ukoliko je t dio događaja m i pritom događaj nije tipa REPORTING niti je sidro a događaja m imenica, pojavnicu t dodajemo u pojednostavljenu rečenicu za događaj m (retci 13–16), vodeći još pritom računa da pojavnicu prebacimo u prošlo vrijeme ukoliko je pojavnica sidro događaja u gerundu (redak 15).

Algoritam 3: Simpl-Event(s)

1: **Ulaz**: rečenica s
2: **Inicijalizacija**:
3: M – skup spominjanja događaja u rečenici s
4: **Pojednostavi**:
5: $\mathscr{S} \leftarrow \emptyset$;
6: $T \leftarrow tokeniziraj(s)$
7: **za svako** spominjanje događaja m iz M **učini**
8: $\mathscr{S}[m] \leftarrow \emptyset$
9: **za svaku** pojavnicu t iz T **učini**
10: **za svako** spominjanje događaja m iz M **učini**
11: $a \leftarrow sidro(m)$
12: $A \leftarrow sve\_pojavnice(m)$
13: **ako** $t \in A$ & $PoS(a) \neq$ 'N' & $type(m) \neq$ REPORTING **učini**
14: **ako** $t = a$ & $gerund(a)$ **učini**
15: $\mathscr{S}[m] \leftarrow \mathscr{S}[m] \cup past\_simple(a)$
16: **inače** $\mathscr{S}[m] \leftarrow \mathscr{S}[m] \cup t$
17: **Izlaz**: skup pojednostavljenih rečenica $\mathscr{S}$

Pojednostavljivanje po događajima s razrješavanjem zamjeničke anafore

Ljudi s poteškoćama u čitanju (u prvom redu ljudi s kognitivnim poteškoćama) imaju problema s uočavanjem anaforičkih veza između zamjenica i entiteta koje one predstavljaju (Ehrlich i dr., 1999; Shapiro i Milkes, 2004). U cilju analize utjecaja zamjenica na jednostavnost i razumljivost teksta, shema pojednostavljivanja po događajima nadograđuje se razrješavanjem zamjeničke anafore (engl. *pronominal anaphora resolution*). Koristeći alat za razrješavanje koreferencije entiteta programskog paketa StanfordCoreNLP (Lee i dr., 2011), umjesto zamjenica su u pojednostavljeni tekst uvršteni entiteti koje zamjenice zamjenjuju.

Tablica 9.4: Primjer automatski pojednostavljenoga teksta

Izvorni tekst
"*Baset al-Megrahi, the Libyan intelligence officer who was convicted in the 1988 Lockerbie bombing has died at his home in Tripoli, nearly three years after he was released from a Scottish prison.*"
Pojednostavljenje na razini rečenice
"*Baset al-Megrahi was convicted in the 1988 Lockerbie bombing has died at his home after he was released from a Scottish prison.*"
Pojednostavljenje po događajima
"*Baset al-Megrahi was convicted in the 1988 Lockerbie bombing. Baset al-Megrahi has died at his home. He was released from a Scottish prison.*"
Pojednostavljenje po događajima s razrješavanjem zamjenica
"*Baset al-Megrahi was convicted in the 1988 Lockerbie bombing. Baset al-Megrahi has died at his home. Baset al-Megrahi was released from a Scottish prison.*"

U tablici 9.4 dan je primjer odlomka novinskog teksta s rezultatima pojednostavljivanja (1) na razini rečenice, (2) po događajima i (3) po događajima s razrješavanjem zamjeničke anafore.

9.2.3 Eksperimentalno vrednovanje

Cilj pojednostavljivanja teksta jest povećanje čitljivosti i razumljivosti teksta. Kvalitetno pojednostavljeni tekst zadržava osnovno značenje izvornog teksta, smanjujući pritom količinu manje bitnih informacija. Prirodno, pojednostavljeni tekst mora biti gramatički ispravan jer negramatičnost samo dovodi do otežanog razumijevanja teksta. Uspješnost predloženih algoritama za automatsko pojednostavljivanje novinskih tekstova temeljem događaja vrednuje se na temelju analize sljedećih svojstava pojednostavljenog teksta:

1. Čitljivost (engl. *readability*) teksta odnosi se na jednostavne mjere složenosti teksta poput broja riječi, broja rečenica, prosječne duljine riječi i sl. Jednostavno rečeno, tekst je to čitljiviji što ima manje rečenica i što su rečenice i riječi od kojih se sastoji kraće. Naravno, kada bi čitljivost teksta bila jedini kriterij koji određuje kvalitetu pojednostavljenog teksta, onda bi svaki "pojednostavljeni tekst" bio prazan (tj. ne bi sadržavao niti jednu riječ);
2. *Gramatičnost* (engl. *grammaticality*) označava u kojoj je mjeri tekst gramatički ispravan. Pri tome se u prvom redu vodi računa o sintaktičkoj ispravnosti rečenica;

3. *Značaj sadržaja* (engl. *content relevance*) označava koliko su informacije sadržane u pojednostavljenom tekstu značajne u odnosu na sadržaj izvornog teksta. Značaj sadržaja uzima u obzir (1) mjeru u kojoj su glavne informacije izvornog teksta zadržane u pojednostavljenom tekstu (i to s nepromijenjenim značenjem) te (2) mjeru u kojoj su neznačajne informacije uklonjene iz izvornog teksta.

Dobar model za pojednostavljivanje teksta pronalazi pravi omjer između čitljivosti teksta s jedne, te gramatičnosti i značajnosti sadržaja teksta s druge strane. I dok se čitljivost teksta mjeri automatski na temelju skupa mjera koje uzimaju u obzir broj i duljine riječi i rečenica, gramatičnost i značaj sadržaja automatski pojednostavljenih tekstova ocjenjuju ljudi jer je te aspekte teksta vrlo teško ocijeniti automatski (Wubben i dr., 2012). Uspješnost predloženih modela uspoređuje se s uspješnošću sintaktički određene temeljne metode za pojednostavljivanje teksta. Temeljna metoda u pojednostavljenom tekstu zadržava samo glavne surečenice složenih rečenica, dok se sve zavisne surečenice odbacuju. Primjerice, ulaznu rečenicu

(42) *Pope Benedict XVI has handed over to church in Benin a pastoral guide which states his spiritual vision for Africa.*

temeljna bi metoda pojednostavnila u rečenicu

(43) *Pope Benedict XVI has handed over to church in Benin a pastoral guide.*

Za određivanje glavnih i zavisnih surečenica za temeljnu metodu pojednostavljivanja teksta korišten je parser strukture rečenice (engl. *phrase structure parser*) iz programskog paketa StanfordNLP (Klein i Manning, 2003).

Automatizirano vrednovanje čitljivosti

Za potrebe vrednovanja čitljivosti pojednostavljenih tekstova prikupljeno je 100 novinskih članaka putem internetskog servisa EMM NewsBrief. Za svaki od 100 članaka napravljena su četiri pojednostavljenja: (1) pojednostavljenje temeljnom metodom, (2) pojednostavljenje na razini rečenica (model P-REČENICA), (3) pojednostavljenje po događajima (model P-DOGAĐAJ) i (4) pojednostavljenje po događajima s razrješavanjem zamjeničke anafore (model P-DOGAĐAJ-Z). Čitljivost izvornog teksta i svih njegovih pojednostavljenih inačica mjerimo dvjema standardno korištenim mjerama, koje su izvedene empirijski na temelju više studija:

1. Kincaid-Fleschov razred (engl. *Kincaid-Flesch Grade Level*, KFL) (Kincaid i dr., 1975) računa se na temelju broja riječi w, broja slogova l te broja rečenica s:

$$KFL = 0,39 \cdot \frac{w}{s} + 11,8 \cdot \frac{l}{w} - 15,59 \tag{9.6}$$

2. Indeks SMOG (McLaughlin, 1969) računa se na temelju ukupnog broja rečenica s i broja

Tablica 9.5: Rezultati vrednovanja čitljivosti pojednostavljenih tekstova

Izvorni vs.	KFL	SMOG	DR	DČ
Tem. metoda	-27,7% ± 12,5%	-14,0% ± 8.0%	-38,5% ± 12,1%	-38,5% ± 12,1%
P-REČENICA	-30,1% ± 13,9%	-16,3% ± 9,2%	-44,3% ± 11,1%	-49,8% ± 11,5%
P-DOGAĐAJ	**-50,3% ± 12,6%**	**-30,8% ± 10,5%**	**-65,5% ± 9,3%**	**-63,4% ± 12,6%**
P-DOGAĐAJ-Z	-47,8% ± 13,9%	-29,4% ± 10,6%	-63,6% ± 10,3%	-61,2% ± 14,4%

višesložnih riječi (riječi koje imaju tri ili više slogova) p:

$$SMOG = 1,043 \cdot \sqrt{p \cdot \frac{30}{s}} + 3,1291 \tag{9.7}$$

Uz prethodne dvije mjere mjerimo i prosječnu duljinu rečenice (DR) te duljinu članka (DČ).

U tablici 9.5 čitljivost je pojednostavljenih tekstova prema navedenim mjerama izražena relativno u odnosu na čitljivost izvornog teksta, za svaku od metoda pojednostavljivanja (prikazani su prosjek i standardna devijacija na 100 članaka). Model P-DOGAĐAJ, koji svaki događaj pretvara u vlastitu rečenicu rezultira pojednostavljenim tekstovima koji su značajno čitljiviji od izvornih tekstova ($p < 0.01$; dvostrani t-test), kao i od pojednostavljenih tekstova proizvedenih modelom P-REČENICA, koji na razini rečenice reducira sadržaj koji ne pripada niti jednom od činjeničnih spominjanja događaja.

Ljudsko vrednovanje gramatičnosti i značaja sadržaja

Mjere čitljivosti analiziraju pojednostavljeni tekst na površnoj razini, ne ulazeći u semantiku pojednostavljenih tekstova. Gramatičnost i značaj informacija pojednostavljenih tekstova nije, nažalost, moguće kvalitetno procijeniti na automatiziran način, pa ta svojstava uobičajeno ocjenjuju ljudi (Knight i Marcu, 2002; Woodsend i Lapata, 2011). Sukladno tome, gramatičnost i značaj sadržaja pojednostavljenih tekstova i u ovim su eksperimentima označavali ljudi. S obzirom na mentalni napor potreban za procjenu gramatičnosti teksta, a pogotovo za usporedbu sadržaja pojednostavljenog i izvornog teksta, označivačima su dani parovi isječaka tekstova (isječak izvornog teksta koji se sastoji od jedne ili dvije rečenice uparen s odgovarajućim isječkom pojednostavljenog teksta), a ne parovi cijelih tekstova. Svakom pojednostavljenom tekstu označivači su dodjeljivali tri vrijednosti:

1. Ocjenu *gramatičnosti* pojednostavljenog teksta na ljestvici od 1 do 3, pri čemu ocjena 1 označava značajnu negramatičnost (npr. nedostajući subjekt kao u "*Was prevented by the Chinese surveillance craft*"), ocjena 2 označava manje gramatičke neispravnosti (npr. ne-

dostajući prijedlog ili veznik kao u primjeru "*Vessels blocked the arrest Chinese fishermen in disputed waters*"), dok ocjena 3 označava gramatički potpuno ispravnu rečenicu;

2. Ocjenu *značenja* koja (na ljestvici 1–3) označava stupanj do kojeg su bitne informacije iz izvornog teksta zadržane u izvornom značenju u pojednostavljenom tekstu. Pri tome ocjena 1 označava da bitne informacije većinom nisu zadržane u pojednostavljenom tekstu ili nisu zadržane u izvornom značenju (npr. "*Russians are tiring of Putin*" → Russians are tiring Putin), ocjena 2 označava da dio bitnih informacija nije zadržan (npr. "*The Kekhvi village was raided and the Rekha village burned*" → "*The Kekhvi village was raided*"), dok ocjena 3 označava da su sve bitne informacije zadržane;
3. Ocjenu *jednostavnosti* koja na ljestvici od 1 do 3 označava stupanj do kojeg su nebitne informacije uklonjene, pri čemu ocjena 1 označava da postoji mnogo nebitnih informacija i u pojednostavljenom tekstu (npr. "*The president, acting as commander in chief, landed in Afghanistan on Tuesday afternoon for an unannounced visit*"), ocjena 2 označava da je dio nebitnih informacija (ali ne sve) uklonjen (npr. "*The president landed in Afghanistan on Tuesday afternoon for an unannounced visit*"), dok ocjena 3 označava da pojednostavljeni tekst ne sadrži nebitne informacije (npr. "*The president landed in Afghanistan on Tuesday*").

Važno je uočiti da se mjere *značenja* i *jednostavnosti* mogu promatrati kao odziv odnosno preciznost značajnosti informacija koje se nalaze u pojednostavljenome tekstu – što je više nebitne informacije sadržano u pojednostavljenome tekstu (zadržane nebitne informacije predstavljaju "lažno pozitivne primjere"), to je *jednostavnost* (preciznost) manja. Također, što je manje bitne informacije zadržano (izostavljene bitne informacije predstavljaju "lažno negativne primjere") to je manji iznos mjere *značenja* (odziv). Kako uspješna metoda za pojednostavljivanje teksta treba zadržati bitne informacije u izvornom značenju, ali i ukloniti nebitne informacije, kvalitetu pojednostavljenja u konačnici mjerimo mjerom *značaja sadržaja* koja predstavlja harmonijsku sredinu mjera *značenja* i *jednostavnosti*.

Ukupno je slučajnim odabirom odabrano 70 isječaka novinskih tekstova koji su potom pojednostavljeni na četiri načina – svakim od tri modela temeljena na događajima (P-REČENICA, P-DOGAĐAJ, P-DOGAĐAJ-Z) te temeljnom metodom – što je rezultiralo s ukupno 280 pojednostavljenja izvornih tekstova. Tri označivača, iskusna u označavanja kvalitete pojednostavljenog teksta, početno su označili kvalitetu istih 40 pojednostavljenih tekstova. Nakon što je na tim tekstovima izmjerena zadovoljavajuća razina slaganja među označivačima, svaki je od tri označivača dodatno označio po 80 od preostalih 240 pojednostavljenih isječaka teksta. Korištene su tri komplementarne mjere slaganja među označivačima: otežana Cohenova kapa (κ) (engl. *weighted Cohen's Kappa*), Pearsonov koeficijent korelacije (r) te srednja apsolutna greška (engl. *mean absolute error*, MAE). Kao i obična Cohenova kapa (Cohen, 1960), otežana Cohenova kapa (Cohen, 1968) uzima u obzir slaganje uslijed slučajnosti, ali se koristi u

slučajevima kada nemaju sva neslaganja u oznakama između označivača jednaku težinu. Primjerice, kod ocjena na ljestvici od 1 do 3, neslaganje u kojem prvi označivač dodijeli ocjenu 1, a drugi ocjenu 3 smatra se većim od neslaganja kod kojeg prvi označivač dodijeli ocjenu 1, a drugi ocjenu 2. Neka je **X** matrica slaganja između dva označivača, pri čemu element te matrice, x_{ij}, označava broj primjera kojima je prvi označivač dodijelio i-tu oznaku, a drugi označivač j-tu oznaku te neka je **W** simetrična matrica koja dodjeljuje težine pojedinim vrstama (ne)slaganja (konkretno $w_{1,2} = 1$ i $w_{1,3} = 2$). Neka je matrica **M** matrica koja definira slučajno (ne)slaganje, odnosno matrica čiji element m_{ij} označava broj primjera koji bi slučajnim označavanjem (na temelju ukupnih distribucija oznaka dvaju označivača) bili označeni i-tom oznakom od strane prvog označivača i j-tom oznakom od strane drugog označivača. Razmotrimo sljedeći primjer: neka je od ukupno 30 primjera prvi označivač označio 10 primjera oznakom "1", 10 primjera oznakom "2" te 10 primjera oznakom "3", a drugi označivač 5 primjera oznakom "1", 20 primjera oznakom "2" te 5 primjera oznakom "3". Tada $m_{1,2} = \frac{10}{30} \cdot \frac{20}{30} \cdot 30 = \frac{20}{3}$ jer apriorna vjerojatnost da prvi označivač dodijeli oznaku "1" iznosi $\frac{10}{30}$, a apriorna vjerojatnost da drugi označivač dodijeli oznaku "2" iznosi $\frac{20}{30}$. Iznos otežane Cohenove kape na temelju matrica **X**, **W** i **M** računamo na sljedeći način:

$$\kappa = 1 - \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij} \cdot x_{ij}}{\sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij} \cdot m_{ij}}, \tag{9.8}$$

gdje je K broj različitih oznaka koje označivači dodjeljuju primjerima. Pearsonov koeficijent korelacije računa linearnu zavisnost između oznaka dvaju označivača promatrajući oznake jednog označivača kao vrijednosti slučajne varijable. Neka je X skup svih oznaka prvog označivača, Y skup svih oznaka drugog označivača, $\bar{x}$ neka je prosjek svih oznaka prvog označivača te $\bar{y}$ prosjek svih oznaka drugog označivača. Pearsonov koeficijent korelacije onda se računa na sljedeći način:

$$r = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2}}, \tag{9.9}$$

gdje je N broj primjera za označavanje, a x_i i y_i predstavljaju oznake prvog odnosno drugog označivača za i-ti primjer. Konačno, srednja apsolutna pogreška (MAE) prosječna je vrijednost razlike u oznakama između označivača nad svim primjerima:

$$MAE = \frac{\sum_{i=1}^{N} |x_i - y_i|}{N}. \tag{9.10}$$

Slaganje između označivača (uprosječene vrijednosti za tri para označivača) za ocjene *gramatičnosti*, *značenja* i *jednostavnosti* pojednostavljenih tekstova prikazano je u tablici 9.6. Primjećujemo da je slaganje označivača za *gramatičnost* veće nego slaganje za *značenje* i *jednostavnost* iz čega zaključujemo da je pojam gramatičnosti manje podložan individualnoj inter-

Tablica 9.6: Slaganje označivača u ocjenama *gramatičnosti*, *značenja* i *jednostavnosti* pojednostavljenih tekstova

Svojstvo	κ	r	MAE
gramatičnost	0,68	0,77	0,18
značenje	0,53	0,67	0,37
jednostavnost	0,54	0,60	0,28

Tablica 9.7: Uspješnost postupaka za pojednostavljivanje novinskih tekstova temeljenih na događajima izražena ocjenama *gramatičnosti* i *značaja sadržaja*

Model	Gramatičnost (1–3)	Značaj sadržaja (1–3)
Temeljna metoda	2,57 ± 0,79	1,90 ± 0,64
P-REČENICA	1,98 ± 0,80	2,12 ± 0,61
P-DOGAĐAJ	**2,70 ± 0,52**	2,30 ± 0,54
P-DOGAĐAJ-Z	2,68 ± 0,56	**2,39 ± 0,57**

petaciji od toga što je bitna, a što nebitna informacija u nekom tekstu. Iako lošija od slaganja za svojstvo *gramatičnosti*, i slaganja za svojstva *značenja* i *jednostavnosti* zadovoljavajuća su ($\kappa > 0.5$, $r \geq 0.6$, MAE < 0.4).

Na posljetku, u tablici 9.7 prikazana je uspješnost predloženih modela automatskog pojednostavljivanja teksta temeljenih na događajima (zajedno s rezultatima temeljne metode) vrednovana od strane ljudi na trima prethodno opisanim svojstvima teksta (*gramatičnost*, *značenje*, *jednostavnost*). U tablici 9.7 su umjesto zasebnih rezultata za *značenje* i *jednostavnost* prikazani rezultati za već spomenutu mjeru *značaja sadržaja* koja predstavlja harmonijsku sredinu mjera *jednostavnosti* i *značenja*. Sva tri modela pojednostavljivanja temeljena na događajima produciraju tekst koji ima značajniji sadržaj od teksta nastalog temeljnom metodom pojednostavljivanja ($p < 0.05$ za model P-REČENICA i $p < 0.01$ za modele P-DOGAĐAJ i P-DOGAĐAJ-Z; značajnost ispitana dvostranim t-testom). Model P-REČENICA, koji pojednostavljivanje vrši na razini rečenice, međutim, stvara tekstove koji su značajno manje gramatični od tekstova koje stvara temeljna metoda. Pregledom pojednostavljenih tekstova dobivenih tom metodom koji su ocjenjeni kao negramatični uočeno je kako se u većini slučajeva, očekivano, radi o izostavljanju prijedloga i veznika koji povezuju podrečenice koje odgovaraju dvama spominjanjima događaja (v. primjer 36). Do izbacivanja takvih veznika i prijedloga, koje uzrokuje negramatičnost,

dolazi zbog toga što takvi prijedlozi i veznici u pravilu nisu dio spominjanja događaja. Modeli P-DOGAĐAJ i P-DOGAĐAJ-Z, koji za svaki događaj generiraju zasebnu pojednostavljenu rečenicu, uspješniji su od modela P-REČENICA ($p < 0.01$) i po značaju sadržaja i po gramatičnosti pojednostavljenih tekstova koje generiraju. Tekstovi dobiveni modelom P-DOGAĐAJ koji imaju niske ocjene značajnosti sadržaja u pravilu su posljedica pogrešaka u crpljenju argumenata događaja. Primjerice, zbog pogreške u crpljenju argumenata za događaje "*killed*" i "*ousted*" u rečenici

(44) *Nearly 3,000 soldiers have been killed in Afghanistan since the Talibans were ousted in 2001.*

dobivamo pojednostavljenu rečenicu kojoj je značenje promijenjeno u odnosu na izvornu rečenicu:

(45) *Nearly 3,000 soldiers have been killed in Afghanistan in 2001.*

jer su izrazi "*in Afghanistan*" i "*in 2001*" pogrešno prepoznati kao argumenti događaja "*killed*" umjesto kao argumenti događaja "*ousted*". Model P-DOGAĐAJ-Z koji dodatno koristi razrješavanje zamjenica povećava *značaj sadržaja* pojednostavljenih tekstova, iako razlika u odnosu na model bez razrješavanja zamjenica (P-DOGAĐAJ-Z) nije statistički značajna.

Konačno, zaključujemo kako su modeli koji pojednostavljuju tekst na temelju događaja na način da svaki događaj predstavljaju zasebnom rečenicom (modeli P-DOGAĐAJ i P-DOGAĐAJ-Z) izuzetno pogodni za pojednostavljivanje složenih novinskih tekstova budući da generiraju tekstove koji su značajno čitljiviji od početnih tekstova, a uz to su i gramatični te zadržavaju bitne, a uklanjaju nebitne informacije izvornih tekstova.

Poglavlje 10

Zaključak

Događaji su sve ono što se ostvaruje ili odvija u vremenu. Osim samom vrstom radnje, događaji su određeni entitetima koji u njima sudjeluju te vremenom i mjestom svoga odvijanja. S jedne strane, količina tekstnih izvora koji opisuju događaje iz stvarnog svijeta u stalnom je porastu. S druge su strane sve izraženije informacijske potrebe korisnika koje su usmjerene na događaje. Stoga su potrebe za automatiziranim crpljenjem informacija o događajima, učinkovitim pronalaženjem informacija o događajima te općenito automatiziranom analizom događaja danas izraženije nego ikad, s mogućim primjenama u medijskoj analitici, istraživačkom novinarstvu, obavještajnim djelatnostima, povijesnim istraživanjima i dr. Učinkovito zadovoljavanje informacijskih potreba korisnika koje su usmjerene na događaje zahtijeva precizno crpljenje informacija o događajima iz teksta, što je vrlo zahtjevan zadatak s obzirom na složenost, nepreciznost i višeznačnost prirodnog jezika.

Događaji stvarnog svijeta u tekstu su predstavljeni spominjanjima događaja, a tekstom su često naznačeni i različiti odnosi između događaja. I samo spominjanje događaja ima određenu strukturu budući da se sastoji od sidra događaja kao riječi koja događaj iz stvarnog svijeta ukorjenjuje u tekst te argumenata događaja kao izraza koji opisuju sudionike događaja i okolnosti odvijanja radnje (npr. lokacija, vrijeme). Stoga tekstovi poput novinskih članaka, koji se u prvom redu bave događajima iz stvarnog svijeta, zapravo određuju strukturu koja međusobno povezuje spominjanja događaja kao što povezuje i sidra pojedinih događaja s njihovim argumentima. Iako su događaji središnji izvor informacija u mnogim tekstovima (npr. novinski članci, policijski izvještaji, biografije, bolnički zapisi), događajima je u području crpljenja informacija posvećeno manje pažnje (u odnosu na, primjerice, imenovane entitete), dok su u području pretraživanja informacija događaji gotovo potpuno zanemareni.

U okviru ove disertacije razvijen je model *grafa događaja* kao strukture koja zadržava sve bitne informacijske aspekte događaja iz stvarnog svijeta. Čvorovi grafa događaja predstavljaju spominjanja događaja (strukturu koja povezuje sidro događaja i argumente događaja), dok bridovi mogu predstavljati različite semantičke odnose između događaja. Istraživanje provedeno u

okviru disertacije bilo je usmjereno na vremenske odnose između događaja. Na temelju modela grafa događaja ostvaren je potpuno automatizirani postupak za crpljenje grafova događaja iz teksta koji kombinira modele za crpljenje informacija temeljene na nadziranom strojnom učenju s modelima temeljenima na pravilima. Ostvareni postupak za izgradnju grafova događaja iz tekstnih dokumenata uključuje (1) model nadziranog strojnog učenja za crpljenje sidara događaja, (2) model temeljen na pravilima za crpljenje argumenata događaja, (3) model nadziranog strojnog učenja za crpljenje vremenskih odnosa među događajima te (4) model za razrješavanje koreferentnih spominjanja događaja. Svaki od četiri navedena modela crpljenja informacija temeljito je vrednovan uobičajeno korištenim mjerama preciznosti i odziva. Pored toga, u disertaciji su predložene i empirijski provjerene dvije nove mjere kojima se ocjenjuje kakvoća cjelokupnog postupka za izgradnju grafova događaja, odnosno kakvoća automatski izgrađenih grafova događaja.

Za potrebe izgradnje i vrednovanja pojedinačnih modela za crpljenje informacija izrađena je velika zbirka novinskih članaka EVEXTRA s ručno označenim činjeničnim spominjanjima događaja. U cilju vrednovanja ukupne kakvoće izgrađenih grafova događaja, dio dokumenata zbirke EVEXTRA ručno je označen cjelovitim grafovima događaja. Zbirka EVEXTRA trostruko je veća od zbirke TimeBank (Pustejovsky i dr., 2003a), koja se do sada koristila u istraživanjima u području crpljenja informacija o događajima.

U cilju provjere jedne od glavnih hipoteza istraživanja prema kojoj se usporedbom grafova događaja mogu pronaći dokumenti koji opisuju iste događaje iz stvarnog svijeta, predložena je inovativna metoda za učinkovitu usporedbu grafova događaja temeljena na umnošku grafova te jezgrenim funkcijama nad grafovima kao učinkovita alternativa tradicionalnim postupcima usporedbe grafova poput pronalaženja izmorfnih podgrafova. Dvije vrste jezgrenih funkcija – jezgrena funkcija umnoška grafova i jezgrena funkcija težinske dekompozicije grafa – proširene su na način da uzimaju u obzir semantiku događaja odnosno uključuju model za prepoznavanje koreferentnih spominjanja događaja kao funkciju podudarnosti vrhova grafova.

Središnja hipoteza disertacije prema kojoj je izgradnjom grafova događaja iz tekstnih podataka te njihovom usporedbom jezgrenim funkcijama nad grafovima moguće ostvariti učinkovit postupak pretraživanja informacija provjerena je ekstrinzičnim vrednovanjem na nekoliko zadataka pretraživanja informacija. Eksperimentalni rezultati pokazuju kako su postupci pretraživanja informacija temeljeni na grafovima događaja i jezgrenim funkcijama nad grafovima učinkovitiji od tradicionalnih modela pretraživanja informacija koji ne strukturiraju informacije o događajima.

Korisnost strukturiranja informacija o događajima dodatno je potvrđena na zadatcima sažimanja i pojednostavljivanja novinskih tekstova. Razvijen je nov algoritam za sažimanje grupa dokumenata temeljen na grafovima događaja koji se pokazao uspješnijim od etabliranih postupaka sažimanja koji nisu usmjereni na događaje. Nadalje, ljudskim vrednovanjem kakvoće

pojednostavljenih tekstova utvrđeno je kako postupak pojednostavljivanja novinskih članaka temeljen na događajima stvara pojednostavljene tekstove koji su gramatični te zadržavaju samo bitne informacije izvornog teksta.

Istraživanje opisano u disertaciji bilo je usmjereno na engleski jezik. Premda je formalizam grafa događaja jezično nezavisan, modeli za crpljenje informacija uključeni u postupak izgradnje grafova događaja u nekim su svojim dijelovima posebno prilagođeni engleskome jeziku. Prilagodba postupka izgradnje grafova događaja za neki drugi jezik (npr. hrvatski) uključivala bi označavanje tekstnih zbirki na ciljnome jeziku spominjanjima događaja (sidra i argumenti) i vremenskim odnosima među događajima kako bi se omogućila izgradnja odgovarajućih modela nadziranog strojnog učenja. Nadalje, sintaktička pravila za crpljenje argumenata događaja trebalo bi prilagoditi gramatici novog jezika, a za ciljni jezik trebali bi biti razvijeni i resursi poput leksičko-semantičke mreže WordNet, temeljni alati za predobradu teksta (razdvajanje rečenica i pojavnica, označavanje vrsta riječi, ovisnosno parsanje) te napredniji alati kao što su alat za prepoznavanje imenovanih entiteta te razrješavanje koreferencije entiteta. S obzirom da za hrvatski jezik postoje temeljni alati za predobradu teksta (Šarić i dr., 2012; Šnajder i dr., 2008), a odnedavno i ovisnosni parser (Agić, 2012a) te pouzdan alat za prepoznavanje imenovanih entiteta (Glavaš i dr., 2012), jedan od smjerova budućeg rada bit će prilagodba postupka za izgradnju grafova događaja tekstovima na hrvatskome jeziku.

Istraživanje obuhvaćeno disertacijom postavlja temelje za analizu dokumenata kroz prizmu događaja te otvara mnoge pravce za buduća istraživanja. U nastavku su razmotreni neki od mogućih smjerova za daljnje istraživanje. Jedna očigledna mogućnost za buduća istraživanja jest unaprjeđivanje grafova događaja, kako u konceptualnom, tako i u izvedbenom pogledu. Izvedbeno, budućim bi istraživanjem trebalo unaprijediti pojedinačne modele koji sudjeluju u izgradnji grafova događaja, pri čemu, sukladno analizi prostora za poboljšanja (v. odjeljak 7.3.2), najviše smisla ima unaprjeđivati model za crpljenje vremenskih odnosa među događajima. U konceptualnom smislu, uz vremenske odnose, grafovima događaja moguće je obuhvatiti i mnoštvo drugih semantičkih odnosa među događajima, poput kauzalnosti i odnosa prostorno-vremenskog sadržavanja. Dodavanje takvih semantički bogatih odnosa u grafove događaja moglo bi dovesti do daljnjeg poboljšanja postupaka za pretraživanje informacija temeljenih na grafovima događaja. Začetak istraživanja u tom smjeru predstavlja rad (Glavaš i dr., 2014), u kojem se dokumenti strukturiraju u obliku stabala u kojima su događaji hijerarhijski povezani odnosom prostorno-vremenskog sadržavanja (engl. *spatio-temporal containment*).

Drugi smjer budućeg istraživanja razmatrat će primjenu grafova događaja na drugim prikladnim zadatcima obrade prirodnog jezika, u prvom redu u području automatiziranog odgovaranja na pitanja, gdje se odgovori na pitanja usmjerena na događaje mogu pronaći među argumentima događaja i vremenskim odnosima među spominjanjima događaja. Činjenica da su dokumenti koji opisuju događaje predstavljeni grafovima omogućava analizu događaja stvarnog svijeta

korištenjem širokog spektra postojećih alata i algoritama za analizu grafova (jedan konkretan primjer čega su i jezgrene funkcije nad grafovima obuhvaćene ovom disertacijom). Osim na različitim zadatcima, grafove događaja moguće je primijeniti i u drugim tekstnim žanrovima i domenama. Zanimljiva bi mogla biti primjena grafova događaja u analizi biografija, s obzirom da su biografije, opisujući ključne događaje nečijeg života, također prvenstveno određene događajima.

Konačno, posebno zanimljiv smjer daljnjeg istraživanja bavio bi se sustavnim razmatranjem događaja na svim razinama apstrakcije i zrnatosti te automatiziranim zaključivanjem (tj. izvođenjem novog znanja) na temelju grafova događaja. Iako grafovi događaja već povezuju događaje na strukturiran način, u trenutnom obliku ne uvažavaju činjenicu da pojedinačni događaji mogu biti na različitim razinama apstrakcije i zrnatosti (npr. jedan događaj može u potpunosti sadržavati drugi). Nadalje, graf događaja, jednom kad je izgrađen, statička je struktura. Drugim riječima, trenutno ne postoje mehanizmi kojima bi se transformacijom postojećih grafova događaja ili kombiniranjem grafova događaja iz više različitih dokumenata izvelo novo znanje. Srž budućeg istraživanja bio bi formalni radni okvir kojim bi se događaji modelirali u kontinuiranom prostoru – od atomskih događaja (tj. događaja koji se ne mogu razdijeliti na manje događaje) do tema (tj. grupa povezanih događaja visoke razine). Takvo modeliranje temeljilo bi se na iscrpnom skupu operatora kojima bi se grafovi događaja prilagođavali razinama apstrakcije i zrnatosti koje određuje konkretna primjena. Opisani radni okvir omogućio bi da se pojedinačni događaji i teme formalno analiziraju na unificirani način.

Dio IV

Dodatci

Dodatak A

Programska izvedba

U ovom poglavlju opisane su programska izvedba postupka za izgradnju grafova događaja te programska izvedba jezgrenih funkcija nad grafovima. Programska izvedba ostvarena je u programskom jeziku C#, odnosno koristeći razvojnu platformu .NET. Programski jezik C# (uz Javu i Python) jedan je od najčešće korištenih objektno-orijentiranih programskih jezika današnjice.

Programski jezik C# ima iznimno izražajnu sintaksu te uz uobičajene koncepte objektno-orijentiranog programiranja kao što su enkapsulacija, sučelja, nasljeđivanje i polimorfizam podržava i napredne konstrukte poput nulabilnih tipova (engl. *nullable types*), delegata (engl. *delegates*), pobrojanih tipova (engl. *enumerations*) i lambda-izraza koji nisu podržani u srodnim programskih jezicima kao što su C++ i Java. Programi pisani programskim jezikom C# izvršavaju se na platformi .NET, koja je bitna sastavnica operacijskog sustava Windows, a koja se sastoji od zajedničkog izvršnog okruženja (engl. *common language runtime*, CLR) i velikog skupa programskih knjižnica potrebnih za česte zadatke (npr. programska knjižnica za upravljanje pisanjem u datotečni sustav i čitanjem iz datotečnog sustava).

U nastavku su ukratko opisana osnovna svojstva programskog jezika C# koja su obilato korištena u programskoj izvedbi. Programska izvedba organizirana je u programske knjižnice koje su opisane u drugom dijelu ovog dodatka.

A.1 Programski jezik C#

Programski jezik C#[1] moderan je objektno-orijentirani programski jezik čiji je razvoj započeo 1999. godine pod kodnim imenom COOL[2], kao Microsoftov odgovor na nedostatke objektno-orijentiranih jezika C++ i Java. Iako je tijekom ranog razvoja bio vrlo sličan Javi, programski

[1] Postoje dvije jednako zastupljene teorije o značenju znaka '#' u nazivu. Prema prvoj teoriji, taj znak predstavlja četiri znaka '+', čime se želi naznačiti da je jezik bolji od programskog jezika C++ (koji ima samo dva plusa). Prema drugoj teoriji, znak ima značenje glazbene povisilice, čime se sugerira da se jezik nalazi "pola tona" iznad drugih programskih jezika (Nagel i dr., 2012).

[2] Od *C-Like Object-Oriented Language*

se jezik C# uvođenjem generičkih tipova (tj. načinom njihove implementacije) bitno udaljio od Jave, a razlike su postale još izraženije kada je Microsoft 2007. godine izdao inačicu jezika C#3.0 s podrškom za koncepte iz funkcijske paradigme kao što su lambda-izrazi, metode za proširenje tipova i anonimni tipovi. Temeljna svojstva programskog jezika C# su sljedeća:

- C# uključuje koncepte iz više programskih paradigmi (engl. *multi-paradigm programming language*). Iako je u osnovi objektno-orijentiran jezik, C# sadrži i elemente drugih programskih paradigmi, poput funkcijske paradigme (npr. delegati) ili deklarativne paradigme (npr. deklarativni upiti LINQ);
- C# je strogo tipiziran (engl. *strongly typed*) programski jezik što znači da sve vrijednosti imaju određene tipove, odnosno da metode razreda (funkcije u objektno-orijentiranoj paradigmi) imaju određene tipove argumenata koje primaju kao i tipove vrijednosti koje vraćaju. Pozivajuća funkcija tada funkciji koju poziva mora predati argumente koji odgovaraju tipovima deklariranim funkcijom koja se poziva. Primjerice, funkcija `string func(int i1, int i2)` na ulazu prima dva cijela broja (tip `int`) te po izvršavanju kao rezultat daje znakovni niz (tip `string`). Programski jezik C# oslanja se na jedinstveni sustav tipova platforme .NET (engl. *Common Type System*, CTS), kojim su određene definicije temeljnih tipova i način njihova predstavljanja u radnoj memoriji;
- Kako je C# u prvom redu objektno-orijentiran jezik, tipski sustav podržava nasljeđivanje (engl. *inheritance*), sučelja (engl. *interfaces*) i polimorfizam. Nasljeđivanje podrazumijeva tzv. "is-a" odnos između razreda, tj. odnos u kojem jedan razred predstavlja specijalizaciju nekog općenitijeg razreda (npr. razred `Kvadrat` može biti specijalizacija razreda `Pravokutnik`). Hijerarhije nasljeđivanja među razredima u jeziku C# mogu biti proizvoljne dubine. Polimorfizam se u objektno-orijentiranim jezicima odnosi na različitu izvedbu istoimene metode (isti naziv metode, isti ulazni i izlazni tipovi) u različitim razredima, pri čemu odluku o tome koja se izvedba izvršava (tj. izvedba u kojem od razreda) dinamički donosi izvršavatelj tijekom izvođenja programa. Polimorfizam se može ostvariti na dva načina: nasljeđivanjem među razredima te definiranjem sučelja. Kod nasljeđivanja, razredi mogu nadjačati (engl. *override*) izvedbu neke metode iz roditeljskog razreda na način da definiraju vlastitu izvedbu istoimene metode. Pri tome metode u roditeljskom razredu mogu biti samo deklarirane, odnosno ne moraju imati konkretnu izvedbu (tzv. apstraktne metode). Primjerice, razred `GeometrijskiLik` može imati apstraktnu metodu `Povrsina` koju onda razredi koji nasljeđuju taj razred (npr. `Krug`, `Pravokutnik`, `Kvadrat`) nadjačaju svojim izvedbama istoimene metode. Nasljeđivanje se uobičajeno koristi kada razredi osim funkcionalnosti od roditeljskog razreda nasljeđuju i statičke podatke u vidu članskih varijabli (engl. *member variables*). Kada za neki skup razreda postoji samo zajednička funkcionalnost (ne i zajedničke članske varijable), tada je za ostvarivanje polimorfizma uputnije koristiti sučelja. Sučelja nisu ništa drugo nego

skupovi deklaracija funkcija. Da bi neki razred zadovoljio sučelje, mora sadržavati konkretne izvedbe svih funkcija deklariranih sučeljem;

- Tipski sustav jezika C# podržava i preotpere ćivanje metoda (engl. *overloading*) što znači da se za pojedini razred može definirati više metoda istog naziva, koje se, međutim, razlikuju u broju ulaznih argumenata koje primaju ili se pak razlikuju u tipovima tih argumenata. Na taj je način za različite tipove podataka (među kojima nema nasljeđivanja) moguće definirati slične operacije. Na primjer, razred `Kvadrat` može sadržavati metode `Povrsina(int a` i `Povrsina(string a)` gdje je u prvoj metodi duljina stranice kvadrata zadana cijelim brojem, a u drugoj znakovnim nizom (npr. "2 cm");
- Programski jezik C# omogućava jezično uklopljeno postavljanje upita (engl. *Language-Integrated Query*, LINQ) nad različitim strukturama podataka. LINQ-upiti pripadaju deklarativnoj programskoj paradigmi jer omogućavaju da se definira što se želi izvući iz podatkovnog izvora, a da se ne specificira način na koji se dohvat treba izvesti. LINQ upiti imaju jedinstvenu sintaksu za sve vrste podatkovnih izvora, dakle neovisno o tome radi li se o dohvatu podataka iz liste u radnoj memoriji ili iz baze podataka;
- U jeziku C# podržani su i lambda-izrazi koji su uobičajeni konstrukt funkcijske programske paradigme. Lambda-izrazi anonimne su funkcije koje se koriste za izgradnju delegata (referenci na funkcije) ili stabala izraza. Korištenjem lambda-izraza moguće je funkcije stvarati lokalno te ih koristiti kao argumente ili rezultantne vrijednosti drugih funkcija. Lambda-izrazi posebno su korisni pri pisanju LINQ upita. Na primjer, u LINQ upitu `lista.Where(x => x.Name == "Goran")` lambda-izraz `x => x.Name == "Goran"` određuje anonimnu funkciju koja kao ulaz prima podatak koji ima tip elementa liste, a na izlazu daje logičku vrijednost (tip `bool`) koja je jednaka istini (vrijednost `true`) samo ako je vrijednost svojstva `Name` elementa jednaka "Goran". Sukladno tome, rezultat danog LINQ upita sadržavat će samo one elemente izvorne liste za koje je anonimna funkcija definirana lambda-izrazom vratila vrijednost `true`.

Popularnosti platforme .NET i programskog jezika C# zasigurno je pridonijelo i izvrsno razvojno okruženje Visual Studio koje uključuje različita pomagala koja omogućavaju učinkovito programiranje i olakšavaju razvojni proces – uređivač izvornog kôda s inteligentnim nadopunjavanjem (engl. *intelligent code completion*) IntelliSense, alat za pronalaženje pogrešaka (engl. *debugger*), alat za analizu i optimizaciju trajanja izvršavanja programa i dr.

A.2 Programske knjižnice

Programski kôd razvijen u okviru disertacije organiziran je u knjižnice od kojih su najvažnije: (1) knjižnica `NLPCommonCode`, koja sadrži programski kôd za temeljnu obradu prirodnog jezika te pristup jezičnim resursima, (2) knjižnica `MachineLearningLib`, koja sadrži pro-

gramski kôd vezan za algoritme strojnog učenja i vrednovanje te (3) knjižnica `EventExtraction.Core`, u kojoj se nalazi izvedba postupka za izgradnju grafova događaja te metode za računanje sličnosti između grafova događaja. Cjelokupno programsko rješenje sadrži još nekoliko knjižnica – `EventExtraction.Crawl` za prikupljanje novinskih članaka s interneta, `EventExtraction.DataAccess` za pohranjivanje novinskih članaka u bazu podataka te `EventExtraction.GUI` u kojoj se nalazi izvedba alata za označavanje teksta.

A.2.1 Knjižnica `NLPCommonCode`

U ovoj se knjižnici nalazi izvedba raznih alata za obradu prirodnog jezika, od temeljnih alata za predobradu teksta (rastavljanje na rečenice i pojavnice, morfološka normalizacija, označavanje vrsta riječi) do parsera, alata za prepoznavanje imenovanih entiteta te alata za razrješavanje koreferencije entiteta. Programski kôd svih navedenih alata preuzet je iz knjižnice StanfordCoreNLP[3] izvorno pisane u Javi, koja je korištenjem alata IKVM[4] prevedena u oblik u kojem se može koristiti u okviru platforme .NET. Programski kôd u ovoj biblioteci organiziran je po mapama od kojih su najvažnije detaljnije opisane u nastavku.

Mapa `Preprocessing`

Mapa `Preprocessing` sadrži razrede koji se tiču temeljne predobrade teksta: razdvajanja teksta na rečenice, razdvajanja rečenica na pojavnice te morfološke normalizacije (lematizacije i korjenovanja) riječi. Razred `SentenceSplitter` omotač je oko razreda za razdvajanje rečenica iz knjižnice `StanfordCoreNLP` u kojem je najvažnija metoda `Split` kojom se zadani tekst rastavlja na rečenice (rezultat je lista objekata razreda `Sentence`).

```
public List<Sentence> Split(string text)
{
  Annotation document = new Annotation(text);
  stanfordPipeline.annotate(document);
  ArrayList sentences = (ArrayList)document.get(typeof(CoreAnnotations.SentencesAnnotation));
  List<Sentence> textSentences = new List<Sentence>();
  for (int i = 0; i < sentences.size(); i++)
  {
    string sentText = sentences.get(i).ToString();
    int startOffset = sentences.get(i).getOffset().intValue();
    textSentences.Add(new Sentence { Text = sentText, StartPosition = startOffset });
  }
  return textSentences;
```

[3] `http://nlp.stanford.edu/software/corenlp.shtml`
[4] `http://www.ikvm.net`

```
}
```

Razred `Tokenizer` omotač je oko razreda `PTBTokenizer` iz knjižnice `StanfordCoreNLP`, koji služi razdvajanju teksta na pojavnice. Najvažnija metoda razreda `Tokenizer` jest metoda `TokenizeText`, koja zadani tekst rastavlja na pojavnice te kao rezultat vraća listu objekata razreda `Token`.

```
public List<Token> TokenizeText(string text)
{
  List<Token> sentenceTokens = new List<Token>();
  PTBTokenizer tokenizer = PTBTokenizer.newPTBTokenizer(new StringReader(text), false, true);
  CoreLabel cLabel = (CoreLabel)tokenizer.next();
  while (cLabel != null)
  {
    Token token = new Token { Value = cLabel.originalText(), StartIndex = cLabel.beginPosition()};
    sentenceTokens.Add(token);
  }
  return sentenceTokens;
}
```

Razred `Lemmatizer` omotač je oko razreda `Morphology` iz knjižnice `StanfordCoreNLP`, koji provodi lematizaciju riječi engleskoga jezika. Najvažnija metoda razreda `Lemmatizer` jest metoda `Lemmatize`, koja za zadanu riječ vraća njenu lemu. Ukoliko lematizacija ne uspije, funkcija vraća zadanu riječ.

```
public string Lemmatize(string word)
{
  if (string.IsNullOrEmpty(word)) return word;
  var lemma = stanfordLemmatizer.lemmatize(new WordTag(word)).lemma();
  if (string.IsNullOrEmpty(lemma)) return word;
  if (char.IsUpper(word[0]))
    return char.ToUpper(lemma[0]) + ((lemma.Length > 1) ? lemma.Substring(1) : "");
  else return lemma;
}
```

Razred `Token` jednostavan je razred koji sadrži samo podatke pojedinačne pojavnice – tekst pojavnice (svojstvo `Value`) i njenu poziciju u tekstu (svojstvo `StartIndex`). Razred `Token` označen je atributom `Serializable` kako bi se informacije o pojavnicama obrađenog teksta mogle zapisati na tvrdi disk kao dio XML-datoteke s informacijama o obrađenom dokumentu.

```
[Serializable]
```

```
public class Token
{
  public string Value { get; set; }
  public int StartIndex { get; set; }
}
```

Razred `Sentence` sadrži sve podatke o jednoj rečenici: tekst rečenice (svojstvo `Text`), početnu poziciju rečenice u tekstu (svojstvo `StartPosition`), sve pojavnice rečenice (svojstvo `Tokens`), sve riječi u rečenici zajedno s lemom i vrstom riječi (svojstvo `TaggedWords`) te sve ovisnosne sintaktičke relacije u rečenici (svojstvo `SyntaxDependencies`). Vrijednosti svih navedenih svojstava (osim samog teksta rečenice i pozicije u izvornom tekstu) računaju se *lijeno* (engl. *lazy loading, lazy computation*), tj. tek kada se iz vanjskog kôda prvi put pokuša dohvatiti vrijednost svojstva. Kao i razred `Token`, i razred `Sentence` označen je atributom `Serializable`.

```
[Serializable]
public class Sentence
{
  public string Text { get; set; }
  public int StartPosition { get; set; }
  private List<Token> tokens;
  public List<Token> Tokens
  {
    get
    {
      if (tokens == null && !string.IsNullOrEmpty(Text))
        tokens = Tokenizer.Instance.TokenizeText(Text);
      return tokens;
    }
  }
  private List<TaggedWord> taggedWords;
  public List<TaggedWord> TaggedWords
  {
    get
    {
      if (taggedWords == null && !string.IsNullOrEmpty(Text))
        taggedWords = POSTagger.Instance.TagSentence(Text);
      return taggedWords;
    }
  }
```

```
  private List<DependencyRelation> syntaxDependencies;
  public List<DependencyRelation> SyntaxDependencies
  {
    get
    {
      if (syntaxDependencies == null && !string.IsNullOrEmpty(Text))
        syntaxDependencies = DependencyParser.Instance.Parse(Text);
      return syntaxDependencies;
    }
  }
}
```

Mapa POS

Mapa POS sadrži razrede koji se tiču označavanja vrsta riječi u rečenici. Razred POSTagger omotač je oko razreda MaxentTagger iz programske knjižnice StanfordCoreNLP, koji označava riječi rečenice vrstama riječi. Najvažnija metoda ovog razreda jest metoda TagSentence, koja zadanu rečenicu rastavlja na pojavnice te svakoj pojavnici pridružuje lemu i vrstu riječi. Izlaz metode lista je objekata razreda TaggedWord.

```
public List<TaggedWord> TagSentence(string text)
{
  List<TaggedWord> taggedWords = new List<TaggedWord>();
  string taggedText = stanfordTagger.TagString(text);
  string[] taggedTokens = taggedText.Split();
  int position = 0;
  for(int i = 0; i < taggedTokens.Length; i++)
  {
    string[] slashSplit = taggedTokens[i].Split(new char[] { '/' });
    string word = slashSplit[0];
    string pos = slashSplit[1];
    int tokenPosition = position + text.IndexOf(word);
    taggedWords.Add(new TaggedWord { Word = word, POSTag = pos, StartPosition = tokenPosition,
                                     Lemma = Lemmatizer.Instance.Lemmatize(word, pos) });
    position += text.IndexOf(word) + word.Length;
    text = text.Substring(text.IndexOf(word) + word.Length);
}
```

Razred `TaggedWord` sadrži sve bitne informacije o pojavnici, uključivo vrstu riječi pojavnice (svojstvo `POSTag`).

```
[Serializable]
public class TaggedWord
{
  public string Word { get; set; }
  public string POSTag { get; set; }
  public string Lemma { get; set; }
  public int StartPosition { get; set; }
  public int DocumentStartPosition { get; set; }

  public bool IsContentWord()
  {
    if (!string.IsNullOrEmpty(POSTag))
      return POSTagger.Instance.IsContentPOS(POSTag);
    else return false;
  }
}
```

Metoda `IsContentWord` provjerava je li pojavnica sadržajna riječ (imenice, glagoli, pridjevi, prilozi).

Mapa `Syntax`

Mapa `Syntax` sadrži programski kôd koji se odnosi na sintaktičku raščlambu rečenica. U ovoj se mapi nalaze razredi koji omogućavaju plitko parsanje rečenica (engl. *chunking*), ovisnosno parsanje rečenica (engl. *dependency parse*) te parsanje stablom strukture fraza (engl. *phrase structure tree parse*). Razred `Chunker` omotač je oko razreda `EnglishTreebankChunker` knjižnice OpenNLP, a ključne metode pomoću kojih se rečenica razlaže na sintaktičke odsječke jesu metoda `Chunk` i metoda *GetChunks* koja kao rezultat daje listu objekata razreda `Chunk`:

```
public string[] Chunk(string text)
{
  string[] tokens = Tokenizer.Instance.TokenizeText(text).Select(x => x.Value).ToArray();
  string[] poses = POS.POSTagger.Instance.TagSentence(text).Select(x => x.POSTag).ToArray();
  return chunker.Chunk(tokens, poses);
}

public List<Chunk> GetChunks(string text)
{
```

```
  List<TaggedWord> taggedWords = POS.POSTagger.Instance.TagSentence(text);
  string[] chunkedTokens = Chunk(text);
  List<Chunk> chunks = new List<Chunk>();
  Chunk currentChunk = new Chunk();
  for (int i = 0; i < chunkedTokens.Length; i++)
  {
    if (chunkedTokens[i].StartsWith("B"))
    {
      string[] split = chunkedTokens[i].Split(new char[] { '-' });
      ChunkType type = (ChunkType)Enum.Parse(typeof(ChunkType), split[1]);
      currentChunk = new Chunk();
      currentChunk.Type = type;
      currentChunk.StartOffset = tokens[i].StartIndex;
      currentChunk.TaggedWords.Add(taggedWords[i]);
      currentChunk.TokenIndexStart = i;
    }
    else if (chunkedTokens[i].StartsWith("I"))
      currentChunk.TaggedWords.Add(taggedWords[i]);
    chunks.Add(currentChunk);
    return chunks;
}
```

Razred `Chunk` sadrži sve bitne podatke o pojedinačnom sintaktičkom odsječku rečenice: vrstu sintaktičkog odsječka (svojstvo `Type`), pojavnice koje čine sintaktički odsječak (svojstvo `TaggedWords`) te poziciju početka odsječka u tekstu (svojstvo `StartIndex`).

```
[Serializable]
public class Chunk
{
  public ChunkType Type { get; set; }
  public List<TaggedWord> TaggedWords
  {
    get
    {
      if (taggedWords == null) taggedWords = new List<TaggedWord>();
      return taggedWords;
    }
    set { taggedWords = value; }
  }
  public int StartOffset { get; set; }
```

```
}
```

Razred `ConstituencyParser` obavlja parsanje rečenice stablom strukture fraza, izravno pozivajući odgovarajući razred knjižnice StanfordCoreNLP (metoda `Parse`). Za temeljnu metodu pojednostavljivanja rečenica (v. poglavlje 9.2.3) vrlo su važne metode `GetMainClause` i rekurzivna metoda `CollectMainClause` kojima se određuje glavna surečenica složene rečenice.

```
public SyntacticNode Parse(string text)
{
  var node = parser.ParseConstituency(text);
  return node;
}

public List<string> GetMainClause(SyntacticNode root)
{
  List<string> words = new List<string>();
  CollectMainClause(root, words);
  return words;
}

private void CollectMainClause(SyntacticNode node, List<string> words)
{
  if (node.Value == "SBAR" || node.Value == "CC") return;
  else if (node.NodeType == SyntacticNodeType.Word) words.Add(node.Value);
  else node.Children.ForEach(child => {CollectMainClause(child, words);});
}
```

Razred `DependencyParser` obavlja ovisnosno parsanje zadane rečenice (metoda `Parse`), pri čemu je rezultat lista ovisnosnih sintaktičkih relacija (tj. objekata razreda `DependencyRelation`).

```
public List<DependencyRelation> Parse(string text)
{
  List<DependencyRelation> dependencies = new List<DependencyRelation>();
  List<DepRel> stanfordDeps = parser.Parse(text);
  List<TaggedWord> taggedSentence = POSTagger.Instance.TagSentence(text);
  foreach (DepRel dep in stanfordDeps)
  {
    dependencies.Add(new DependencyRelation(taggedSentence, dep));
  }
```

```
  return dependencies;
}
```

Razred `DependencyRelation` sadrži sve bitne podatke pojedinačne ovisnosne sintaktičke relacije: vrstu relacije (svojstva `Relation` i `ShortRelation`), riječ koja upravlja relacijom i njen indeks u rečenici (svojstva `Governor` i `GovernorTokenIndex`) te zavisnu riječ relacije i njen indeks u rečenici (svojstva `Dependent` i `DependentTokenIndex`).

```
[Serializable]
public class DependencyRelation
{
  public string Relation { get; set; }
  public string ShortRelation { get; set; }
  public TaggedWord Governor { get; set; }
  public TaggedWord Dependent { get; set; }
  public int GovernorTokenIndex { get; set; }
  public int DependentTokenIndex { get; set; }
}
```

Mapa `NERC`

Mapa `NERC` sadrži programski kôd koji pronalazi imenovane entitete u tekstu. Razred `NamedEntityTagger` oslanja se na razred `CRFClassifier` knjižnice StanfordCoreNLP, koji po učitavanju prethodnog naučenog modela pronalazi sve imenovane entitete u danome tekstu. Najvažnije metode razreda `NamedEntityTagger` jesu metode `FindNamedEntities` i `FindDistinctNamedEntities` koje u danome tekstu dohvaćaju sva spominjanja imenovanih entiteta (`FindNamedEntities`), odnosno sve različite imenovane entitete (`FindDistinctNamedEntities`) koji pripadaju nekom od zadanih razreda imenovanih entiteta (argument `types`). Metode kao rezultat vraćaju listu imenovanih entiteta (listu objekata razreda `NamedEntity`).

```
public List<NamedEntity> FindNamedEntities(string text, List<NamedEntityType> types)
{
  List<NamedEntity> allNamedEntities = new List<NamedEntity>();
  foreach (NamedEntityType type in types)
  {
    allNamedEntities.AddRange(FindEntitiesSingleClass(text, type));
  }
  return allNamedEntities;
}
```

```
public List<NamedEntity> FindDistinctNamedEntities(string text, List<NamedEntityType> types)
{
  List<NamedEntity> allNamedEntities = FindNamedEntities(text, types);
  List<NamedEntity> distinct = new List<NamedEntity>();
  allNamedEntities.ForEach(x => {
    var distExist = distinct.Where(y => y.NamedEntityOccurrence == x.NamedEntityOccurrence);
    if (distExist == null) distinct.Add(x);
  });
  return distinct;
}
```

Razred NamedEntity sadrži podatke pojedinačnog imenovanog entiteta: tekst imenovanog entiteta (svojstvo `NamedEntityOccurrence`), vrstu imenovanog entiteta (svojstvo `Type`) te poziciju imenovanog entiteta u izvornom tekstu (svojstvo `Position`).

```
[Serializable]
public class NamedEntity
{
  public string NamedEntityOccurrence { get; set; }
  public NamedEntityType Type { get; set; }
  public int Position { get; set; }
}
```

Mapa `Coreference`

Mapa `Coreference` sadrži razrede zadužene za razrješavanje koreferencije entiteta. Razred `CoreferenceResolutor` obavlja razrješavanje koreferencije entiteta oslanjajući se na razred `CorefCoreAnnotations` iz knjižnice StanfordCoreNLP. Metoda `ResolveCoreference` koja izvršava razrješavanje koreferencije kao rezultat daje listu lanaca koreferencije (tj. objekata razreda `CoreferenceChain`).

```
public List<CoreferenceChain> ResolveCoreference(string text)
{
  Annotation document = new Annotation(text);
  corefPipeline.annotate(document);
  HashMap graph = document.get(typeof(CorefCoreAnnotations.CorefChainAnnotation));
  List<CoreferenceChain> allCoreferenceChains = new List<CoreferenceChain>();
  foreach (CorefChain stanfordCorefChain in graph.values().toArray())
  {
    CoreferenceChain chain = new CoreferenceChain();
```

```
    var repCorefMention = stanfordCorefChain.getRepresentativeMention();
    chain.RepresentativeMention = new CoreferenceMention(repCorefMention, true);
    chain.AllMentions.Add(chain.RepresentativeMention);

    foreach (CorefChain.CorefMention mention in stanfordCorefChain.getCorefMentions().toArray())
    {
      if (mention.mentionID != repCorefMention.mentionID)
        chain.AllMentions.Add(new CoreferenceMention(mention, false));
    }
    allCoreferenceChains.Add(chain);
  }
  return allCoreferenceChains;
}
```

Objekti razreda `CoreferenceChain` predstavljaju pojedinačne lance koreferentnih spominjanja te sadrže listu svih spominjanja koja tvore lanac (svojstvo `AllMentions`), kao i podatak o tome koje je od tih spominjanja reprezentativno za lanac (svojstvo `RepresentativeMention`).

```
[Serializable]
public class CoreferenceChain
{
  private List<CoreferenceMention> allMentions;
  public List<CoreferenceMention> AllMentions
  {
    get
    {
      if (allMentions == null) allMentions = new List<CoreferenceMention>();
      return allMentions;
    }
  }
  public CoreferenceMention RepresentativeMention { get; set; }
}
```

Objekti razreda `CoreferenceMention` predstavljaju pojedinačna spominjanja entiteta, a sadrže podatke o tekstu spominjanja (svojstvo `Mention`), o tome je li spominjanje reprezentativno za lanac u kojem se nalazi (svojstvo `Representative`) te podatke o poziciji spominjanja u izvornom tekstu (indeks rečenice unutar izvornog teksta te indeksi početne i završne pojavnice spominjanja unutar rečenice).

```
[Serializable]
public class CoreferenceMention
```

```
{
  public string Mention { get; set; }
  public bool Representative { get; set; }
  public int SentenceIndex { get; set; }
  public int StartTokenIndex { get; set; }
  public int EndTokenIndex { get; set; }
}
```

Mapa `Temporal`

Mapa `Temporal` sadrži programski kôd zadužen za crpljenje vremenskih izraza iz teksta. Razred `TemporalTagger` putem metode `GetTemporalExpressions` obavlja crpljenje vremenskih izraza pomoću razreda `TimeAnnotator` knjižnice StanfordCoreNLP. Rezultat izvršavanja metode jest lista vremenskih izraza pronađenih u zadanome tekstu (tj. lista objekata razreda `TemporalExpression`).

```
public List<TemporalExpression> GetTemporalExpressions(string text, DateTime creationTime)
{
  List<TemporalExpression> temporalExpressions = new List<TemporalExpression>();
  Annotation document = new Annotation(text);
  corefPipeline.annotate(document);
  ArrayList sentences = (ArrayList)document.get(typeof(CoreAnnotations.SentencesAnnotation));
  for (int i = 0; i < sentences.size(); i++)
  {
    ArrayCoreMap sentence = (ArrayCoreMap)sentences.get(i);
    var timexlist = annotator.annotateSingleSentence(sentence, creationTime.Value.ToString());
    List<Timex> timexes = new List<Timex>();
    for (int k = 0; k < timexlist.size(); k++)
    {
      Timex timex = (Timex)(timexlist.get(k)).get(typeof(TimeAnnotations.TimexAnnotation));
      int startOffset = timex.get(typeof(CoreAnnotations.OffsetBeginAnnotation)).intValue();
      int endOffset = timex.get(typeof(CoreAnnotations.OffsetEndAnnotation)).intValue() − 1;
      int startTokenIndex = timex.get(typeof(CoreAnnotations.TokenBeginAnnotation)).intValue();
      int endTokenIndex = timex.get(typeof(CoreAnnotations.TokenEndAnnotation))).intValue() − 1;
      TemporalExpression tExpression = new TemporalExpression
      {
        Text = timex.text(), TimexID = timex.tid(),
        Type = timex.timexType(), Value = timex.value(),
        StartOffset = startOffset, EndOffset = endOffset,
        StartTokenIndex = startTokenIndex, EndTokenIndex = endTokenIndex
```

```
        };
        temporalExpressions.Add(tExpression);
      }
    }
    return temporalExpressions;
}
```

Objekti razreda TemporalExpression predstavljaju pojedinačne vremenske izraze iz teksta, a sadrže tekst vremenskog izraza (svojstvo `Text`), normaliziranu vremensku vrijednost (precizno određeni datum i vrijeme na vremenskom pravcu; svojstvo `Value`), vrstu vremenskog izraza (npr. datum, vrijeme, raspon; svojstvo `Type`) te poziciju izraza u izvornom tekstu (početne i završne pozicije znakova i indeksi pojavnica).

```
[Serializable]
public class TemporalExpression
{
  public string Text { get; set; }
  public string TimexID { get; set; }
  public string Value { get; set; }
  public string Type { get; set; }
  public int StartOffset { get; set; }
  public int EndOffset { get; set; }
  public int StartTokenIndex { get; set; }
  public int EndTokenIndex { get; set; }
}
```

Mapa `Lexics/WordNet`

U podmapi `WordNet` mape `Lexics` nalazi se programski kôd koji omogućava pristup podacima leksičko-semantičke mreže WordNet. Razred `WordNetProvider` omotač je za funkcionalnost programske knjižnice `LAIR.ResourceAPIs.WordNet` razvijene u okviru projekta OpenSim4OpenCog.[5] Dvije ključne metode na koje se oslanjaju modeli koji sudjeluju u izgradnji grafova događaja jesu `GetHypernymConcepts` i `SemanticSimilarity`. Metoda `GetHypernymConcepts` za zadanu riječ ispisuje sve njene hipernime, a korištena je prilikom određivanja skupova vremenskih i lokacijskih koncepata za model crpljenja argumenata događaja (v. poglavlje 6.2).

```
public void GetHypernymConcepts(string word, string pos)
{
```

[5] `https://code.google.com/p/opensim4opencog/`

```
  SynSet s1 = wordNetSimilarityModel.WordNetEngine.GetMostCommonSynSet(word,
                WordNetPOSMapper.MapPTBPoSToWordNetPoS(pos));
  var synsets = s1.GetRelatedSynSets(WordNetEngine.SynSetRelation.Hypernym, true);
  foreach (var syn in synsets) Console.WriteLine(syn.ToString());
}
```

Metoda `SemanticSimilarity` daje mjeru semantičke sličnosti dviju zadanih riječi na temelju algoritma Wua i Palmer (1994), a obilato ju koriste model za crpljenje argumenata događaja (v. poglavlje 6.2) te model za razrješavanje koreferencije spominjanja događaja (v. poglavlje 6.4).

```
public double SemanticSimilarity(string word1, string word2)
{
  if (word1.ToLower() == word2.ToLower()) return 1;
  word1 = word1.ToLower(); word2 = word2.ToLower();
  List<SynSet> firstWordSynsets = GetSynset(word1,
                new List<WordNetEngine.POS> { Noun, Verb, Adjective });
  List<SynSet> secondWordSynsets = GetSynset(word2,
                new List<WordNetEngine.POS> { Noun, Verb, Adjective });
  double maxSimilarity = 0;
  foreach (SynSet s1 in firstWordSynsets)
  { foreach (SynSet s2 in secondWordSynsets)
    { if (s1.POS != s2.POS) continue;
      double similarity = wordNetSimilarityModel.GetSimilarity(s1, s2,
              Strategy.WuPalmer1994Maximum,
              new WordNetEngine.SynSetRelation[] { WordNetEngine.SynSetRelation.Hypernym });
      if (similarity > maxSimilarity) maxSimilarity = similarity;
    }
  }
  return maxSimilarity;
}
```

A.2.2 Knjižnica `MachineLearningLib`

U knjižnici `MachineLearningLib` nalazi se izvedba temeljne infrastrukture za korištenje i vrednovanje algoritama nadziranog strojnog učenja (mapa `Learning`). U ovoj se knjižnici (mapa `Graph`) nalazi i izvedba operacije umnoška grafova te izračun jezgrenih funkcija nad grafovima (v. poglavlje 4.2.2).

Mapa `Learning`

Mapa `Learning` sadrži programski kôd koji izvršava algoritme nadziranog strojnog učenja te omogućava vrednovanje uspješnosti modela na klasifikacijskim zadacima. U ovoj su mapi definirana sučelja za izvedbu modela za rješavanje klasifikacijskih i regresijskih problema. Razredi koji sadrže izvedbu konkretnih klasifikacijskih modela (u knjižnici `EventExtraction.Core`) pridržavaju se tih sučelja.

Sučelje `IClassifier` definira funkcionalnost koju mora imati svaki razred koji predstavlja izvedbu nekog konkretnog klasifikatora. Svaki takav razred mora imati dvije metode: metodu `Train` kojom se na temelju skupa za učenje trenira model nadziranog strojnog učenja te metodu `Predict` kojom se prethodno naučeni model primjenuje na nove primjere.

```
public interface IClassifier
{
  void Train(string trainingSetPath, string modelPath, bool probabilityEstimates);
  List<Tuple<string, double>> Predict(string testSetPath, string modelPath,
                                      string outputPath, bool probabilityEstimate);
}
```

Razred `LibSVMExecutor` omotač je koji poziva knjižnice `LibSVM` i `LibLinear` kao vanjske procese, a zadovoljava sučelje `IClassification`. Ovaj razred koriste svi modeli za izgradnju grafova događaja koji kao model nadziranog strojnog učenja koriste SVM ili logističku regresiju. Budući da zadovoljava sučelje `IClassification` ovaj razred ima konkretne izvedbe metoda `Train` i `Predict`.

```
public class LibSVMExecutor : IClassifier
{
  public string LibSVMTrainExePath { get; set; }
  public string LibSVMPredictExePath { get; set; }
  public string TrainParameters { get; set; }
  public bool Classification { get; private set; }

  public LibSVMExecutor(){}
  public LibSVMExecutor(bool classification)
  {
    this.Classification = classification;
  }

  public void Train(string trainSetPath, string modelDestinationPath,
                    bool probabilityEstimates)
  {
```

```
    ProcessStartInfo psi = new ProcessStartInfo(LibSVMTrainExePath);
    psi.Arguments = (probabilityEstimates? " −b 1 " : "") + TrainParameters +
                            " \"" + trainSetPath + "\" " + "\"" + modelDestinationPath + "\"";
    psi.UseShellExecute = false;
    Process p = Process.Start(psi);
    p.WaitForExit();
    if (p.ExitCode != 0)
    {
        throw new Exception("LibSVM svm−train failed with exit code: " + p.ExitCode);
    }
}

public List<Tuple<string, double>> Predict(string testSetPath, string modelPath,
                                                    string outputDestinationPath, bool probabilityEstimates)
{
    ProcessStartInfo psi = new ProcessStartInfo(LibSVMPredictExePath);
    psi.Arguments = (probabilityEstimates ? " −b 1 " :"") + "\"" + testSetPath + "\" "
                            + "\"" + modelPath + "\" " + "\"" + outputDestinationPath + "\"";
    psi.UseShellExecute = false;
    psi.RedirectStandardOutput = true;
    Process p = Process.Start(psi);
    p.WaitForExit();
    if (p.ExitCode != 0)
    {
        throw new Exception("LibSVM svm−pred failed with exit code: " + p.ExitCode);
    }
    StreamReader reader = new StreamReader(outputDestinationPath);
    var preds = (reader.ReadToEnd().Split(new char[] { '\r', '\n' },
                    StringSplitOptions.RemoveEmptyEntries)).ToList();
    if (probabilityEstimates) preds.RemoveAt(0);
    reader.Close();
    List<Tuple<string, double>> predictionsWithConfidence = new List<Tuple<string, double>>();
    for (int i = 0; i < preds.Count; i++)
    {
        string[] split = preds[i].Split();
        string justPrediction = split[0];
        List<double> probabilities = new List<double>();
        for (int j = 1; j < split.Length; j++) probabilities.Add(Double.Parse(split[j]));
        double confidence = probabilityEstimates ? probabilities.Max() : 0;
        predictionsWithConfidence.Add(new Tuple<string, double>(justPrediction, confidence));
```

```
    }
    return predictionsWithConfidence;
  }
}
```

Sučelje `ILearningProcess` definira funkcionalnost koju treba imati razred koji rješava neki konkretan klasifikacijski ili regresijski problem. Konkretna izvedba rješenja nekog klasifikacijskog problema mora (1) moći registrirati značajke koje će se koristiti za rješavanje problema nadziranog strojnog učenja (metode `RegisterFeatures` i `RegisterMultinomialFeatures`), (2) imati dostupan pripadni objekt za izračunavanje značajki koji zadovoljava sučelje `IFeatureExtractor` (svojstvo `FeatureExtractor`), (3) moći pokrenuti postupak učenja i ispitivanja modela (metode `StartLearningProcess` i `TrainAndTest`) te (4) provesti vrednovanje i dostaviti rezultate vrednovanja (metoda `GetResults` i svojstvo `FinalScore`).

```
public interface ILearningProcess
{
  IFeatureExtractor FeatureExtractor { get; set; }
  void RegisterFeatures();
  void RegisterMultinomialFeatures();
  void StartLearningProcess();
  string GetResults();
  double FinalScore { get; set; }
  List<string> RegisteredFeatures { get; }
  void MarkFeatureSelected(string feature);
  void ClearSelections();
  string Confusions { get; }
}
```

Sučelje `IFeatureExtractor` definira funkcionalnost koju mora imati razred koji obavlja izračun vrijednosti značajki za skup zadanih ulaznih primjera za neki klasifikacijski ili regresijski problem. U nekom konkretnom razredu koji zadovoljava ovo sučelje metoda `ExtractFeatures` računa vrijednosti značajki za svaki pojedinačni primjer dok metoda `CalculateTrainSetAggregationFeatures` računa vrijednosti značajki koje nije moguće izračunati samo na temelju promatranog primjera, već na temelju cijelog skupa za učenje (npr. značajke $f^S_{43} - f^S_{46}$ kod modela za crpljenje sidara događaja; v. poglavlje 6.1). Rezultat računanja značajki za dani skup primjera za učenje je skup objekata `TrainingExample`.

```
public interface IFeatureExtractor
{
  List<TrainingExample> ExtractFeatures(object input);
```

```
  void CalculateTrainSetAggregationFeatures(List<TrainingExample> trainingExamples);
}
```

Razred `TrainingExample` predstavlja jedan primjer za učenje za neki klasifikacijski ili regresijski problem. Svaki primjer za učenje ima skup dozvoljenih binarnih odnosno numeričkih značajki (članska varijabla `features`) te dozvoljeni skup multinomijalnih značajki (članska varijabla `multinomialFeatures`). Skup dozvoljenih značajki nekog modela definira se pozivanjem metode `RegisterFeature`. Vrijednost pojedine značajke za pojedini primjer pohranjuje se u objekt razreda `TrainingExample` pomoću metode `SetFeatureValue`. Klasifikacijski razred kojem pripada pojedini primjer pohranjuje se putem svojstva `Label`.

```
[Serializable()]
public class TrainingExample
{
  public static List<string> AllowedFeatures = new List<string>();
  public static void RegisterFeature(string featureName)
  {
    if (!AllowedFeatures.Contains(featureName)) AllowedFeatures.Add(featureName);
    else throw new NotSupportedException("Feature already registered!");
  }
  public Dictionary<string, object> features = new Dictionary<string,object>();
  public Dictionary<string, object> multinomialFeatures = new Dictionary<string, object>();
  public string Label { get; set; }

  public void SetFeatureValue(string featureName, object value, bool multinomial = false)
  {
    if (multinomial)
    {
        if (MultinomialFeature.IsFeatureRegistered(featureName))
        {
          if (multinomialFeatures.ContainsKey(featureName))
          {
            multinomialFeatures[featureName] = value;
          }
          else multinomialFeatures.Add(featureName, value);
        }
        else throw new NotSupportedException("Multinomial feature is not registered!");
      }
      else
      {
```

```
        if (AllowedFeatures.Contains(featureName))
        {
          if (features.ContainsKey(featureName)) features[featureName] = value;
          else features.Add(featureName, value);
        }
        else throw new NotSupportedException("Feature is not registered!");
      }
  }
}
```

Razred `MultinomialFeature` predstavlja multinomijalne značajke za klasifikacijske probleme. Za svaku je multinomijalnu značajku potrebno definirati skup diskretnih vrijednosti koje može poprimiti (članska varijabla `indexDictionary` pridjeljuje redne brojeve svim mogućim diskretnim vrijednostima multinomijalne značajke). U nekim klasifikacijskim problemima neki primjeri mogu istovremeno poprimiti više različitih vrijednosti za jednu multinomijalnu značajku (npr. značajka f_{14}^{VO} kod modela za crpljenje vremenskih odnosa među događajima; v. poglavlje 6.3), a radi li se o takvoj značajki definira se putem svojstva `Multivalued`. Prilikom pripremanja klasifikacijskih vektora za klasifikaciju putem razreda `LibSVMExecutor`, metoda `GetIndexForValue` dohvaća indeks za primljenu vrijednost značajke (na temelju mapiranja rječnikom `indexDictionary`).

```
public class MultinomialFeature
{
  private Dictionary<string, int> indexDictionary;
  public string Name { get; set; }
  public bool Multivalued { get; set; }

  public MultinomialFeature(string featureName, List<string> allValues)
  {
    Name = featureName;
    indexDictionary = new Dictionary<string, int>();
    int count = 1;
    foreach (string val in allValues)
    {
      indexDictionary.Add(val, count);
      count++;
    }
  }

  public int GetIndexForValue(string value)
```

```
  {
    if (string.IsNullOrEmpty(value)) return −1;
    if (indexDictionary.ContainsKey(value)) return indexDictionary[value];
    else return −1;
  }
}
```

Razred `CrossValidation` sadrži izvedbu postupka unakrsne provjere za modele madziranog strojnog učenja. Svaka unakrsna provjera odnosi se na neki konkretan klasifikacijski algoritam koji se unakrsno provjerava (članska varijabla `classifier`), a sadrži i konačnu matricu zabune (svojstvo `FinalConfusionMatrix`) u kojoj se akumuliraju klasifikacijske pogreške iz svih preklopa (engl. *folds*) unakrsne provjere. Ključna metoda ovog razreda jest metoda `XValidate` u kojoj se provodi sama unakrsna provjera, kojoj je potrebno proslijediti skup primjera za učenje i broj preklopa.

```
public class CrossValidation
{
  private IClassifier classifier;
  private XValidationFolding foldType;
  private ConfusionMatrix finalConfMatrix;
  public ConfusionMatrix FinalConfusionMatrix
  {
    get
    { if (finalConfMatrix == null) finalConfMatrix = new ConfusionMatrix(AllLabels);
        return finalConfMatrix;
    }
  }

  public void XValidate(List<TrainingExample> examples, int numFolds, string tempTrainPath,
                        string tempTestPath, string tempModelPath)
  {
     List<List<TrainingExample>> bins = CreateBins(examples, foldType);
     for (int i = 0; i < bins.Count; i++)
     {
        List<TrainingExample> trainExamples = new List<TrainingExample>();
        for (int j = 0; j < bins.Count; j++) if (j != i) trainExamples.AddRange(bins[j]);
        foreach (var trainExample in trainExamples)
        {
          trainSetExamples.Add(trainExample.ToSparseSVMString(SelectedFeatures, " "));
        }
```

```
        List<string> testSetExamples = new List<string>();
        foreach (TrainingExample testExample in bins[i])
        {
          testSetExamples.Add(testExample.ToSparseSVMString(SelectedFeatures, " "));
        }
        WriteTrainOrTestSet(tempTestPath, testSetExamples);
        WriteTrainOrTestSet(tempTrainPath, trainSetExamples);
        classifier.Train(tempTrainPath, tempModelPath, true);
        List<Tuple<string, double>> predictionValues = classifier.Predict(tempTestPath,
                                        tempModelPath, "SVM\\predictions−" + i + ".txt", true);
        List<string> trueValues = (from t in bins[i] select t.Label).ToList();
        for (int k = 0; k < predictionValues.Count; k++)
        {
          tresholdedPredictions.Add(predictionValues[k].Item1);
          tresholdedGS.Add(trueValues[k]);
        }
        ConfusionMatrix confMatrix = new ConfusionMatrix(AllLabels, tresholdedGS,
                                        tresholdedPredictions);
        FinalConfusionMatrix.AddMatrix(confMatrix);
      }
    }
}
```

Razred `ConfusionMatrix` predstavlja matricu zabune za neki klasifikacijski algoritam. Matrica zabune gradi se na temelju broja različitih razreda u klasifikacijskom problemu (pri čemu se nazivi razreda mapiraju slijedno u cijele brojeve putem članskog delegata `LabelMappingFunction`). Na temelju matrice zabune računa se uspješnost klasifikacijskog algoritma i to ukupna uspješnost putem metoda `CalculateMicroF1` i `CalculateMacroF1` te uspješnost za pojedini razred pomoću metode `ClassPerformance`.

```
public class ConfusionMatrix
{
  public LabelMappingDelegate LabelMappingFunction { get; set; }
  private List<string> classes;
  private int[,] matrix;

  public ConfusionMatrix(List<string> classes, List<string> trueValues, List<string> predictions)
  {
    this.classes = classes;
    matrix = new int[classes.Count, classes.Count];
```

```
    List<int> indexes = new List<int>();
    for (int i = 0; i < predictions.Count; i++)
    {
        int rowIndex = classes.IndexOf(predictions[i]);
        int columnIndex = classes.IndexOf(trueValues[i]);
        matrix[rowIndex, columnIndex] += 1;
    }
}

private void CalculateMacroF1()
{
    int[] truePositives = new int[classes.Count];
    int[] falsePositives = new int[classes.Count];
    int[] falseNegatives = new int[classes.Count];
    for (int i = 0; i < classes.Count; i++)
    { for (int j = 0; j < classes.Count; j++)
        { if (i == j) truePositives[i] = matrix[i, j];
            else {
                falsePositives[i] += matrix[i, j];
                falseNegatives[j] += matrix[i, j];
            }
        }
    }
    double sumF1 = 0;
    for(int i = 0; i < classes.Count; i++)
    {
        double classPrecision = ((double)truePositives[i]) /
                                    ((double)truePositives[i] + (double)falsePositives[i]);
        double classRecall = ((double)truePositives[i]) /
                                ((double)truePositives[i] + (double)falseNegatives[i]);
        sumF1 += (2 * classPrecision * classRecall) / (classPrecision + classRecall);
    }
    macroF1 = sumF1 / ((double)classes.Count);
}

private void CalculateMicroF1()
{
    int trace = 0;
    int sumAll = 0;
    for (int i = 0; i < classes.Count; i++)
```

```
    { for (int j = 0; j < classes.Count; j++)
      {
        sumAll += matrix[i, j];
        if (i == j) trace += matrix[i, j];
      }
    }
    microF1 = ((double)trace) / ((double)sumAll);
  }

  public Tuple<double, double, double> ClassPerformance(string clas)
  {
    int index = classes.IndexOf(clas);
    double tp = matrix[index, index];
    double tpfp = 0;
    double tpfn = 0;
    for (int i = 0; i < classes.Count; i++)
    {
      tpfp += matrix[index, i];
      tpfn += matrix[i, index];
    }
    double precision = tp / tpfp;
    double recall = tp / tpfn;
    double f1 = 2 * precision * recall / (precision + recall);
    return new Tuple<double, double, double>(precision, recall, f1);
  }
}
```

Mapa `Graph`

Mapa `Graph` sadrži razrede koji predstavljaju strukturu grafa te računaju jezgrene funkcije nad grafovima (v. poglavlje 4.2.2). Razredi `GraphNode`, `GraphEdge` i `Graph` generički su razredi kojima je graf definiran na općenit način, dopuštajući objektima proizvoljnih razreda da tvore vrhove i bridove. Jedini uvjet na razrede koji predstavljaju vrhove i bridove grafa jest da zadovoljavaju sučelje `IUniquelyIdentifiable` koje propisuje da svaki objekt razreda ima jedinstveni identifikator. Na taj je način svaki vrh i svaki brid grafa moguće razlikovati od svih drugih vrhova i bridova. Objekti razreda `GraphNode` predstavljaju vrhove događaja, a sadržaj vrha (svojstvo `NodeContent`) može biti objekt bilo kojeg razreda koji zadovoljava sučelje `IUniquelyIdentifiable`. Objekti razreda GraphEdge predstavljaju bridove grafa, pri čemu sadržaj brida (svojstvo `EdgeContent`) može biti objekt bilo kojeg razreda koji zadovoljava su-

čelje `IConnector`.

```
public interface IUniquelyIdentifiable
{
  object UniqueID { get; }
}

public interface IConnector<T> : IUniquelyIdentifiable
        where T : IUniquelyIdentifiable
{
  T FirstObject { get; }
  T SecondObject { get; }
}

public class GraphNode<T> where T : IUniquelyIdentifiable
{
  public int NodeID { get; set; }
  public string NodeRepresentation { get; set; }
  public T NodeContent { get; set; }
}

public class GraphEdge<T> where T : IUniquelyIdentifiable
{
  public int EdgeID { get; set; }
  public string EdgeRepresentation { get; set; }
  public T EdgeContent { get; set; }
  public int FirstNodeID { get; set; }
  public int SecondNodeID { get; set; }
}
```

Objekti razreda `Graph` predstavljaju cjelokupne grafove, a grade se na temelju listi objekata koji trebaju tvoriti vrhove odnosno bridove (konstruktor razreda `Graph`). U ovom se razredu nalaze i izvedbe nekih uobičajenih algoritama nad grafovima, poput pronalaženja svih povezanih komponenti grafa (potrebno za vrednovanje grafova događaja mjerom LCC opisanom u poglavlju 7), ali i metoda za računanje umnoška grafa s nekim drugim grafom (metoda `ProductGraph`).

```
public class Graph<N, E> where N : IUniquelyIdentifiable
                                where E : IConnector<N>
{
  private Dictionary<int, GraphNode<N>> nodesIndexDictionary;
  private Dictionary<string, int> nodesDictionary;
```

```
private Dictionary<int, GraphEdge<E>> edgesIndexDictionary;
private Dictionary<string, int> edgesDictionary;
private List<int>[,] adjacencyMatrix;
public List<int>[,] AdjacencyMatrix
{
  get{ return adjacencyMatrix; }
}

public Graph(List<N> nodeObjectsList, List<E> edgeObjectsList)
{
  nodesDictionary = new Dictionary<string, int>();
  edgesDictionary = new Dictionary<string, int>();
  edgesIndexDictionary = new Dictionary<int, GraphEdge<E>>();
  nodesIndexDictionary = new Dictionary<int, GraphNode<N>>();
  int cnt = 0;
  // Izgradnja vrhova
  foreach (var nodeObject in nodeObjectsList)
  { GraphNode<N> newNode = new GraphNode<N>();
    newNode.NodeID = cnt; cnt++;
    newNode.NodeRepresentation = nodeObject.UniqueID.ToString();
    newNode.NodeContent = nodeObject;
    if (!nodesDictionary.ContainsKey(newNode.NodeRepresentation))
    { nodesIndexDictionary.Add(newNode.NodeID, newNode);
      nodesDictionary.Add(newNode.NodeRepresentation, newNode.NodeID);
    }
  }
  // Izgradnja bridova
  cnt = 0;
  foreach (var edgeObject in edgeObjectsList)
  { GraphEdge<E> newEdge = new GraphEdge<E>();
    newEdge.EdgeID = cnt; cnt++;
    newEdge.EdgeRepresentation = edgeObject.UniqueID.ToString();
    newEdge.EdgeContent = edgeObject;
    newEdge.FirstNodeID = nodesDictionary[edgeObject.FirstObject.UniqueID.ToString()];
    newEdge.SecondNodeID = nodesDictionary[edgeObject.SecondObject.UniqueID.ToString()];
    if (!edgesDictionary.ContainsKey(newEdge.EdgeRepresentation))
    { edgesDictionary.Add(newEdge.EdgeRepresentation, newEdge.EdgeID);
      edgesIndexDictionary.Add(newEdge.EdgeID, newEdge);
    }
  }
```

```
  }

  public Graph<ProductGraphNode<N>, ProductGraphEdge<E, N>>
          ProductGraph(Graph<N,E> secondGraph, IGraphNodeMatch<N, E> nodeMatcher,
          IGraphEdgeMatch<N, E> edgeMatcher)
  {
    List<ProductGraphNode<N>> pgNodes = nodeMatcher.GetProductNodes(this, secondGraph);
    List<ProductGraphEdge<E, N>> pgEdges = edgeMatcher.GetProductEdges(productGraphNodes,
                                                                       this, secondGraph);
    return (new Graph<ProductGraphNode<N>, ProductGraphEdge<E, N>>(pgNodes, pgEdges));
  }
}
```

Razredi `ProductGraphNode` i `ProductGraphEdge` predstavljaju vrhove i bridove grafa koji je rezultat umnoška dvaju grafova. Ti razredi sadrže podatke o vrhovima odnosno bridovima ulaznih grafova na temelju kojih su vrhovi odnosno bridovi umnoška stvoreni.

```
public class ProductGraphNode<N> : IUniquelyIdentifiable
                where N : IUniquelyIdentifiable
{
  public N FirstGraphNode { get; set; }
  public N SecondG
  raphNode { get; set; }
  public int FirstNodeID { get; set; }
  public int SecondNodeID { get; set; }
  public object UniqueID
  {
    get { return FirstGraphNode.UniqueID + " :: " + SecondGraphNode.UniqueID; }
  }
}

public class ProductGraphEdge<E, N> : IConnector<ProductGraphNode<N>>
          where E : IConnector<N>
          where N : IUniquelyIdentifiable
{
  public ProductGraphNode<N> FirstProductGraphNode { get; set;}
  public ProductGraphNode<N> SecondProductGraphNode { get; set; }
  public E FirstGraphEdge { get; set; }
  public E SecondGraphEdge { get; set; }
  public int FirstGraphEdgeID { get; set; }
```

```
  public int SecondGraphEdgeID { get; set; }
  public object UniqueID
  {
    get { return FirstGraphEdge.UniqueID + " :: " + SecondGraphEdge.UniqueID; }
  }
}
```

Sučelja `IGraphNodeMatch` i `IGraphEdgeMatch` definiraju ulaze i izlaze funkcija na temelju kojih se određuju vrhovi i bridovi grafa koji nastaje umnoškom dvaju ulaznih grafova. U knjižnici `EventExtraction.Core` postoje razredi koji zadovoljavaju ova sučelja, a na različite načine određuju podudarnost vrhova odnosno bridova ulaznih grafova (npr. tenzorski umnožak nasuprot konormalnog umnoška). Sučelje `IGraphLabelMatch` definira funkciju koja samo provjerava podudarnost dva vrha (i daje mjeru njihove sličnosti), bez stvaranja zajedničkog kompozitnog vrha (kao što je to slučaj za metodu `GetProductNodes` sučelja `IGraphNodeMatch`).

```
public interface IGraphNodeMatch<N, E> where N : IUniquelyIdentifiable
        where E : IConnector<N>
{
        List<ProductGraphNode<N>> GetProductNodes(Graph<N, E> firstGraph,
                                                  Graph<N, E> secondGraph);
}

public interface IGraphEdgeMatch<N, E> where N : IUniquelyIdentifiable
                where E : IConnector<N>
{
  List<ProductGraphEdge<E, N>> GetProductEdges(List<ProductGraphNode<N>> pgNodes,
                                  Graph<N, E> firstGraph, Graph<N, E> secondGraph);
}

public interface IGraphLabelMatch
{
  bool NodeLabelMatch(object firstNode, object secondNode);
  double NodeSimilarityMeasure(object firstNode, object secondNode);
}
```

Sučelje `IGraphKernelFunction` određuje metodu koju mora implementirati svaki razred koji računa neku jezgrenu funkciju nad grafovima. Dva konkretna razreda, `ProductGraphKernel` i `WeightedDecompositionKernel` zadovoljavaju ovo sučelje, a implementiraju računanje jezgrene funkcije umnoška grafova, odnosno jezgrene funkcije težinske dekompozicije grafa (v. poglavlje 4.2.2).

```
public interface IGraphKernelFunction<N, E> where N : IUniquelyIdentifiable
                        where E : IConnector<N>
{
  double CalculateKernelValue(Graph<N, E> firstGraph, Graph<N, E> secondGraph);
  double KernelValue { get; }
}

public class ProductGraphKernel<N, E> : IGraphKernelFunction<N, E>
                  where N : IUniquelyIdentifiable
                  where E : IConnector<N>
{
  private double kernelValue = −1;
  public double KernelValue { get { return kernelValue; }}
  public IGraphNodeMatch<N,E> nodeMatcher { get; set; }
  public IGraphEdgeMatch<N,E> edgeMatcher { get; set; }
  public double CalculateKernelValue(Graph<N, E> firstGraph, Graph<N, E> secondGraph)
  {
    double lambda = (1.0 / ((double)graph.MaxDegree + 1));
    var productGraph = firstGraph.ProductGraph(secondGraph, nodeMatcher, edgeMatcher);
    kernelValue = MatrixOperations.SumAllElements(
                        MatrixOperations.DenseMatrixInverse(
                        MatrixOperations.SubtractFromUnitMatrix(
                        MatrixOperations.MultiplyWithScalar(graph.AdjacencyMatrix, lambda))));
    return kernelValue;
  }
}

public class WeightedDecompositionKernel<N, E> : IGraphKernelFunction<N, E>
                  where N : IUniquelyIdentifiable
                  where E : IConnector<N>
{
  private double kernelValue = −1;
  public double KernelValue { get { return kernelValue; }}
  public IGraphLabelMatch SelectorMatcher { get; set; }
  public IGraphKernelFunction<N, E> ContextKernelFunction { get; set; }
  public double CalculateKernelValue(Graph<N, E> firstGraph, Graph<N, E> secondGraph)
  {
    foreach (var firstNode in firstGraph.AllNodes)
    { foreach (var secondNode in secondGraph.AllNodes)
```

```
        { double selScore =
            SelectorMatcher.NodeSimilarityMeasure(firstNode.NodeContent,
                                                  secondNode.NodeContent);
          double ctextSim =
            ContextKernelFunction.CalculateKernelValue(firstGraph.SubGraphForNode(firstNode),
                                                       secondGraph.SubGraphForNode(secondNode));
          kValue += selScore * ctextSim;
        }
      }
      return kValue;
    }
}
```

A.2.3 Knjižnica `EventExtraction.Core`

Knjižnica `EventExtraction.Core` sadrži programski kôd koji implementira funkcionalnost izgradnje grafova događaja i mjerenja sličnosti između grafova događaja jezgrenim funkcijama. U ovoj se knjižnici nalaze konkretni razredi koji implementiraju mnoga od prethodno prikazanih sučelja iz knjižnice `MachineLearningLib`. Kao i u prethodnim knjižnicama, i u ovoj su razredi grupirani po mapama u skladu s funkcionalnošću koju implementiraju.

Mapa `DataContainers`

U ovoj se mapi nalaze podatkovno orijentirani razredi koji sadrže podatke dobivene primjenom modela za crpljenje informacija na izvorni tekst. Objekti razreda `MicroEvent` predstavljaju pojedinačna spominjanja događaja. Svaki objekt tog razreda sadrži jedinstveni identifikator događaja (svojstvo `ID`), sidro događaja (svojstvo `EventCarrier`), lemu i vrstu riječi sidra događaja (svojstvo `EventCarrierTaggedWord`), poziciju sidra događaja u tekstu (svojstvo `Position`), semantički razred događaja (svojstvo EventClass) te listu argumenata događaja (svojstvo `Arguments`). Semantički razredi događaja definirani su putem enumeracije `EventClass`.

```
public enum EventClass
{
  Occurrence, I_Action, Reporting, Perception, StateChange,
}
[Serializable]
public class MicroEvent : IUniquelyIdentifiable
{
  public int ID { get; set; }
```

```
  public string EventCarrier { get; set; }
  public EventClass EventClass { get; set; }
  public int Position { get; set; }
  private TaggedWord eventCarrierTaggedWord;
  public TaggedWord EventCarrierTaggedWord
  { get
    { if (eventCarrierTaggedWord == null)
        eventCarrierTaggedWord = POSTagger.Instance.TagSingleToken(EventCarrier);
      return eventCarrierTaggedWord;
    }
  }
  private List<Argument> arguments;
  public List<Argument> Arguments
  { get
    { if (arguments == null) arguments = new List<Argument>();
      return arguments;
    }
  }
}
```

Objekti razreda `Argument` predstavljaju pojedinačne argumente spominjanja događaja. Svojstvo `ArgumentWord` određuje glavnu riječ argumenta (riječ, lema, vrsta riječi), dok svojstvo `ArgumentChunk` određuje sintaktički odsječak glavne riječi argumenta. Za argument se još pamte i koreferentno spominjanje entiteta koje argument čini (svojstvo `ArgumentCoreference`) te imenovani entitet odnosno vremenski argumenta (ako argument čini neki imenovani entitet ili vremenski izraz; svojstva `NamedEntity` i `Tempex`). Svaki argument događaja ima priduženu semantičku ulogu (svojstvo `Type`). Dopuštene semantičke uloge argumenata definirane su enumeracijom `ArgumentType`.

```
public enum ArgumentType
{
  Agent, Target, Location, Time, Other
}
[Serializable]
public class Argument
{
  public TaggedWord ArgumentWord { get; set; }
  public int ArgumentTokenIndex { get; set; }
  public int StartPosition { get; set; }
  public NamedEntity NamedEntity { get; set; }
```

```
  public TemporalExpression Tempex { get; set; }
  public Chunk ArgumentChunk { get; set; }
  public ArgumentType Type { get; set; }
  public CoreferenceMention ArgumentCoreference { get; set; }
  private string argumentPhrase;
  public string ArgumentPhrase
  { get
    { if (string.IsNullOrEmpty(argumentPhrase))
      {
          argumentPhrase = string.Empty;
          ArgumentChunk.TaggedWords.ForEach(x => argumentPhrase += x.Word + " ");
          argumentPhrase = argumentPhrase.Trim();
      }
      return argumentPhrase;
    }
  }
}
```

Objekti razreda `Relation` predstavljaju pojedinačne odnose između para događaja. Svaki je objekt ovog razreda određen parom događaja koje povezuje (svojstva `FirstEvent` i `SecondEvent`) i vrstom odnosa (vremenski odnos ili koreferencija; svojstvo RelationType). Dodatno, za odnos se pamti je li nastao izravno crpljenjem iz teksta ili tranzitivnim zatvaranjem na temelju drugih odnosa koji su ekstrahirani iz teksta (svojstvo `TransitivelyCreated`). Dopuštene odnose između događaja definira enumeracija `RelationType`.

```
public enum RelationType
{
  TemporalBefore,
  TemporalOverlap,
  TemporalIdentity,
  TemporalAfter,
  Coreferent
}
[Serializable]
public class Relation : IConnector<MicroEvent>
{
  public MicroEvent FirstEvent { get; set; }
  public MicroEvent SecondEvent { get; set; }
  public RelationType RelationType { get; set; }
  public bool TransitivelyCreated { get; set; }
```

```
}
```

Konačno, objekt razreda `ProcessedArticle` sadrži sve podatke koji su nastali obradom jednog dokumenata. Uz podatke dobivene obradom jezičnim alatima iz knjižnice NLPCommonCode kao što su rečenice, pojavnice, ovisnosne sintaktičke relacije rečenica, imenovani entiteti, vremenski izrazi i dr., svaki objekt ovog razreda predstavlja jedan graf događaja budući da sadrži sva ekstrahirana spominjanja događaja i sve ekstrahirane vremenske odnose i odnose koreferencije među spominjanjima događaja (svojstva `MicroEvents` i `Relations`). Vrijednosti svojstava razreda `ProcessedArticle` računaju se *lijeno*, tj. kada se prvi put dohvaća vrijednost svojstva iz nekog drugog dijela programskog kôda.

```
[Serializable]
public class ProcessedArticle : IClusteringExample
{
  public string Text { get; set; }
  public string FileName { get; set; }
  private List<Sentence> sentences;
  public List<Sentence> Sentences
  { get
    { if (sentences == null && !string.IsNullOrEmpty(Text))
          sentences = SentenceSplitter.Instance.StanfordSplit(Text);
      return sentences;
    }
  }
  private List<Token> tokens;
  public List<Token> Tokens
  { get
    { if (tokens == null && !string.IsNullOrEmpty(Text))
          tokens = Tokenizer.Instance.TokenizeText(Text);
      return tokens;
    }
  }
  private List<TaggedWord> allTokensFlat;
  public List<TaggedWord> AllTokensFlat
  { get
    { if (allTokensFlat == null)
      { allTokensFlat = new List<TaggedWord>();
        for (int i = 0; i < sentences.Count; i++)
        { for (int j = 0; j < sentences[i].TaggedWords.Count; j++)
          { allTokensFlat.Add(sentences[i].TaggedWords[j]);
```

```
                sentences[i].TaggedWords[j].DocumentStartPosition = sentences[i].StartPosition
                                                    + sentences[i].TaggedWords[j].StartPosition;
            }
        }
    }
    return allTokensFlat;
  }
}
private List<MicroEvent> microEvents;
public List<MicroEvent> MicroEvents
{ get
  { if (microEvents == null) microEvents = new List<MicroEvent>();
    return microEvents;
  }
}
private List<Relation> relations;
public List<Relation> Relations
{ get
  { if (relations == null) relations = new List<Relation>();
    return relations;
  }
}
}
```

Mapa `Learning`

Mapa `Learning` sadrži razrede koji implementiraju sučelja `ILearningProcess` i `IFeatureExtractor` (v. poglavlje A.2.2) za tri modela nadziranog strojnog učenja: (1) model za crpljenje činjeničnih sidara događaja (razredi `EventsLearningProcess` i `EventsFeatureExtractor`), (2) model za crpljenje vremenskih odnosa među događajima (razredi *TimeMLRelationsLearningProcess* i TimeMLRelationsFeatureExtractor) te (3) model za razrješavanje koreferencije parova događaja (razredi `ECBCoreferenceLearningProcess` i `ECBCoreferenceFeatureExtractor`). U nastavku je dan odsječak programskog kôda razreda `EventsLearningProcess` kao primjera implementacije sučelja `ILearningProcess`.

```
public class EventsLearningProcess : ILearningProcess
{
  private MachineLearningLib.SVM.LibSVMExecutor svmExecutor;
  private MachineLearningLib.Learning.CrossValidation crossValidator;
  private List<TrainingExample> allTrainingExamples;
```

```
private List<string> registeredFeatures = new List<string>();
private List<string> selectedFeatures = new List<string>();

public EventsLearningProcess(object input)
{
   this.input = input;
   FeatureExtractor = new EventsFeatureExtractor();
   svmExecutor = new MachineLearningLib.SVM.LibSVMExecutor(true);
   svmExecutor.TrainParameters = " −s 0 −c 1 ";
   svmExecutor.LibSVMTrainExePath = "SVM\\LibLinear\\train.exe";
   svmExecutor.LibSVMPredictExePath = "SVM\\LibLinear\\predict.exe";
   crossValidator = new CrossValidation(svmExecutor, XValidationFolding.Interleaved);

   RegisterFeatures();
   RegisterMultinomialFeatures();
}

public void RegisterFeatures()
{
   TrainingExample.RegisterFeature("capitalized");
   TrainingExample.RegisterFeature("wordUsuallyEvent");
   TrainingExample.RegisterFeature("hasModal");
   TrainingExample.RegisterFeature("hasDirectObject");
   ...
   registeredFeatures.AddRange(TrainingExample.AllowedFeatures);
}

public void RegisterMultinomialFeatures()
{
   MultinomialFeature mfWord= new MultinomialFeature("word",
                              CorpusStats.LoadList("allWords.txt"));
   MultinomialFeature.RegisterMultinomialFeature(mfWord);
   MultinomialFeature mfLemma = new MultinomialFeature("lemma",
                                        CorpusStats.LoadList("allLemmas.txt"));
   MultinomialFeature.RegisterMultinomialFeature(mfLemma);
   ...
   registeredFeatures.AddRange(MultinomialFeature.AllRegisteredFeatures);
}
```

```
  public void StartLearningProcess()
  {
    results = Execute();
  }

  private Dictionary<string, object> Execute()
  {
    selectedFeatures = registeredFeatures;
    if (!(input is List<Article>)) throw new NotSupportedException();
    var allTrainingExamples = FeatureExtractor.ExtractFeatures(input);
    results = new Dictionary<string, object>();
    crossValidator.AllLabels = new List<string> { "1", "2", "3", "4", "5"};
    //crossValidator.AllLabels = new List<string> { "0", "1"};
    crossValidator.LabelMappingFunction = EventsFeatureExtractor.GetRealLabel;
    //crossValidator.LabelMappingFunction = EventsFeatureExtractor.GetRealLabelBinary;
    crossValidator.SelectedFeatures = selectedFeatures;
    crossValidator.FeatureExtractor = FeatureExtractor;
    crossValidator.XValidate(allTrainingExamples, 10, "tempTrainEE.txt", "tempTestEE.txt",
                                    "anchorEx−Class.model");
    results.Add("Overall", crossValidator.FinalConfusionMatrix.PerformanceString);
    FinalScore = crossValidator.FinalConfusionMatrix.MicroF1;
    return results;
  }
  ...
}
```

Razredi koji implementiraju sučelje `IFeatureExtractor` sadrže programski kôd koji računa vrijednosti značajki za dane primjere za učenje (ili primjere za ispitivanje). U nastavku je dan odsječak programskog kôda razreda `EventsFeatureExtractor` kao primjera implementacije sučelja `IFeatureExtractor`.

```
public class EventsFeatureExtractor : IFeatureExtractor
{
  public List<TrainingExample> ExtractFeatures(object input)
  {
    List<Article> articles = (List<Article>)input;
    List<TrainingExample> examples = new List<TrainingExample>();
    articles.ForEach(x => {
        x.Sentences.ForEach(y => {
          y.TaggedWords.ForEach(z =>
```

```
            {
              var example = GetExampleForToken(z, y, x.Events);
              examples.Add(example);
            });
          });
      });
      return examples;
    }
    public TrainingExample GetExampleForToken(TaggedWord taggedWord, Sentence sentence,
                                              List<EventInfo> events)
    {
      TrainingExample example = new TrainingExample();
      int tokenIndex = sentence.TaggedWords.IndexOf(taggedWord);
      List<DependencyRelation> tokenDependencies = sentence.SyntaxDependencies.Where(x =>
        (x.Governor.Word == taggedWord.Word && x.GovernorTokenIndex == tokenIndex)
      || (x.Dependent.Word == taggedWord.Word && x.DependentTokenIndex == tokenIndex)).ToList();
      var matchingEvent = events.Where(x => x.Event.Trim() == taggedWord.Word.Trim() &&
        x.PositionInText == (sentence.StartPosition + taggedWord.StartPosition)).FirstOrDefault();
      example.Label = (matchingEvent != null ? GetLearningLabel(matchingEvent.EventType) : "0";

      example.SetFeatureValue("capitalized", char.IsUpper(taggedWord.Word[0]));
      example.SetFeatureValue("word", taggedWord.Word.ToLower(), true);
      example.SetFeatureValue("lemma", taggedWord.Lemma.ToLower(), true);
      ...
      example.SetFeatureValue("dependencies", stringTokenDependencies, true);
      ...
      return example;
    }
    ...
}
```

Mapa `ArgumentExtraction`

Kako model za crpljenje argumenata događaja nije model nadziranog strojnog učenja, već model temeljen na pravilima, programski kôd za crpljenje argumenata događaja izdvojen je u zasebnu mapu. Model za crpljenje argumenata (v. poglavlje 6.2) implementiran je pomoću dva razreda: `EventArgsExtraction` i `RoleSimilarityHelper`. Razred `EventArgsExtraction` implementira sintaktička pravila za prepoznavanje kandidata za argumente događaja. Za svako se spominjanje događaja razmatraju sve ovisnosne sintaktičke relacije između sidra događaja

i ostalih riječi u rečenici (metode `ExtractEventArguments` i `SyntacticPatternExtraction`). Za svaki ekstraktivni sintaktički uzorak (v. poglavlje 6.2) postoji zasebna metoda koja provjerava zadovoljava li neka riječ u rečenici (u odnosu na sidro događaja) taj uzorak. U nastavku je prikazan programski kôd za uzorak *Prijedložni objekt* (metoda `PerpositionalObject`).

```
public class EventArgsExtraction
{
  ...
  private void ExtractEventArguments(MicroEvent mEvent, Sentence sentence)
  {
    var mEventTaggedWordInSentence = sentence.TaggedWords.Where(x =>
     x.Word == mEvent.EventCarrier && x.DocumentStartPosition == mEvent.Position);
    int mEventTokenIndex = sentence.TaggedWords.IndexOf(mEventTaggedWordInSentence);
    var mEventDependencies = sentence.SyntaxDependencies.Where(x =>
                              (x.Governor.Word == mEvent.EventCarrier
                            && x.Governor.DocumentStartPosition == mEvent.Position)
                            || (x.Dependent.Word == mEvent.EventCarrier
                            && x.Dependent.DocumentStartPosition == mEvent.Position)).ToList();
    SyntacticPatternExtraction(mEvent, mEventTokenIndex, mEventDependencies, sentence);
  }

  private void SyntacticPatternExtraction(MicroEvent mEvent, int mEventTokenIndex,
                        List<DependencyRelation> mEventDependencies, Sentence sentence)
  {
    NominalSubject(mEvent, mEventTokenIndex, mEventDependencies, sentence);
    NominalSubjectPassive(mEvent, mEventTokenIndex, mEventDependencies, sentence);
    DirectObject(mEvent, mEventTokenIndex, mEventDependencies, sentence);
    ...
    PrepositionalObject(mEvent, mEventTokenIndex, mEventDependencies, sentence);
    ...
  }
  private void PrepositionalObject(MicroEvent mEvent, int mEventTokenIndex,
              List<DependencyRelation> mEventDependencies, Sentence sentence)
  {
    var preps = mEventDependencies.Where(x => x.ShortRelation == "prep"
                    && x.Governor.Word == mEvent.EventCarrier
                    && x.GovernorTokenIndex == mEventTokenIndex).ToList();
    preps.ForEach(x => {
      string preposition = x.Dependent.Word.ToLower();
```

```
        var pobj = sentence.SyntaxDependencies.Where(y => y.ShortRelation == "pobj"
                    && y.Governor.Word == x.Dependent.Word
                    && y.GovernorTokenIndex == x.DependentTokenIndex).FirstOrDefault();
        if (pobj != null)
        {
          var argChunk = Chunker.Instance.GetSpecificChunk(sentence.Chunks, pobj.Dependent.Word,
                          pobj.DependentTokenIndex, pobj.Dependent.POSTag);
          var role = RoleSimilarityHelper.DetermineArgumentType(mEvent, preposition,
                                                    pobj.Dependent, argChunk, sentence);
          mEvent.AddArgument(new Argument{ ArgumentWord = pobj.Dependent,
                    ArgumentTokenIndex = pobj.DependentTokenIndex,
                    Type = role ,ArgumentChunk = argChunk});
        }
      });
  }
  ...
}
```

Za sintaktičke uzorke koje mogu zadovoljiti argumenti više različitih semantičkih uloga (poput uzorka *Prijedložni objekt*) razrješavanje semantičke uloge obavlja se u razredu `RoleSimilarityHelper` (metoda `DetermineArgumentType`).

```
public class RoleSimilarityHelper
{
  ...
  public ArgumentType DetermineArgumentType(MicroEvent mEvent, string preposition,
                          TaggedWord argumentWord, Chunk argChunk, Sentence sentence)
  {
      bool hasSubject = mEvent.Arguments.Where(x => x.Type == ArgumentType.Agent).Count() > 0;
      bool neMatch = sentence.NamedEntities.Where(x =>
        x.Type == NLPCommonCode.NERC.NamedEntityType.Organization
        || x.Type == NLPCommonCode.NERC.NamedEntityType.Person).
        Any(x => x.NamedEntityOccurrence.Contains(prepositionalObject.Word));
      if ((!hasSubject || neMatch) && preposition == "by") return ArgumentType.Agent;
      if (preposition == "against") return ArgumentType.Target;
      foreach (var x in sentence.NamedEntities)
      {
        if (x.NamedEntityOccurrence.Contains(prepositionalObject.Word)
            && x.Type == NLPCommonCode.NERC.NamedEntityType.Location)
                return ArgumentType.Location;
```

```
      }
      foreach (var x in sentence.TemporalExpressions)
      {
        if (x.Text.Contains(prepositionalObject.Word))return ArgumentType.Time;
      }
      if (locativePrepositions.Contains(preposition)) return ArgumentType.Location;
      if (temporalPrepositions.Contains(preposition)) return ArgumentType.Time;

      return WordNetSimilarityDiscrimination(prepositionalObject, 0.5, 0.7);
  }
  ...
}
```

Mapa `Kernels`

Razredi u ovoj mapi implementiraju sučelja za usporedbu vrhova i bridova između grafova na način da općenitu funkcionalnost usporedbe grafova jezgrenim funkcijama prilagođavaju grafovima događaja. Razred `CoreferenceNodeMatch` implementira sučelje `IGraphNodeMatch` na način da se vrhovi produktnog grafova grade na temelju parova koreferentnih spominjanja događaja iz ulaznih grafova.

```
public class CoreferenceNodeMatch : IGraphNodeMatch<MicroEvent, Relation>
{
  private ECBCoreferenceLearningProcess learningProcess;
  public CoreferenceNodeMatch(ECBCoreferenceLearningProcess learningProcess)
  {
    this.learningProcess = learningProcess;
  }
  public List<ProductGraphNode<MicroEvent>> GetProductNodes(
        Graph<MicroEvent, Relation> firstGraph, Graph<MicroEvent, Relation> secondGraph)
  {
    List<ProductGraphNode<MicroEvent>> productGraphNodes =
                                  new List<ProductGraphNode<MicroEvent>>();
    List<MEventSimilarityInfo> eventCorefInfos = new List<MEventSimilarityInfo>();
    foreach (var firstGraphNode in firstGraph.AllNodes)
    { foreach (var secondGraphNode in secondGraph.AllNodes)
      {
        var firstMicroEvent = firstGraphNode.NodeContent;
        var secondMicroEvent = secondGraphNode.NodeContent;
        eventCorefInfos.Add(new MEventSimilarityInfo{ FirstMicroEvent = firstMicroEvent,
```

```
                                                    SecondMicroEvent = secondMicroEvent});
        }
      }
      learningProcess.Predict(eventCorefInfos);
      eventCorefInfos.ForEach(eci =>
      {
        var evCorefInfo = eci.Item1;
        if (evCorefInfo.ContextConfidence == 1)
        {
          var firstGraphSelectedNode = firstGraph.AllNodes.Where(x =>
                        x.NodeContent.ID == evCorefInfo.FirstMicroEvent.ID).Single();
          var secondGraphSelectedNode = secondGraph.AllNodes.Where(x =>
                        x.NodeContent.ID == evCorefInfo.SecondMicroEvent.ID).Single();
          ProductGraphNode<MicroEvent> newNode = new ProductGraphNode<MicroEvent>();
          newNode.FirstGraphNode = firstGraphSelectedNode.NodeContent;
          newNode.SecondGraphNode = secondGraphSelectedNode.NodeContent;
          newNode.FirstNodeID = firstGraphSelectedNode.NodeID;
          newNode.SecondNodeID = secondGraphSelectedNode.NodeID;
          productGraphNodes.Add(newNode);
        }
        productGraphNodes.Add(newNode);
      });
      return productGraphNodes;
    }
}
```

Razredi `TensorEdgeMatch` i `ConormalEdgeMatch` implementiraju sučelje `IGraphEdgeMatch` tako da grade tenzorski odnosno konormalni umnožak (v. poglavlje 4.2.3) između zadanih grafova. U razredu je `TensorEdgeMatch` definirano da se brid u umnožak dodaje samo ako odgovarajući bridovi ulaznih grafova imaju podudarnu oznaku (isti vremenski odnos) u oba ulazna grafa.

```
public class TensorEdgeMatch : IGraphEdgeMatch<MicroEvent, Relation>
{
  public List<ProductGraphEdge<Relation, MicroEvent>> GetProductEdges(
      List<ProductGraphNode<MicroEvent>> productGraphNodes,
      Graph<MicroEvent,Relation> firstGraph, Graph<MicroEvent, Relation> secondGraph)
  {
      List<ProductGraphEdge<Relation, MicroEvent>> productGraphEdges =
                                    new List<ProductGraphEdge<Relation, MicroEvent>>();
```

```
      for (int i = 0; i < productGraphNodes.Count − 1; i++)
      { for (int j = i + 1; j < productGraphNodes.Count; j++)
        {
            var firstGraphCorrespondingEdge = firstGraph.AllEdges.Where(x =>
                (x.FirstNodeID == productGraphNodes[i].FirstNodeID
              && x.SecondNodeID == productGraphNodes[j].FirstNodeID).FirstOrDefault();
            var secondGraphCorrespondingEdge = secondGraph.AllEdges.Where(x =>
                (x.FirstNodeID == productGraphNodes[i].SecondNodeID
              && x.SecondNodeID == productGraphNodes[j].SecondNodeID).FirstOrDefault();
            if (firstGraphCorrespondingEdge == null
                || secondGraphCorrespondingEdge == null) continue;
            else if (firstGraphCorrespondingEdge.EdgeContent.RelationType ==
                  secondGraphCorrespondingEdge.EdgeContent.RelationType )
            {
              ProductGraphEdge<Relation, MicroEvent> newProductEdge =
                                                  new ProductGraphEdge<Relation, MicroEvent>();
              newProductEdge.FirstGraphEdge = firstGraphCorrespondingEdge.EdgeContent;
              newProductEdge.SecondGraphEdge = secondGraphCorrespondingEdge.EdgeContent;
              newProductEdge.FirstGraphEdgeID = firstGraphCorrespondingEdge.EdgeID;
              newProductEdge.SecondGraphEdgeID = secondGraphCorrespondingEdge.EdgeID;
              productGraphEdges.Add(newProductEdge);
            }
          }
      }
      return productGraphEdges;
  }
}
```

U razredu je `ConormalEdgeMatch` definirano da se brid u umnožak dodaje ako odgovarajući brid postoji u barem jednome ulaznom grafu.

```
public class ConormalEdgeMatch : IGraphEdgeMatch<MicroEvent, Relation>
{
  public List<ProductGraphEdge<Relation, MicroEvent>> GetProductEdges(
      List<ProductGraphNode<MicroEvent>> productGraphNodes,
      Graph<MicroEvent,Relation> firstGraph, Graph<MicroEvent, Relation> secondGraph)
  {
      List<ProductGraphEdge<Relation, MicroEvent>> productGraphEdges =
                                        new List<ProductGraphEdge<Relation, MicroEvent>>();
      for (int i = 0; i < productGraphNodes.Count − 1; i++)
```

```
    { for (int j = i + 1; j < productGraphNodes.Count; j++)
      {
        var firstGraphCorrespondingEdge = firstGraph.AllEdges.Where(x =>
              (x.FirstNodeID == productGraphNodes[i].FirstNodeID
            && x.SecondNodeID == productGraphNodes[j].FirstNodeID).FirstOrDefault();
        var secondGraphCorrespondingEdge = secondGraph.AllEdges.Where(x =>
              (x.FirstNodeID == productGraphNodes[i].SecondNodeID
            && x.SecondNodeID == productGraphNodes[j].SecondNodeID).FirstOrDefault();
        if (firstGraphCorrespondingEdge != null || secondGraphCorrespondingEdges != null)
        {
            ProductGraphEdge<Relation, MicroEvent> newProductEdge =
                                                  new ProductGraphEdge<Relation, MicroEvent>();
            newProductEdge.FirstGraphEdge = firstGraphCorrespondingEdge.EdgeContent;
            newProductEdge.SecondGraphEdge = secondGraphCorrespondingEdge.EdgeContent;
            newProductEdge.FirstGraphEdgeID = firstGraphCorrespondingEdge.EdgeID;
            newProductEdge.SecondGraphEdgeID = secondGraphCorrespondingEdge.EdgeID;
            productGraphEdges.Add(newProductEdge);
        }
      }
    }
    return productGraphEdges;
  }
}
```

Razred `EvExtraPipeliner`

Razred `EvExtraPipeliner` zadužen je za povezivanje svih modela i izgradnju grafa događaja za proizvoljno zadani ulazni tekst. Programski kôd ovog razreda ulančava izvršavanje modela za crpljenje sidara činjeničnih događaja (preko članske varijable `anchorExtractionProcess`), crpljenje argumenata događaja (preko članske varijable `argumentExtractor`), crpljenje vremenskih odnosa među događajima (preko članske varijable `relationExtractionProcess`) i razrješavanje koreferencije (preko članske varijable `coreferenceResolutionProcess`). Rezultat metode `EvExtraPipeline` jest objekt razreda `ProcessedArticle` koji efektivno sadrži graf događaja (liste svih spominjanja događaja i odnosa među njima).

```
public class EvExtraPipeliner
{
  private EventsLearningProcess anchorExtractionProcess;
  private TimeMLRelationsLearningProcess relationExtractionProcess;
  private ECBCoreferenceLearningProcess coreferenceResolutionProcess;
```

```
private EventArgsExtraction argumentExtractor;
public EvExtraPipeliner(string anchorexBinaryModelPath, string anchorexClassificationModelPath,
        string relexIdentModelPath, string relexClassModelPath, string corefModelPath)
{
  anchorExtractionProcess = new EventsLearningProcess(null);
  anchorExtractionProcess.PredictionModelPath = anchorexBinaryModelPath;
  anchorExtractionProcess.ClassPredictionModelPath = anchorexClassificationModelPath;
  argumentExtractor = new EventArgsExtraction();
  relationExtractionProcess = new TimeMLRelationsLearningProcess(null);
  relationExtractionProcess.IdentificationModelPath = relexIdentModelPath;
  relationExtractionProcess.ClassificationModelPath = relexClassModelPath;
  coreferenceResolutionProcess = new ECBCoreferenceLearningProcess(null);
  coreferenceResolutionProcess.PredictionModelPath = corefModelPath;
}
public ProcessedArticle EvExtraPipeline(string articleText, string fileName,
                                        ProcessedArticle gsGraph = null)
{
  EventExtraction.Core.DataContainers.Article article = new Core.DataContainers.Article();
  article.Text = articleText;
  // anchor extraction
  ProcessedArticle pArticle = ExtractMicroEvents(article);
  pArticle.FileName = fileName;
  // argument extraction
  argumentExtractor.ExtractArguments(pArticle);
  // temporal relation extraction
  ExtractTemporalRelations(pArticle);
  // event coreference resolution
  ResolveEventCoreference(pArticle);
  pArticle.CloseTransitively();
  return pArticle;
}
private ProcessedArticle ExtractMicroEvents(Article article)
{
  anchorExtractionProcess.Predict(new List<Core.DataContainers.Article> { article });
  return article.ToProcessedArticle();
}
private void ExtractTemporalRelations(ProcessedArticle pArticle)
{
  CreateCloseScopePossibleRelations(pArticle);
  relationExtractionProcess.Predict(pArticle);
```

```
  }
  private void ResolveEventCoreference(ProcessedArticle pArticle)
  {
    List<Tuple<EventCoreferenceInfo, ProcessedArticle, ProcessedArticle>> evCorefInfos =
                  new List<Tuple<EventCoreferenceInfo, ProcessedArticle, ProcessedArticle>>();
    for (int i = 0; i < pArticle.MicroEvents.Count − 1; i++)
    { for (int j = i + 1; j < pArticle.MicroEvents.Count; j++)
      { evCorefInfos.Add(new Tuple<EventCoreferenceInfo, ProcessedArticle, ProcessedArticle>(
        new EventCoreferenceInfo{ FirstMicroEvent = pArticle.MicroEvents[i],
              SecondMicroEvent = pArticle.MicroEvents[j], ContextConfidence = −2}, pArticle, null));
      }
    }
    coreferenceResolutionProcess.Predict(evCorefInfos)
    evCorefInfos.ForEach(x => {
      if (x.Item1.ContextConfidence == 1)
      {
        Relation relation = new Relation { FirstEvent = x.Item1.FirstMicroEvent,
                                           SecondEvent = x.Item1.SecondMicroEvent,
                                           RelationType = RelationType.Coreferent };
        pArticle.Relations.Add(relation);
      }
    });
  }
  ...
}
```

Dodatak B

Upute za označavanje

Ručno označene zbirke tekstova nužne su za razvoj i vrednovanje modela koji sudjeluju u izgradnji grafova događaja (v. poglavlje 5), ali i vrednovanje kakvoće automatski izgrađenih grafova događaja (v. poglavlje 7). Pri izgradnji ručno označenih zbirki tekstova vrlo je važno osigurati da svi označivači u potpunosti razumiju zadatak te da svi označavaju na isti način. Jedino je tako moguće izgraditi zbirku tekstnih podataka koja je kvalitetno (točno i dosljedno) označena. Kakvoća (u prvom redu dosljednost) ručno označenih tekstnih podataka ima presudnu ulogu za modele nadziranog strojnog učenja koje učimo na temelju ručno označenih primjera. Kako bi označivači razumjeli i uspješno obavili zadatak potrebno im je dati i objasniti upute za označavanje.

U ovom poglavlju prikazane su (1) upute za označavanje činjeničnih sidara događaja (v. odjeljak 5.1), (2) upute za označavanje argumenata događaja (v. odjeljak 5.2) te (3) upute za označavanje različitih odnosa među događajima (neke od označenih odnosa poput kauzalnosti i dijeljenih argumenata nisu korišteni u izgradnji grafova događaja) (v. poglavlje 5.3). Sve upute prikazane su u izvornom obliku u kojem su dane označivačima.

B.1 Upute za označavanje činjeničnih sidara događaja

Novinski tekstovi govore o događajima iz stvarnog svijeta. Događaji označavaju da je netko nešto napravio ili da se nešto dogodilo. Svaki događaj u stvarnom svijetu dogodio se u nekom trenutku u vremenu, stoga je svaki par događaja iz stvarnog svijeta moguće staviti u vremensku relaciju. U pisanom tekstu, ipak, nisu svi događaji precizno smješteni u vrijeme niti je moguće sve parove događaja staviti u vremensku relaciju.

U ovom zadatku označavanja potrebno je označiti događaje u novinskim tekstovima na engleskome jeziku kao i vremenske relacije među pojedinim parovima događaja. Dodatno, naknadno je potrebno označiti koji događaji u tekstu označavaju isti događaj stvarnog svijeta (koreferencija događaja). Tri zadatka označavanja, dakle, jesu:

1. označavanje događaja u tekstu
2. označavanje vremenskih relacija između parova događaja u tekstu
3. označavanje događaja u tekstu koji predstavljaju jedan te isti događaj u stvarnom svijetu (konsolidacija događaja)

U nastavku ovog dokumenta opisan je prvi zadatak. Druga dva zadatka rješavat će se u kasnijoj fazi.

B.1.1 Označavanje događaja

Različiti ljudi pod pojmom "događaj" podrazumijevaju različite stvari. U kontekstu ovog označavanja događajem se smatra svaka **STVARNA** akcija ili djelovanje, neovisno o svom vremenskom trajanju (događaji mogu biti trenutačni ili mogu trajati kroz duži period). Događaj je **STVARAN** ukoliko je jasno da se akcija doista dogodila ili se događa u stvarnom svijetu. Buduće ili hipotetske akcije ili djelovanja **NE** smatraju se događajima u okviru ovog označavanja. Događajima se, također, **NE** smatraju predikati koji opisuju stanja ili okolnosti u kojima nešto vrijedi, kao ni mentalna stanja, želje, namjere i sl. Kao događaji se **NE** smiju označavati niti radnje koje imaju ponavljajući karakter, kao što su navike, običaji i sl.

Napomena: U svim primjerima koji slijede riječi koje treba označiti kao događaje biti će označeni ***podcrtano, podebljano i u kurzivu***. Riječi koje su potencijalni kandidati za događaj, ali koje ipak (na temelju nekog od pravila definiranih ovim uputama) nećemo označiti kao događaj bit će označene ***podebljano i u kurzivu***. Riječi koje označavaju radnje o kojima se govori u nekom dijelu teksta, neovisno o tome radi li se o događaju koji treba označiti kao događaj ili samo o potencijalnom kandidatu, bit će označene sivom pozadinom.

Što označavati?

Potrebno je označavati **REALNE** događaje. To su događaji stvarnog svijeta koji su naznačeni u tekstu, a za koje je jasno da su se ili dogodili ili se događaju. Instance koje ćemo označavati u tekstu moraju predstavljati jedan konkretni događaj iz stvarnog svijeta koji ima svoj početak, trajanje i kraj. Instance koje podrazumijevaju više radnji (repetitivnost) neće se označavati kao događaji. Primjeri **REALNIH** događaja:

"Ferdinand Magellan, a Portuguese explorer, first ***reached*** the islands in search of spices."
"11,024 people, including Aeta aborigines, were ***evacuated*** to 18 disaster relief centers."
"The number of bodies ***discovered*** displayed the proportion of the ***massacre***."
"Israel has ***bought*** more masks abroad, after a ***shortage*** of several thousand gas masks."

Iako radnje uglavnom izražavamo glagolima, događaji mogu biti označeni i drugim vrstama riječi, kao što su imenice (npr. *massacre* ili *shortage* u prethodnim primjerima) ili pridjevi

(*discovered* u trećem primjeru).

Što NE označavati?

Ne označavaju se:

1. Hipotetski, budući i negirani događaji. Označavaju se samo događaji u tekstu za koje se pouzdano zna da su se ili dogodili ili se događaju. Primjeri hipotetskih i budućih događaja kakvi se **NEĆE** označavati su sljedeći:

   "Shiite militia leaders, who *might* be ***worried*** about becoming the targets."
   "I ***find*** it totally inappropriate that our children *may* ***grow*** up with this war continuing."
   "He ***said*** that if they *could not* ***reach*** Davos unhindered, they *would* ***demonstrate*** elsewhere"
   "US *will* ***display*** its power"

2. Izrazi koji označavaju stanja, mišljenja, razmatranja, želje, stavove i sl. Primjeri takvih izraza su sljedeći:

   "A Philippine volcano, ***dormant*** for six centuries, ***exploded*** with searing gases, thick ash and deadly debris."
   "There is no reason why we would not be ***prepared***."
   "John ***believes*** this could be the solution to the problem."
   "I ***prefer*** football over basketball."

3. Generička spominjanja koja ne označavaju konkretan događaj već općenit skup događaja nekog tipa. Primjeri takvih spominjanja su:

   "***Use*** of corporate jets for political ***travel*** is legal."
   "Jews are ***prohibited*** from killing one another."
   "When it comes to ***learning***, people ***disagree*** on the methodology."

4. Radnje koje imaju ponavljajući karakter i zapravo podrazumijevaju skup od više događaja istog tipa iz stvarnog svijeta. Takve radnje obično se odnose na navike ili običaje:

   "He usually ***swam*** the 4 km distance."
   "Jack Johnson used to ***work*** as a waiter."
   "Mycah ***sings*** at the 'The Cat' on Tuesdays."

Označavanje događaja s negacijama i drugim modifikatorima

U tekstu, negacija uz događaj označava da se nešto nije dogodilo. Međutim, takva negacija u tekstu može označavati dvije prilično različite situacije iz stvarnog svijeta.
U primjerima:

"Merkel and Berlusconi did *not* ***meet*** on Saturday" .
"The Rolling Stones did *not* ***visit*** New York during their last tour."

negacija označava izostanak događaja u stvarnom svijetu. Nešto se, dakle, nije dogodilo. Takve radnje **NEĆE** biti označene kao događaji. S druge strane, u primjerima:

"Despite this foothold, the Swedes were *not able* to ***hold*** the Russians."
"Despite all the effort, the goalkeeper simply *couldn't* ***make*** that last save."

negacija označava događaj u stvarnom svijetu (Rusi su probili švedsku obranu; igrač je zabio gol) koji je formuliran negacijom radnje koja je inverz one koja se stvarno dogodila. U ovakvim slučajevima radnju (*hold* odnosno *make* u prethodnim primjerima) **TREBA** označiti kao događaj.

Osim negacija, postoje i drugi modifikatori koji se mogu nalaziti uz radnju, a koji mogu utjecati na sigurnost procjene da se događaj uistinu dogodio u stvarnom svijetu. Takvi modifikatori su riječi poput *maybe*, *nearly*, *almost*, *certainly*, *surely* i sl. U takvim situacijama potrebno je procijeniti kakvu semantiku modifikator unosi, odnosno potrebno je procijeniti je li se događaj, uz semantiku koju unosi modifikator, ipak dogodio u stvarnom svijetu. Ukoliko je procjena da se uistinu dogodio u stvarnom svijetu, potrebno ga je označiti. U primjerima:

"John *most certainly* ***saw*** the guy who robbed the bank."
"Mary *surely* ***painted*** the fence this morning."

modifikatori *surely* i *most certainly* naznačuju da su se događaji inidicirani riječima koji ih prate uistinu dogodili u stvarnom svijetu. S druge strane, u primjerima:

"*Maybe* the hooligans already ***arrived*** in the morning."
"The Swedish defense *nearly* ***broke*** when their hero ***arrived***."

modifikatori *maybe* i *nearly* naznačuju kako se događaj koji ih slijedi nije sa sigurnošću dogodio (u prvom slučaju ne znamo sa sigurnošću je li se dogodio, dok u drugom slučaju znamo da se sigurno nije dogodio – skoro se dogodio, ali ipak nije). U takvim slučajevima radnju **NEĆEMO** označavati kao događaj.

Kako označavati?

Događaje u tekstu uvijek označavamo samo jednom riječju, koju nazivamo **sidrom** događaja:

1. Ako je događaj izražen kao glagolska fraza (npr. *has been scrambling*, *to buy*, *were reported*) potrebno je označiti samo glavnu riječ glagolskog izraza, kao što je to označeno na sljedećim primjerima:

   "Israel has been ***scrambling*** to buy more masks abroad."
   "After we had been ***attacked***, we…"

   Pomoćne glagole (razni oblici pomoćnih glagola *be* i *have* poput *was*, *were*, *have*, *has*, *had*, *been* i dr.) **NE** označavati kao dio događaja.

2. Ukoliko se radi o glagolskoj frazi koja uz glagol podrazumijeva i prijedlog (npr. *set up*, *take down*, *build upon*), potrebno je označiti samo glagolsku riječ unutar fraze, kako je prikazano donjim primjerima. **NE** označavati prijedlog koji se nalazi uz glagol kao dio događaja.

   "Additional distribution centers were ***set*** *up* last week."
   "Their love was ***built*** *upon* long lasting friendship and mutual understanding."

3. Ako je događaj izražen glagolskom frazom koja se sastoji od aspektnog glagola (npr. *begin*, *stop*, *end*, *keep*) i glavnog glagola, kao događaj je potrebno označiti samo glavni glagol, kao što je označeno u sljedećim primjerima:

   "The private sector ***began*** ***establishing*** a private agency."
   "US had ***stopped*** ***interfering*** in other countries policies long ago."

4. Ako se radi o imeničkoj frazi, tada je kao događaj potrebno označiti samo glavnu riječ fraze, kako je prikazano u primjeru:

   "The young industry's rapid ***growth*** is also attracting regulators eager to police its many facets."

5. Ako je događaj naveden imeničkom frazom popraćen s realnim glagolskim predikatom, potrebno je označiti oboje kao događaje, kako je prikazano sljedećim primjerima:

   "The young industry's rapid ***growth*** is also ***attracting*** regulators eager to police its many facets."
   "Several pro-Iraq ***demonstrations*** have ***taken*** place in the last week."

B.1.2 Određivanje vrste događaja

Svakom označenom događaju potrebno je dodijeliti i vrstu, odnosno razred. Događaji od interesa u okviru ovog označavanja pripadaju jednom od pet razreda: REPORTING, PERCEPTION, I-ACTION, OCCURRENCE. Slijedi detaljniji opis događaja koji pripadaju pojedinom razredu.

1. REPORTING – ovoj kategoriji pripadaju događaji koji opisuju akcije u kojima osobe ili organizacije nešto objavljuju, deklariraju ili informiraju javnost. Ovakvi događaji imaju narativni karakter. Primjeri glagola koji imaju narativni karakter jesu: *say*, *report*, *tell*, *explain*, *state*.

   "Punongbayan ***said*** that the 4,795-foot-high volcano was ***spewing*** gases up to 1,800 degrees."
   "Eight different injuries were ***reported*** over the weekend."
   "***Citing*** an example, John ***exclaimed*** . . ."

2. PERCEPTION – Ovom razredu pripadaju događaji koji uključuju fizičku percepciju nekoga ili nečega. Takvi događaji često su opisani glagolima poput: *see*, *watch*, *glimpse*, *view*, *hear*, *listen*, *overhear*.

   "Witnesses ***tell*** Birmingham police they ***saw*** a man ***running***."
   "John ***heard*** the thousands of small ***explosions*** down there."
   "I ***hear*** their children ***crying***."

3. I-ACTION – Događaji u ovom razredu predstavljaju "akciju s namjerom" (engl. *intentional action*). Događaj koji pripada ovom razredu otvara mjesto drugom događaju koji mora biti eksplicitno naveden u tekstu. Taj drugi događaj opisuje akciju ili situaciju iz koje nešto možemo zaključiti na temelju njegove relacije s događajem koji je tipa I-ACTION. Slijedi lista glagola i primjera događaja koji pripadaju ovom razred:

   attempt, ***try***, ***scramble***
   "Companies such as Microsoft are ***trying*** to ***monopolize*** Internet."
   "Israel has been ***scrambling*** to ***buy*** more masks abroad"

   investigate, ***look*** at, ***delve***
   "The Organization of African Unity ***investigates*** the Hutu-organized ***genocide*** of more than 500,000 minority Tutsis."
   "A new Essex County task force ***began*** ***delving*** Thursday into the ***slayings*** of 14 black women."

   avoid, ***prevent***, ***cancel***

"Palestinian police ***prevented*** a ***planned*** pro-Iraq ***rally*** by the Palestinian Professionals' Union"

promise, ***offer***, ***propose***, ***agree***, ***decide***
"Germany has ***agreed*** to ***lend*** Israel 180,000 protective kits against chemical and biological weapons, and Switzerland ***offered*** to ***give*** Israel another 25,000 masks."

Još neki glagoli koji tipično izažavaju karakter namjere i otvaraju mjesto drugom događaju su *swear*, *vow*, *name*, *nominate*, *appoint*, *declare*, *proclaim*, *claim*, *allege*, *suggest*. Zgodno je primijetiti kako je između događaja tipa I-ACTION događaja i druge radnje kojoj on otvara mjesto često prisutan prijedlog "*to*" (npr. "*ordered to assemble*", "*asked to delay*"). Taj prijedlog je dosta dobar indikator za događaje razreda I-ACTION.

4. STATECHANGE – ovom razredu pripadaju događaji koji označavaju da je došlo do promjene stanja nekog objekta ili osobe. Promjene stanja uključuju promjene fizičkih stanja, ali i promjene u phisičkim stanjima i razmišljanjima (npr. *realized*).

   "Because of the early minor ***defeats*** and the Jin formation, Fu Jian ***overestimated*** the amount of Jin forces."
   "***Realizing*** that they could not withstand another attack, the Greeks ***evacuated*** Corsica, and initially ***sought*** refuge in Rhegion in Italy."
   "The CBI Index ***rose*** 5%."

5. OCCURRENCE – Najveći broj događaja pripada ovome razredu. Ovaj razred obuhvaća sve događaje koji opisuju da se nešto dogodilo (*happens or occurs*). Primjeri za događaje iz razreda OCCURRENCE su "raznovrsniji" u odnosu na ostale razrede:

   "The Defense Ministry ***said*** 16 planes have ***landed*** so far with protective equipment against biological and chemical warfare."
   "Mordechai ***said*** all the gas masks from abroad have ***arrived*** and have been ***distributed*** to the public."
   "Two moderate ***eruptions*** shortly before 3 p.m. Sunday ***signaled*** a larger ***explosion***."

B.1.3 Dodatne napomene i primjeri

U slučaju događaja za koje u kontekstu prvog pojavljivanja nije jasno jesu li se zaista dogodili (i kao takvi neće biti označeni), a kasnije se utvrdi da su se doista dogodili, **NE** treba se vraćati na prethodnu (nesigurnu) instancu događaja i označiti je.

"He ***pushed*** me to ***contribute***. In the end I ***contributed*** the most of all."
"He ***said*** he would ***come***. He ***came*** eventually, but it was already half past 10."

Ovakvo označavanje omogućava nam da se fokusiramo na uži kontekst te ne moramo pamtiti semantiku cijelog teksta. Kad imamo par događaja koji su neposredno jedan pored drugoga, a koji zapravo referenciraju isti događaj te je jedan imeničkog, a drugi glagolskog karaktera potrebno je označiti oboje kao zasebne događaje, a pri tome glagolski događaj treba označiti razredom I-ACTION.

"We have ***used*** some of our cash to ***make*** (I-ACTION) great ***investments*** (OCCURRENCE) in our business"
"The ***demonstrations*** (OCCURRENCE) have ***taken*** (I-ACTION) place on the Red Square."

Primjeri

U sljedećim primjerima riječi koje trebaju biti označene kao događaji u okviru ovog označavanja označene su ***podebljano, u kurzivu i podcrtano***, dok su potencijalni događaji koje **NE** treba označiti u okviru ovog označavanja označeni samo ***podebljano i u kurzivu*** (nisu podcrtani).

"He was ***pushing*** me to the edge to ***contribute***."

Riječ "*pushing*" sidro je događaja (razred OCCURRENCE), dok riječ "*contribute*" to nije budući da se ne zna je li taj netko tko je bio potican uistinu i ostvario doprinos. Ne znamo je li se "*contribute*" doista dogodio.

"He was ***pushing*** me to the edge and it ***paid*** off."

Kao i u prethodnom primjeru riječ "*pushing*" je događaj razreda OCCURRENCE, baš kao i "*paid off*" (pri čemu je potrebno označiti samo "*paid*").

"...where the stakes are more often ***measured*** in frothy pints."

Riječ "*measured*" nije sidro događaj jer predstavlja uobičajenu, odnosno ponavljajuću radnju.

"...which has ***televised*** marquee tournaments."

Riječ "*televised*" sidro je događaja i to razreda I-ACTION jer uvodi mjesto za drugi događaj. Riječ "*tournaments*" popunjava to mjesto, ali ne predstavlja događaj jer se radi o više turnira (repetitivnost, nespecifičnost, generičnost).

"...which has ***televised*** commercials"

Riječ "*televised*" je događaj, ali ovoga puta razreda OCCURRENCE jer **NE** otvara mjesto drugom događaju, već imeničkom objektu.

"Mr. Taylor, the 12-time world champion, who ***practices*** six hours a day, ***said*** the game's boozy pedigree is unshakeable and a big part of its appeal."

Riječ "*practices*" **NIJE** događaj jer se radi o navici odnosno uobičajenoj radnji. Riječ "*said*" je tipični događaj razreda REPORTING.

"But I don't ***think*** it's a bad thing."

Riječ "*think*" **NIJE** događaj jer se radi o mišljenju odnosno stavu.

"...as he was ***pointing*** to the fact that ***drinking*** alcohol is not ***allowed*** during a match."

Riječ "*pointing*" sidro je događaja razreda REPORTING (netko izrečenime ukazuje na nešto). Riječ "*drinking*" nije sidro realnog događaja jer se radi o generičkoj radnji. Riječ "*allowed*" također nije sidro događaja jer predstavlja stanje (nešto nije dozovoljeno), a ne događaj s određenim vremenskim trajanjem.

"...because whenever he ***stepped*** on his home scale, as he ***put*** it, it ***read*** error."

Riječi "*stepped*" i "*read*" nisu sidra činjeničnih događaja zbog ponavljajuće prirode koju unosi modifikator "*whenever*" koji označava da se zapravo radi o više odvojenih događaja vaganja u stvarnom svijetu. Riječ "*put*" je događaj razreda REPORTING.

"To ***relax*** before a match, he used to ***drink*** 25 bottles of Holsten Pils."

Riječi "*relax*" i "*drink*" nisu sidra činjeničnih događaja, prva zbog generičnosti, a druga zbog repetitivnosti radnje koju predstavlja.

B.2 Upute za označavanje argumenata događaja

Novinski tekstovi govore o događajima iz stvarnog svijeta. Događaji označavaju da je netko nešto napravio ili da se nešto dogodilo. Osim samom radnjom (npr. kupovina, plesanje, ubojstvo) događaji su određeni sudionicima (protagonistima) te okolnostima odvijanja radnje (npr. vrijeme, mjesto, način). U ovom zadatku označavanja potrebno je označiti argumente događaja u tekstovima na engleskome jeziku. Sidra događaja (riječi koje nose temeljno značenje radnje ili aktivnosti događaja) već su označena.

Napomena: U svim primjerima koji slijede sidra događaja biti će označena ***podvučeno, podebljano i u kurzivu***. Riječi ili fraze koje predstavljaju semantičke argumente događaja bit će označeni **podebljano i u boji**, pri čemu boja upućuje na semantičku ulogu argumenta.

B.2.1 Vrste argumenata događaja

U okviru ovog označavanja označavat će se semantički argumenti događaja. Riječi ili fraze u tekstu koji predstavljaju semantičke argumente događaja mogu, ali i ne moraju biti sintaktički argumenti sidra događaja. Ovisno o primjeni, moguće je definirati mnoštvo različitih semantičkih uloga za argumente događaja. U ovom označavanju usredotočit ćemo se na one vrste semantičkih argumenata događaja koje su primjenjive u velikoj većini situacija i za veliku većinu događaja, a ujedno nose i najviše informacije o samome događaju (imaju najveći informacijski doprinos). Razmatrat će se sljedeće semantičke kategorije argumenata:

1. AGENT – označava izvršitelje, pokretače, odnosno nositelje radnje događaja. Semantička uloga AGENT često odgovara sintaktičkoj ulozi subjekta, no to ne mora nužno biti slučaj:

   "**Van Persie** ***scored*** his sixth goal of the season."
   "The only player to ***score*** six goals in the first five matches was **Robin Van Persie**."

2. TARGET – označava nekoga ili nešto prema komu ili čemu je radnja usmjerena, odnosno nekoga (ili nešto) tko trpi radnju. Semantički argument s ulogom TARGET često odgovara sintaktičkoj ulozi objekta, no to ne mora nužno biti slučaj:

   "Van Persie ***scored*** **his sixth goal** of the season."
   "**Shiite monks** were the target of the ***bombing***."

3. LOCATION – označava mjesto na kojem se radnja odvijala. Pri tome lokacijom smatramo kako lokacijske imenovane entitete (npr. gradove, države, regije), tako i ostale izraze koji imaju lokacijski karakter (mjesta, pozicije):

   "A huge building was ***blown*** up this morning **in Jakarta**"
   "A helpless two-months old baby was ***found*** **on the back seat of the abandoned car**."

4. TIME – označava vrijeme odvijanja događaja. Pri tome vremenskim argumentima događaja smatramo kako pune i potpuno vremenski određene oznake (npr. datumi) tako i relativne vremenske oznake:

   "Forty-three Northland Chiefs ***signed*** the treaty **in 1840**."
   "A huge building was ***blown*** up **this morning** in Jakarta."

B.2.2 Postupak označavanja

Kao argumente događaja označavamo samo one riječi i fraze koje se nalaze unutar iste rečenice sa sidrom događaja. Dijelove teksta koji se nalaze izvan rečenice u kojoj je sidro događaja **NEĆEMO** označavati kao argumente za te događaje. Primjer označavanja argumenata događaja:

"**Tobacco giant Philip Morris *launched* legal action** against **Australia's government on Monday** less than an hour after **Parliament *passed* legislation *banning* all logos** from cigarette packages."

Potrebno je prvo identificirati sve fraze koje su izravno vezane (tj. izravno se odnose) na sidro događaja i to na način da se maksimizira broj različitih argumenata. Dakle, ako smo u dvojbi predstavlja li neki izraz (najčešće imenička fraza) jedan ili dva argumenta, odlučit ćemo se za dva argumenta. U prethodnom primjeru mogli bismo biti u dvojbi oko izraza "*legal action against Australia's goverment*" odnosno u dvojbi oko toga odnosi li se "*against Australia's government*" na sidro "*action*" ili na sidro "*launched*". U ovom slučaju, oba su tumačenja moguća (sintaktička analiza rečenice je višeznačna). Međutim, kako preferiramo veći broj argumenata, odlučit ćemo se za tumačenje u kojem je izraz "*against Australia's government*" izravno vezan na sidro "*launched*" i time predstavlja argument za sebe, odnosno odvojen argument u odnosu na izraz "*legal action*".

Jednom kada smo u skladu s prethodnim naputkom odredili sve izraze za koje smatramo da su argumenti događaja (i koji pripadaju jednoj od četiri semantičke uloge koje nas zanimaju), trebamo označiti cijele (imeničke) fraze koje su argumenti događaja. Dakle, označavamo maksimalno prostiranje svakog argumenta. U prethodnom primjeru stoga smo kao agenta događaja "*launched*" označili cijeli frazu "*Tobacco giant Philip Morris*" umjesto, primjerice, samo "*Philip Morris*" ili samo "*Morris*". Također smo označili cijeli izraz "*legal action*" (umjesto samo "*action*"), kao i cijeli izraz "*Australia's government*" umjesto samo "*government*".

Ponekad je teško razlučiti gdje prestaje imenička fraza nekog argumenta. Pogledajmo sljedeći primjer:

"John ***saw*** **the yellow Maserati** that ***caused*** **the terrible *accident* a week earlier**" .

TARGET događaja "*saw*" jest izraz "*the yellow Maserati*". Međutim, zavisna rečenica u nastavku — "*that caused the terrible accident*" opisuje upravo "*Maserati*" tj. sintaktički gledano vezana je na riječ "*Maserati*". U skladu s tim, potencijalno zaključivanje bi moglo biti da se cijeli izraz "*the yellow Maserati that caused the terrible accident a week earlier*" treba označiti kao argument događaja "*saw*" s ulogom TARGET. Međutim, zavisno složena rečenica ima svoju vlastitu sintaktičku strukturu (vlastiti predikat), a u velikoj većini slučajeva imamo i druge događaje unutar te podrečenice (u primjeru su to "*caused*" i "*accident*"). Budući da ne želimo kao argumente imati cijele rečenice, u ovakvim slučajevima prostiranje argumenta prestaje s početkom zavisno složene rečenice koja ga dodatno opisuje.

Prethodni primjer ukazuje i na nerijetku situaciju u kojoj je jedan događaj ("*accident*") argument drugoga ("*caused*"). U takvim je situacijama događaj koji je argument drugog događaja najčešće imeničkog tipa.

Posebni slučajevi

Postoji nekoliko posebnih slučajeva koji zahtijevaju podrobnije razmatranje:

1. *Višestruka iskoristivost* – u nekim situacijama jedna te ista riječ ili fraza može biti argument dvaju ili više sidara događaja, a da pri tome ta riječ ili fraza nema istu ulogu za oba događaja (primjerice predstavlja argument s ulogom AGENT jednog i ulogom TARGET drugog događaja). Tako u primjeru:

   "**Tobacco giant Philip Morris *launched* legal action** against **Australia's government on Monday** less than an hour after **Parliament *passed* legislation *banning* all logos** from cigarette packages."

   izraz "*legislation*" predstavlja argument s ulogom TARGET za sidro događaja "*passed*", ali i argument s ulogom AGENTA za sidro događaja "*banning*".

2. *Distributivnost argumenata* — često je jedna te ista riječ ili fraza argument dvaju izravno povezanih događaja (npr. povezanost veznikom "*and*" ili sintaktička dominacija između sidara događaja), pri čemu kao argument ima istu ulogu za oba događaja (AGENT obaju događaja):

   "**Woody Allen *produced*** and ***directed* the movie**."
   (konjunkcija događaja — "*Woody Allen*" je ujedno agent događaja "*produced*" i događaja "*directed*"; isto tako izraz "*the movie*" je argument s ulogom TARGET i događaja "*produced*" i događaja "*directed*")

   "**Putin *rejected* the Syria peace plan**, ***calling* it** another provocation by the State Department."
   (sintaktička dominacija između događaja – "*Putin*" je argument s ulogom AGENT i za događaj "*rejected*" i za događaj "*calling*")

3. *Distributivnost događaja* – moguće je da jedan događaj ima više argumenata istog tipa (npr. više argumenata s ulogom AGENT ili TARGET) koji su u pravilu sintaktički povezani. Iako bi u nekim slučajevima bilo moguće cijeli izraz označiti kao jedan argument, preferiramo razbijanje na više argumenata istog tipa ukoliko je to moguće smisleno napraviti. U primjeru

   "**Woody Allen** and **Billy Crystal *directed*** the movie."

   imamo dva agenta ("*Woody Allen*" i "*Billy Crystal*") za događaj "*directed*". Iako smo cijelu imeničku frazu "*Woody Allen and Billy Crystal*" mogli označiti kao jednog agenta događaja "*directed*", ovakve ćemo slučajeve rješavati razdvajanjem u više argumenata istog tipa jer je takvo tumačenje bliže semantici događaja u stvarnom svijetu (gdje su

zaista dva različita čovjeka sudjelovala u tom događaju).

Dvojbe i konkretni primjeri

Nekoliko dvojbi pojavilo se nakon prve runde označavanja. Problematični slučajevi su sistematizirani te su ove upute proširene na način da obuhvaćaju i te slučajeve.

1. **Označavanje prijedloga i članova**

   Prijedloge i članove (ako postoje) označavat ćemo kao dio argumenata s ulogama LOCATION i TIME.

   "The bomb that ***exploded*** **in the western part of Damascus** **on Monday morning**..."

   Članove (ali ne i prijedloge) označavamo i kao dio argumenata s ulogama AGENT i TARGET:

   "The ***objection*** by **Romney** against **the healthcare reforms**..."

2. **Odnosna zamjenica kao subjekt**

   Kada imamo situaciju u kojoj je subjekt događaja odnosna zamjenica (*who*, *that* ili *which*), a izvorni argument (na kojeg se zamjenica odnosi) se nalazi u istoj rečenici, kao argument ćemo označiti izvorni izraz. Ovo će tipično biti slučaj za argumente tipa AGENT.

   "**Mozart**, who started ***playing*** at the age of four"

   Međutim, kada se ne radi o odnosnoj zamjenici, već ulogu sintaktičkog subjekta ima osobna zamjenica (*he*, *she*, *it*), pa kao argument tipa AGENT treba označiti osobnu zamjenicu.

   "**Mozart** started ***playing*** **at the age of four** and it is **at that age** that **he** ***developed*** **passion for music**."

3. **Količinski izrazi**

   Kod burzovnih tekstova često se možemo naći u dvojbi je li neki količinski izraz (postotak) argument tipa TARGET. Takvi izrazi, međutim odgovaraju na pitanje "koliko", a ne "koga" ili "što" i stoga ih nećemo označavati kao argumente tipa TARGET.

   "**CP&M index** ***rose*** 3.2 percent **this morning** **on the NY Stock Exchange**, while **D&M** ***fell*** 5 points **in Japan**."

4. **Argument koji sadrži zarez**

Interpunkcija (u prvom redu zarez) ne mora nužno prekinuti prostiranje argumenta. Postavite si pitanje je li moguće bez promjene značenja poništiti inverziju, tj. dio koji je odvojen zarezom staviti ispred preostalog dijela izraza i ukloniti zarez, te predstavlja li tada cijeli izraz jedan argument.

"**China Merchant Holdings, a major port operator**, ***announced*** **yesterday** that..."

Prethodnu rečenicu možemo prepisati tako da poništimo inverziju, a da pritom ne promijenimo značenje rečenice:

"**A major port operator China Merchant Holdings** ***announced*** **yesterday** that..."

5. **Lokacije u širem smislu**

   Lokacijskim argumentima smatrat ćemo sve izraze koji definiraju nekakav prostor odnosno prostiranje, makar ne označavaju nužno prostor koji možemo "uzemljiti" u smislu pridjeljivanja geografskih koordinata (ulice, trgove, adrese, gradove itd.).

   "**The man** was ***shot*** [**in the chest**] [**at the scene**]."

   U prethodnom primjeru imamo dva lokacijska argumenta (koji su, radi jasnoće, odvojeni uglatim zagradama). Prvi argument, "*in the chest*", ne možemo "uzemljiti" u smislu geografskih koordinata, ali svejedno taj izraz smatramo lokacijskim argumentom jer definira nekakvo prostiranje. Drugi argument, "*at the scene*", klasičan je primjer lokacijskog argumenta koji je moguće "uzemljiti" u smislu pridjeljivanja geografskih koordinata.

6. **Vremenski odnos između događaja ili vremenska oznaka jednog događaja**

   Ponekad je teško razlikovati izraze koji predstavljaju vremenske oznake pridijeljene pojedinačnom događaju od izraza koji predstavljaju vremenski odnos između dvaju različitih događaja. Izraze koji označavaju vremenske odnose između događaja **NE** želimo pridjeljivati kao vremenske argumente tim događajima.

   "**The** ***robbery*** ***happened*** **three hours ago**."
   "**The** ***robbery*** ***happened*** three hours before **the treasurer** ***became*** aware of it."

   U prvoj rečenici izraz "*three hours ago*" označavamo kao vremenski argument budući da se on odnosi na vrijeme odvijanja pojedinačne radnje ("*robbery*" odnosno "*happened*") te ne određuje vremenski odnos s nekim drugim događajem. U drugoj rečenici izraz "*three hours before*" ne definira vrijeme kad se događaj "*robbery*" odvio već definira (i kvantificira) relativni vremenski odnos između događaja "*robbery*" i "*became*". Kao pravilo za donošenje odluke kod ovakve dvojbe poslužite se sljedećim trikom: pokušajte odgovoriti

na pitanje "*Kad se odvio događaj?*". Dotični izraz treba označiti kao vremenski argument događaja samo ako je moguće dati informacijski potpun odgovor koristeći isključivo taj izraz (i ne koristeći druge događaje).

B.3 Upute za označavanje odnosa među događajima

Novinski tekstovi govore o događajima iz stvarnog svijeta. Događaji označavaju da je netko nešto napravio ili da se nešto dogodilo. Događaji se u stvarnom svijetu ne odvijaju izolirano i nepovezano već su u različitim odnosima s drugim događajima (npr. vremenski odnos – događaj A dogodio se prije događaja B ili kauzalni odnos – događaj A uzrokovao je događaj B). Odnosi u kojima se nalaze događaji u stvarnom svijetu preslikavaju se (više ili manje jasno i jednoznačno) i u tekst gdje su događaji iz stvarnog svijeta opisani.

U ovom zadatku označavanja potrebno je označiti različite odnose između događaja u tekstovima na engleskome jeziku. Sidra događaja (riječi koje nose temeljno značenje radnje ili aktivnosti događaja) kao i pripadni argumenti već su označeni.

Napomena: U svim primjerima koji slijede sidra događaja biti će označena ***podcrtano, podebljano i u kurzivu***.

B.3.1 Vrste odnosa između događaja

U okviru ovog označavanja označavat će se četiri grupe odnosa između događaja koji donose najviše informacije o međusobnim odnosima događaja: (1) vremenski odnosi među događajima, (2) kauzalni odnosi među događajima, (3) odnos istovjetnosti (tj. koreferencije) događaja i (4) odnosi dijeljenih argumenata.

Vremenski odnosi

Razmatraju se četiri različite vremenske relacije:

1. **Prije(A, B)** – označava da se događaj A odvio prije događaja B. Preciznije, ovaj vremenski odnos znači da je događaj A završio prije nego je događaj B započeo;
2. **Poslije(A, B)** – označava da se događaj A odvio poslije događaja B. Preciznije, ovaj vremenski odnos označava da je događaj A započeo nakon što je događaj B završio.

   Odnosi *prije* i *poslije* međusobno su inverzni, tj. **Prije(A, B) = Poslije(B, A)**. Prilikom označavanja dopušteno je ravnopravno koristiti odnose *prije* i *poslije*, ali pri tome treba voditi računa koji je događaj odabran kao prvi događaj za odnos. Odnose *prije/poslije* ponekad je moguće jednostavno prepoznati na temelju jasno izraženih indikatora (npr. uporaba vremenskih prijedloga "*before*" ili "*after*").

"Tear gas was ***fired*** after protesters ***managed*** to ***breach*** a barbed wire cordon surrounding the palace."
Prije(managed, fired) ili **Poslije(fired, managed)**

Međutim, puno češće odnos *prije/poslije* nije tako eksplicitno naznačen u tekstu, već je implicitno određen prirodom događaja.

"Many of those ***gathered*** outside the palace, in the suburb of Heliopolis, ***chanted*** slogans against the regime."
Prije(gathered, chanted) ili **Poslije(chanted, gathered)**

U prethodnom primjeru znamo da su se prosvjednici prvo morali okupiti kako bi mogli početi pjevati prosvjedne slogane. Ponekad je vremenski odnos moguće implicitno iščitati iz vremenskih oznaka pridruženih događajima:

"Many of those ***gathered*** outside the palace, in the suburb of Heliopolis, ***chanted*** slogans similar to those directed against the regime of former president Hosni Mubarak during the ***uprising*** in February 2011."
Prije(uprising, gathered) ili **Poslije(gathered, uprising)** i
Prije(uprising, chanted) ili **Poslije(chanted, uprising)**

U prethodnom primjeru naznačeni vremenski odnosi proizlaze iz vremenske oznake "*February 2011*" pridružene događaju "*uprising*" i konteksta iz kojeg se vidi da su događaji "*gathered*" i "*chanted*" recentni.

3. **Istovremeno(A, B)** – ovaj odnos označava da su događaji A i B počeli u (otprilike) isto vrijeme te su u isto vrijeme i završili. Iz teksta je najčešće vrlo teško točno odrediti trenutke početka i završetka nekog događaja, pa dva događaja smatramo istovremenima ukoliko su im počeci i svršeci približno podudarni. Istovremenost događaja ponekad je moguće prepoznati na temelju izraza koji eksplicitiraju takav odnos.

   "John ***cried*** and ***laughed*** at the same time"
   Istovremeno(cried, laughed)
   "The rocket launch and the air attacks went on simoultaneously"
   Istovremeno(launch, attacks)

   Puno češće, međutim, takvi eksplicitni indikatori vremenskog odnosa istovremenosti nisu prisutni u tekstu. Odnos istovremenosti u načelu je nešto teže prepoznati od odnosa *prije/poslije* jer nije lako razlikovati odnos istovremenosti od nekih oblika odnosa vremenskog preklapanja. Ipak, u situacijama gdje smo skloni vjerovati da su se (po nekakvoj prirodi stvari) dva događaja odvijala otprilike istovremeno označit ćemo takav vremenski

odnos, čak i kada nemamo eksplicitni dokaz za takvu odluku. Drugim riječima, treba nam jači argument da se radi o nekom drugom vremenskom odnosu (tj. dokaz da događaji nisu istovremeni) nego da se radi o istovremenosti.

"Large crowds ***protested*** outside as night fell, while thousands of demonstrators also ***gathered*** in Cairo's Tahrir Square."
Istovremeno(protested, gathered)
"School classes were suspended in many cities, and dozens of flights were canceled."
Istovremeno(suspended, canceled)

U prvom prethodnom primjeru nemamo u tekstu jamstvo da su ljudi koji su "protestirali vani kako je noć padala" to činili u točno istom vremenskom periodu dok su se "drugi skupljali na trgu Tahrir" (eventualno prijedlog "*while*" daje nekakvu naznaku istovremenasti). Drugim riječima ne možemo znati točan početak i završetak kako "protestiranja" tako ni "skupljanja". Međutim, nekako je prirodno zamisliti da se te radnje odvijaju istovremeno. Također, u drugom primjeru nemamo jamstvo da se otkazivanje nastave u školama dogodilo u točno istom trenutku kada i otkazivanje letova, ali u kontekstu informacije koju daje članak možemo pretpostaviti istovremenost (odnosno, ako postoji nekakvo vremensko odstupanje, ono je za informaciju koju članak daje nebitno).
Kod označavanja sidara događaja imali smo slučajeve u kojima su u tekstu dva događaja postojala zato jer je takva bila fraza, iako se u stvarnom svijetu dogodila samo jedna stvar (npr. "*accident happened*" ili "*managed to breach*"). Kod takvih smo situacija jedan događaj označavali razredom I-ACTION, tj. označavali smo da on otvara mjesto drugom događaju. U takvim ćemo situacijama označavati vremenski odnos istovremenosti (a kasnije i koreferenciju takvih događaja jer se radi o jednom te istom događaju u stvarnom svijetu):

"Tear gas was ***fired*** after protesters ***managed*** to ***breach*** a barbed wire cordon surrounding the palace."
Istovremeno(managed, breach)
"An accident occurred on the corner of the Fifth and Vermouth."
Istovremeno(accident, occurred)

4. **Preklapanje(A, B)** – ovaj vremenski odnos označava da događaji A i B imaju preklapanje u vremenskom trajanju, ali nisu potpuno vremenski podudarni. Ovaj vremenski odnos obuhvaća nekoliko "finijih" vremenskih odnosa (npr. događaji istovremeno počeli, ali je jedan završio ranije od drugoga ili je pak jedan događaj započeo i završio za vrijeme trajanja drugoga) i upravo je zbog te činjenice takve vremenske odnose možda i najteže prepoznati u tekstu. Evo nekoliko primjera:

"The ***summit*** began with Merkel ***addressing*** the parliament."
Preklapanje(summit, addressing)
"Putin was ***interrupted*** in the middle of his ***talk***."
Preklapanje(interrupted, talk)

Slučajeve u kojima jedan događaj označava početak ili kraj drugog događaja označavat ćemo vremenskim odnosom preklapanja:

"The ***war*** officially ***ended*** in May 1995."
Preklapanje(war, ended)
"Everything was shaking when the ***quake*** ***started***."
Preklapanje(quake, started)

Kauzalni odnosi

Ova vrsta odnosa označava da je jedan događaj uzrok, a drugi događaj posljedica. Označavat ćemo dvije različite kauzalne relacije:

1. **Uzrokuje(A, B)** – ovaj odnos naznačava da je događaj A uzrokovao događaj B, odnosno da je početak događaja B izazvan ostvarenjem događaja A;
2. **Uzrokovan_s(A, B)** – ovaj odnos naznačava da je događaj A uzrokovan događajem B, odnosno da je početak događaja A izazvan ostvarenjem događaja B.

Odnosi **Uzrokuje** i **Uzrokovan_s** međusobno su inverzni, tj. **Uzrokuje(A, B) = Uzrokovan_s(B, A)**. Prilikom označavanja dopušteno je ravnopravno koristiti odnose *uzrokuje* i *uzrokovan_s*, ali pri tome treba voditi računa koji je događaj odabran kao prvi događaj za odnos.

"Typhoon Bopha ***struck*** on the large southern island of Mindanao. School classes were ***suspended*** in many cities, and dozens of flights were ***cancelled***."
Uzrokuje(struck, suspended) ili **Uzrokovan_s(suspended, struck)** i
Uzrokuje(struck,canceled) ili **Uzrokovan_s(canceled, struck)**

Treba primijetiti kako kauzalni odnos u pravilu implicira i vremenski odnos *prije/poslije*. Odnos **Uzrokuje(A, B)** u pravilu znači da vrijedi i odnos **Prije(A, B)**, a odnos **Uzrokovan_s(A, B)** u pravilu znači da vrijedi i odnos **Poslije(A, B)**.

Istovjetnost događaja

Istovjetnost (tj. koreferencija) događaja znači da se dva spominjanja događaja u tekstu odnose na isti događaj u stvarnome svijetu. U sljedećim su primjerima označeni samo međusobno koreferentni događaji:

"President Obama has welcomed world leaders to Camp David near Washington for the G8 ***summit***. Friday evening was arguably the more straightforward part of the ***summit***."
Istovjetni(summit, summit)
"Israel has ***fired*** missiles on the offices of the Palestinian Authority causing 7 deaths with many injuries. Israel helicopter gunships ***launched*** missiles across the Gaza Strip for more than two hours. The ***attack*** in Gaza has been said to cause more violence in Gaza and West Bank."
Istovjetni(fired, launched)
Istovjetni(fired, attack)
Istovjetni(launched, attack)

Odnosi dijeljenih argumenata

Dva događaja mogu imati podudarne (koreferentne) argumente. Pri tome ćemo označavati posebno dijeljene protagoniste (podudaranja u argumentima tipa AGENT ili TARGET), a posebno dijeljene lokacije (dva događaja koja su se dogodila na istom mjestu). U primjerima će biti označeni samo parovi događaja koji dijele protagoniste.

"Bush ***visited*** Panama on Tuesday. The ***protests*** against Bush spread immediately throughout the country."
Dijeljeni_protagonist (visited, protests)

Primijetimo da u prethodnom primjeru protagonist nema istu ulogu za oba događaja. Za događaj "*visit*" protagonist "*President Bush*" je AGENT, dok je "*Bush*" u ulozi TARGET za događaj "*protests*". Dakle, nije nužno da protagonist bude u istoj ulozi za oba događaja. Zajednički protagonist, u jednom od događaja može biti izražen anaforično odnosno upotrebom zamjenice. U takvim slučajevima ćemo isto htjeti označiti relacije dijeljenih protagonista.

"Obama ***rejected*** Republicans' initiative on South America. He ***explained*** his position at a press conference this morning."
Dijeljeni_protagonist(rejected, explained)
"Brazilian authorities ***arrested*** dozens of police officers on Tuesday, accusing them of taking bribes from drug traffickers. The 61 officers were ***paid*** to ***turn*** a blind eye to criminal activity in Brazil's Rio de Janeiro state."
Dijeljeni_protagonist(arrested, paid)
Dijeljeni_protagonist(arrested, turn)
Dijeljeni_protagonist(paid, turn)

Uz zajedničke protagoniste, označavat ćemo i parove događaja koji imaju zajedničku lokaciju. Pri tome lokacije mogu, ali i ne moraju biti imenovani entiteti.

"Tropical storm Katrina ***rushed*** through New Orleans ***crushing*** down buildings in the city."
Dijeljena_lokacija(rushed, crushing)

Lokacija događaja "*rushed*" u prethodnom primjeru jest imenovani entitet "*New Orleans*" dok je lokacija događaja "*crushing*" izraz "*in the city*". Međutim, kako je "*the city*" koreferentno spominjanje od "*New Orleans*", događaji imaju istu lokaciju. Događaji "*rushed*" i "*crushing*" imaju, naravno, i zajedničkog protagonista, a to je "*Tropical storm Katrina*".

"There were deadly ***clashes*** between pro- and anti-Morsy demonstrators outside the palace. Supporters and critics of Morsy ***hurled*** Molotov cocktails, rocks and fireworks at each other in front of the palace."
Dijeljena_lokacija(clashes, hurled)

U prethodnom primjeru vidimo da se događaji "*clashes*" i "*hurled*" odvijaju "ispred palače", a budući da imamo jasnu indikaciju da se radi o istoj palači, događaji imaju istu lokaciju .

B.3.2 Metodologija označavanja

Jasno je da je praktički neizvedivo razmatrati sve moguće parove događaja unutar dokumenta te razmatrati za svaku od vrsta relacija je li takvu relaciju moguće uspostaviti između događaja promatranog para.

Kod označavanja ćemo se voditi pretpostavkom koherentnosti teksta, odnosno pretpostaviti da dokument neće govoriti o više nepovezanih tema. Ova pretpostavka je u novinskim tekstovima u pravilu zadovoljena.

Slijedno čitanje i označavanje uz pretpostavku koherentnosti teksta

Označavanje ćemo provoditi čitajući tekst slijedno te označavajući sve odnose koje takvim slijednim prolaskom kroz tekst uočimo. Prirodno, veći broj relacija uočavat ćemo između međusobno bližih parova događaja. To, međutim, ne predstavlja problem, već je u skladu s pretpostavkom kohrerentnog teksta. Za očekivati je da odnosi (pogotovo vremenski i uzročno-posljedični) uistinu postoje između bližih (u smislu udaljenosti u tekstu) događaja jer pretpostavljamo da je autor teksta isti oblikovao upravo na način da se događaji koji su međusobno povezani nalaze blizu jedan drugoga (u suprotnom bi tekst bilo teže razumjeti, odnosno bila bi narušena pretpostavka koherentnog teksta).

Povezanost svih događaja u tekstu

Pretpostavka koherentnosti teksta također implicira da bi na neki način svi događaji u tekstu trebali biti povezani (u terminima grafova koji se prikazuju prilikom označavanja – ne bi trebalo

biti međusobno nepovezanih podgrafova). Stoga ćemo nastojati (ako je to moguće) uspostaviti neke relacije između „glavnih“ događaja pojedinih podcjelina kao što su to, primjerice, odlomci. Ukoliko tekst uistinu nije koherentan i različiti odlomci govore o nepovezanim stvarima (odnosno nemoguće je odrediti relacije među događajima iz različitih odlomaka) veze između takvih nepovezanih događaja nećemo forsirati (odnosno izmišljati) samo da bi graf bio povezan.

Svojstvo tranzitivnosti nekih odnosa

Vremenske odnose *prije*/*poslije*/*istovremeno* u nekim je slučajevima moguće izvesti automatski iz prethodno označenih odnosa. Konkretno, relacije *prije*/*poslije*/*istovremeno* imaju svojstvo tranzitivnosti. Za te odnose vrijede sljedeća pravila tranzitivnosti:

- **Prije(A, B)** i **Prije(B, C)** povlači da vrijedi i **Prije(A,C)**;
- **Poslije(A, B)** i **Poslije(B, C)** povlači da vrijedi **Poslije(A, C)**;
- **Istovremeno(A, B)** i **Istovremeno(B, C)** povlači da vrijedi **Istovremeno(A, C)**;
- **Prije(A, B)** i **Istovremeno(B, C)** povlači da vrijedi **Prije(A, C)**;
- **Poslije(A, B)** i **Istovremeno(B, C)** povlači da vrijedi **Poslije(A, C)**.

Isto svojstvo tranzitivnosti vrijedi i za odnose *uzrokuje*/*uzorokovan_s*, *dijeljena_lokacija* te *istovjetnost*:

- **Uzrokuje(A, B)** i **Uzrokuje(B, C)** povlači da vrijedi i **Uzrokuje(A, C)**;
- **Uzrokovan_s(A, B)** i **Uzrokovan_s(B, C)** povlači da vrijedi i **Uzrokovan_s(A, C)**;
- **Istovremeni(A, B)** i **Istovremeni(B, C)** povlači da vrijedi i **Istovremeni (A, C)**;
- **Dijeljena_lokacija (A, B)** i **Dijeljena_lokacija (B, C)** povlači da vrijedi i **Dijeljena_lokacija(A,C)**.

Primijetite da svojstvo tranzitivnosti **NE VRIJEDI** za odnos *dijeljeni_protagonist* budući da događaj može imati više protagonista, pa događaji A i B mogu dijeliti jednog protagonista, događaji B i C drugoga, a da A i C nemaju niti jednog zajedničkog protagonista. Odnose koje je moguće izvesti iz drugih označenih odnosa temeljem svojstva tranzitivnosti (jedno od gore navedenih pravila) ne morate nužno označavati. Ipak, označite li takav odnos, to je potpuno u redu. Ukoliko niste sigurni slijedi li neki odnos tranzitivno iz već označenih onda ga svakako eksplicitno označite (po načelu “od viška glava ne boli”).

Dodatne veze promatranjem prethodno označenih argumenata događaja

Nakon slijednog prolaska kroz tekst i označavanja svih uočenih odnosa, ponekad je moguće uočiti dodatne odnose (koje su nam promakle pri prolasku kroz tekst) gledajući događaje i njihove argumente (promatrajući prikazani graf događaja):

1. Uspoređujući vremenske argumente koje imaju neki događaji možemo utvrditi vremenski odnos među njima;

2. Uspoređujući argumente s ulogom AGENT/TARGET možemo utvrditi da neki događaji imaju dijeljene protagoniste;
3. Uspoređujući lokacijske argumente događaja možemo ponekad utvrditi da neki događaji imaju dijeljene lokaciju.

B.3.3 Primjer označavanja

"Tanks and armored personnel carriers ***rolled*** into the area near the presidential palace Thursday, ***trying*** to bring some calm to the country's latest center of turmoil. Piles of rubble and burned cars ***littered*** the streets. The front doors of nearby storefronts were ***smashed*** in.

Five people have been ***killed*** and 446 ***injured*** in deadly ***clashes*** between pro- and anti-Morsy demonstrators outside the palace, the Egyptian health ministry ***said*** Thursday. At least 35 police officers are among the injured, the state-run MENA news agency ***reported***.

Clashes flared Wednesday and early Thursday after a week of largely peaceful ***protests*** in Cairo. Supporters and critics of Morsy ***hurled*** Molotov cocktails and ***threw*** rocks and fireworks at each other in front of the palace."

Prepoznati odnosi su: **Istovremeno(rolled, trying)**
Istovremeno(rolled, littered)
Istovremeno(littered, smashed)
Istovremeno(rolled, smashed) (ovaj odnos nije nužno morao biti označen jer se može tranzitivno izvesti na temelju prethodna dva odnosa)
Dijeljena_lokacija(rolled, killed)
Dijeljena_lokacija(killed, injured)
Dijeljena_lokacija(injured, clashes)
Dijeljena_lokacija(killed, clashes) (ovaj odnos nije nužno morao biti označen jer se može tranzitivno izvesti na temelju prethodna dva odnosa)
Preklapanje(clashes, killed)
Preklapanje(clashes, injured)
Istovremeno(killed, injured)
Poslije(said, killed)
Poslije(said, injured) (ovaj odnos nije nužno morao biti označen jer se može tranzitivno izvesti na temelju prethodna dva odnosa)
Poslije(reported, killed)
Poslije(reported, injured) (ovaj odnos nije nužno morao biti označen jer se može tranzitivno

izvesti iz **Istovremeno(killed, injured)** i **Poslije(reported, killed)**)
Istovjetnost(Clashes, flared)
Istovremenost(Clashes, flared) (nije nužno označavati jer istovjetnost implicira istovremenost)
Istovjetnost(Clashes (treći odlomak)**, clashes** (drugi odlomak)**)**
Prije(Clashes (treći odlomak)**, reported)** (primjer "povezivanja odlomaka")
Poslije(Clashes, protests)
Poslije(flared, protests) (nije nužno označavati jer slijedi tranzitivno iz **Istovjetni(Clashes, flared)** i **Poslije(Clashes, protests)**)
Preklapanje(Clashes, hurled)
Preklapanje(Clashes, threw)
Istovremeno(hurled, threw)
Dijeljeni_protagonist(hurled, threw)
Poslije(hurled, protests)
Poslije(threw, protests) (nije nužno označavati jer slijedi tranzitivno iz **Istovremeno(threw, hurled)** i **Poslije(hurled, protests)**)
Dijeljena_lokacija(threw, hurled)
Dijeljena_lokacija(rolled (prvi odlomak)**, hurled** (treći odlomak)**)** (primjer odnosa koji je teško uočiti slijednim prolaskom kroz tekst)
Dijeljena_lokacija(rolled (prvi odlomak)**, threw**(treći odlomak)**)** (nije nužno označiti jer slijedi tranzitivno iz prethodne dvije)

Dodatak C

Skupovi tekstnih podataka

Na ovom su mjestu pobrojani i ukratko opisani svi skupovi podataka koji su izgrađeni u okviru istraživanja opisanog ovom disertacijom.

- Zbirka `EvExtra` zbirka je novinskih tekstova s ručno označenim sidrima činjeničnih događaja te njihovim semantičkim razredima. Zbirka sadrži 759 novinskih članaka odnosno ukupno preko 330,000 pojavnica, što ju trenutno čini najvećom zbirkom s ručno označenom događajima u literaturi (v. poglavlje 5.1). Zbirka je javno dostupna na adresi:
  `http://takelab.fer.hr/grapheve/evextra-corpus.rar`
- Zbirka ručno označenih grafova događaja sadrži 105 novinskih članaka iz zbirke EvExtra za koje su ručno označeni cjelokupni grafovi događaja (sidra događaja, argumente događaja, vremenske odnose i koreferenciju između događaja) (v. poglavlje 5.3). Zbirka je javno dostupna na adresi:
  `http://takelab.fer.hr/grapheve/grapheve-goldgraphs.rar`
- Skup podataka za prepoznavanje parova dokumenata koji opisuju istovjetne događaje sastoji se od 10 grupa u kojima se nalaze novinski članci koji opisuju iste događaje. Grupe su prikupljene putem internetske usluge EMM NewsBrief te su dodatno ručno pročišćene (v. poglavlje 8.2.1). Ovaj je skup podataka javno dostupan na adresi:
  `http://takelab.fer.hr/evkernels/evkernels-dataset-recognizing.rar`
- Skup podataka za rangiranje parova dokumenata prema sličnosti događaja sadrži 60 parova dokumenata od kojih su 30 parovi koji opisuju iste događaje, a 30 parovi koji opisuju različite događaje (v. poglavlje 8.2.2). Skup podataka javno je dostupan na adresi:
  `http://takelab.fer.hr/evkernels/evkernels-dataset-ranking.rar`
- Dvije su ispitne zbirke korištene za vrednovanje modela za pretraživanje informacija. Obje zbirke sadrže po 50 upita. Prva zbirka sastoji se od 25,948 tematski različitih novinskih članaka, dok druga zbirka sadrži 1,387 novinskih članaka koji se tiču građanskog rata u Siriji (v. poglavlje 8.3.1). Obje ispitne zbirke sadrže oznake relevantnosti dokumenata za sve upite. Ispitne zbirke javno su dostupne na adresi:

http://takelab.fer.hr/retrographs/rgir-collections-clear.rar

- Skup podataka korišten za automatsko vrednovanje čitljivosti tekstova pojednostavljenih temeljem događaja sastoji se od 100 novinskih članaka i pripadnih im pojednostavljenih inačica (v. poglavlje 9.2.3).[1] Izvorni i pojednostavljeni novinski članci dostupni su na adresi:
  http://takelab.fer.hr/data/evsimplify/EventSimplify.rar
- Skup podataka korišten za ručno vrednovanje gramatičnosti i značajnosti sadržaja tekstova pojednostavljenih temeljem događaja sastoji se od 280 parova tekstnih odsječaka, pri čemu se svaki par sastoji od izvornog tekstnog odsječka i pojednostavljenja nekom od vrednovanih metoda (v. poglavlje 9.2.3). Parovi izvornih i pojednostavljenih tekstnih odsječaka također su dostupni na adresi:
  http://takelab.fer.hr/data/evsimplify/EventSimplify.rar

[1] Skupovi podataka koji služe vrednovanju postupaka za automatizirano pojednostavljivanje novinskih tekstova temeljenih na događajima izgrađeni su u suradnji s istraživačima sa Sveučilišta u Wolverhamptonu.

Literatura

ACE. 2005. The ACE 2005 (ACE05) Evaluation Plan: Evaluation of the Detection and Recognition of ACE Entities, Values, Temporal Expressions, Relations, and Events.

ACE. 2007. The ACE 2007 (ACE07) Evaluation Plan: Evaluation of the Detection and Recognition of ACE Entities, Values, Temporal Expressions, Relations, and Events.

Agić, Ž. 2012a. K-best Spanning Tree Dependency Parsing with Verb Valency Lexicon Reranking. *Str. 1–12 u: Proceedings of 24th International Conference on Computational Linguistics (COLING 2012)*.

Agić, Ž. 2012b. *Pristupi ovisnosnom parsanju hrvatskih tekstova*. Doktorska disertacija, Sveučilište u Zagrebu.

Agirre, E. i Soroa, A. 2009. Personalizing PageRank for Word Sense Disambiguation. *Str. 33–41 u: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09)*. Association for Computational Linguistics.

Agirre, E., Diab, M., Cer, D. i Gonzalez-Agirre, A. 2012. Semeval-2012 task 6: A Pilot on Semantic Textual Similarity. *Str. 385–393 u: Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Ahn, D. 2006. The Stages of Event Extraction. *Str. 1–8 u: Proceedings of COLING/ACL 2006 Workshop on Annotating and Reasoning about Time and Events*.

Allan, J. 2002. *Topic Detection and Tracking: Event-Based Information Organization*. The Information Retrieval Series, vol. 12. Springer.

Allen, J. F. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, **26**(11), 832–843.

Aluísio, S. M., Specia, L., Pardo, T. A. S., Maziero, E. G. i Fortes, R. P. M. 2008. Towards Brazilian Portuguese Automatic Text Simplification Systems. *Str. 240–248 u: Proceedings of the eighth ACM symposium on Document engineering*. New York, NY, USA: ACM.

Amati, G. 2003. *Probability Models for Information Retrieval Based on Divergence from Randomness*. Doktorska disertacija, Sveučilište u Glasgowu.

Aone, C. i Ramos-Santacruz, M. 2000. REES: A Large-Scale Relation and Event Extraction System. *Str. 76–83 u: Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics.

Artstein, R. i Poesio, M. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, **34**(4), 555–596.

Atkinson, M. i Van der Goot, E. 2009. Near Real Time Information Mining in Multilingual News. *Str. 1153–1154 u: Proceedings of the 18th International Conference on World Wide Web*. ACM.

Bagga, A. i Baldwin, B. 1999. Cross-Document Event Coreference: Annotations, Experiments, and Observations. *Str. 1–8 u: Proceedings of the Workshop on Coreference and its Applications*. Association for Computational Linguistics.

Baker, C. F., Fillmore, C. J. i Lowe, J. B. 1998. The Berkeley Framenet Project. *Str. 86–90 u: Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics.

Barlacchi, G. i Tonelli, S. 2013. ERNESTA: A Sentence Simplification Tool for Children's Stories in Italian. *Str. 476–487 u: Proceedings of Computational Linguistics and Intelligent Text Processing*. Springer.

Barzilay, R., McKeown, K. R. i Elhadad, M. 1999. Information Fusion in the Context of Multi-Document Summarization. *Str. 550–557 u: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics.

Bejan, C. A. i Hathaway, C. 2007. UTD-SRL: A Pipeline Architecture for Extracting Frame Semantic Structures. *Str. 460–463 u: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*.

Bejan, C. i Harabagiu, S. 2008. A Linguistic Resource for Discovering Event Structures and Resolving Event Coreference. *Str. 2881–2887 u: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*.

Bejan, C. i Harabagiu, S. 2010. Unsupervised Event Coreference Resolution with Rich Linguistic Features. *Str. 1412–1422 u: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*. Association for Computational Linguistics.

Bell, A. 1991. *The Language of News Media*. Blackwell Oxford.

Bennett, J. F. 1988. *Events and Their Names*. Hackett publishing.

Bethard, S. 2008. *Finding Event, Temporal and Causal Structure in Text: A Machine Learning Approach*. Doktorska disertacija, Sveučilište u Boulderu, Kolorado.

Bethard, S. 2013. ClearTK-TimeML: A Minimalist Approach to TempEval 2013. *Str. 10–14 u: Second Joint Conference on Lexical and Computational Semantics (* SEM)*, vol. 2.

Boguraev, B. i Ando, R. K. 2005. TimeBank-Driven TimeML Analysis. *Annotating, Extracting and Reasoning about Time and Events*.

Borgwardt, K. M. 2007. *Graph Kernels*. Doktorska disertacija, Ludwig-Maximilians-Universität München.

Bramsen, P., Deshpande, P., Lee, Y. K. i Barzilay, R. 2006. Inducing Temporal Graphs. *Str. 189–198 u: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*. Association for Computational Linguistics.

Brand, M. 1977. Identity Conditions for Events. *American Philosophical Quarterly*, **14**(4), 329–337.

Bruce, B. C. 1972. A Model for Temporal References and its Application in a Question Answering Program. *Artificial intelligence*, **3**, 1–25.

Bunke, H. i Allermann, G. 1983. Inexact Graph Matching for Structural Pattern Recognition. *Pattern Recognition Letters*, **1**(4), 245–253.

Burstein, J., Shore, J., Sabatini, J., Lee, Y. i Ventura, M. 2007. The Automated Text Adaptation Tool. *Str. 3–4 u: Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. NAACL-Demonstrations '07. Stroudsburg, PA, USA: Association for Computational Linguistics.

Büttcher, S., Clarke, C. L., Yeung, P. C. i Soboroff, I. 2007. Reliable Information Retrieval Evaluation with Incomplete and Biased Judgements. *Str. 63–70 u: Proc. of the ACM SIGIR*. ACM.

Carbonell, J., Yang, Y., Lafferty, J., Brown, R. D., Pierce, T. i Liu, X. 1999. CMU Report on TDT-2: Segmentation, Detection and Tracking. *Str. 117–120 u: Proceedings of the DARPA Broadcast News Workshop*.

Carreras, X. i Màrquez, L. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. *Str. 152–164 u: Proceedings of the Ninth Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

Carretti, B., Belacchi, C. i Cornoldi, C. 2010. Difficulties in Working Memory Updating in Individuals with Intellectual Disability. *Journal of Intellectual Disability Research*, **54**(4), 337–345.

Carroll, J., Minnen, G., Pearce, D., Canning, Y., Devlin, S. i Tait, J. 1999. Simplifying Text for Language-Impaired Readers. *Str. 269–270 u: Proceedings of the 9th Conference of the European Chapter of the ACL (EACL'99)*.

Casati, R. i Varzi, A. C. 2008. Event concepts. *Understanding Events. From Perception to Action, Oxford*, 31–53.

Castañeda, H.-N. 1967. Comments. *U:* Rescher, N. (ed), *The Logic of Decision and Action*. University of Pittsburgh Press.

Chambers, N. i Jurafsky, D. 2008. Unsupervised Learning of Narrative Event Chains.

Chambers, N. i Jurafsky, D. 2009. Unsupervised Learning of Narrative Schemas and Their Participants. *Str. 602–610 u: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics.

Chang, A. X. i Manning, C. D. 2012. SUTIME: A Library for Recognizing and Normalizing Time Expressions. *U: Proceedings of 8th International Conference on Language Resources and Evaluation (LREC 2012)*.

Chang, C. C. i Lin, C. J. 2011. LibSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, 1–27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chen, B., Su, J., Pan, S. i Tan, C. 2011. A Unified Event Coreference Resolution by Integrating Multiple Resolvers. *U: Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*.

Chomsky, N. 2000. *New Horizons in the Study of Language and Mind*. Cambridge University Press.

Chowdhury, G. 2010. *Introduction to Modern Information Retrieval*. Facet publishing.

Christensen, J., Mausam, S. S. i Etzioni, O. 2013. Towards Coherent Multi-Document Summarization. *Str. 1163–1173 u: Proceedings of NAACL-HLT*.

Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and psychological measurement*, **20**(1).

Cohen, J. 1968. Weighted Kappa: Nominal Scale Agreement Provision for Scaled Disagreement or Partial Credit. *Psychological bulletin*, **70**(4), 213–220.

Conroy, J. M., Schlesinger, J. D., Goldstein, J. i O'leary, D. P. 2004. Left-Brain/Right-Brain Multi-Document Summarization. *U: Proceedings of the Document Understanding Conference (DUC 2004)*.

Cordella, L. P., Foggia, P., Sansone, C. i Vento, M. 2004. A Subgraph Isomorphism Algorithm for Matching Large Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(10), 1367–1372.

Cortes, C. i Vapnik, V. 1995. Support-Vector Networks. *Machine learning*, **20**(3), 273–297.

Dang, H. T. i Owczarzak, K. 2008. Overview of the TAC 2008 Update Summarization Task. *Str. 1–16 u: Proceedings of Text Analysis Conference*.

Daniel, N., Radev, D. i Allison, T. 2003. Sub-Event Based Multi-Document Summarization. *Str. 9–16 u: Proceedings of the HLT-NAACL 03 on Text Summarization Workshop, Volume 5*. Association for Computational Linguistics.

Davidson, D. 1967. The Logical Form of Action Sentences. *Essays on Actions and Events*, **5**, 105–148.

De Marneffe, M. C., MacCartney, B. i Manning, C. D. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. *Str. 449–454 u: Proceedings of 5th International Conference on Language Resources and Evaluation (LREC 2006)*, vol. 6.

De Marneffe, M.-C. i Manning, C. D. 2008. Stanford Typed Dependencies Manual. *URL http://nlp. stanford. edu/software/dependencies manual. pdf*.

Derczynski, L. i Gaizauskas, R. 2013. Temporal Signals Help Label Temporal Relations. *U: Proceedings of the annual meeting of the Association for Computational Linguistics, ACL*, vol. 78.

Devlin, S. 1999. *Simplifying Natural Language Text for Aphasic Readers*. Doktorska disertacija, Sveučilište u Sunderlandu, Velika Britanija.

Devlin, S. i Unthank, G. 2006. Helping Aphasic People Process Online Information. *Str. 225–226 u: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. Assets '06. New York, NY, USA: ACM.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. i Weischedel, R. M. 2004. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. *U: LREC*. Citeseer.

Dowty, D. R. 1989. On the Semantic Content of the Notion of Thematic Role. *Str. 69–129 u: Properties, Types and Meaning*. Springer.

Drndarević, B., Štajner, S., Bott, S., Bautista, S. i Saggion, H. 2013. Automatic Text Simplication in Spanish: A Comparative Evaluation of Complementing Components. *Str. 488–500 u: Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics. Lecture Notes in Computer Science. Samos, Greece, 24-30 March, 2013.*

Du, P., Guo, J., Zhang, J. i Cheng, X. 2010. Manifold Ranking with Sink Points for Update Summarization. *Str. 1757–1760 u: Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM.

Ehrlich, M., Remond, M. i Tardieu, H. 1999. Processing of Anaphoric Devices in Young Skilled and Less Skilled Comprehenders: Differences in Metacognitive Monitoring. *Reading and Writing*, **11**(1), 29–63.

Fan, R. E., Chang, K., Hsieh, C. J., Wang, X. R. i Lin, C. J. 2008. LibLinear: A Library for Large Linear Classification. *The Journal of Machine Learning Research*, **9**, 1871–1874.

Fellbaum, C. 2010. *WordNet*. Springer.

Feng, L. 2009. Automatic Readability Assessment for People with Intellectual Disabilities. *Str. 84–91 u: SIGACCESS Access. Comput.* New York, NY, USA: ACM.

Filatova, E. i Hatzivassiloglou, V. 2004. Event-Based Extractive Summarization. *U: Proceedings of ACL Workshop on Summarization*, vol. 111.

Fillmore, C. J. 1967. The Case for Case.

Fillmore, C. J. 1976. Frame Semantics and the Nature of Language. *Annals of the New York Academy of Sciences*, **280**(1), 20–32.

Finkel, J. R., Grenager, T. i Manning, C. D. 2005. Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling. *Str. 363–370 u: Proceedings of the*

43rd Annual Meeting on Association for Computational Linguistics (ACL '05). Association for Computational Linguistics.

Fiscus, J. G. i Doddington, G. R. 2002. Topic Detection and Tracking Evaluation Overview. *Str. 17–31 u: Topic detection and tracking*. Springer.

Freyhoff, G., Hess, G., Kerr, L., Tronbacke, B. i Van Der Veken, K. 1998. *Make it Simple, European Guidelines for the Production of Easy-to-Read Information for People with Learning Disability*. ILSMH European Association, Brussels.

Gael, J. V., Teh, Y. W. i Ghahramani, Z. 2008. The Infinite Factorial Hidden Markov Model. *Str. 1697–1704 u: Advances in Neural Information Processing Systems*.

Gaizauskas, R., Humphreys, K., Cunningham, H. i Wilks, Y. 1995. University of Sheffield: Description of the LaSIE System as Used for MUC-6. *Str. 207–220 u: Proceedings of the 6th conference on Message understanding*. Association for Computational Linguistics.

Gao, J., Nie, J.-Y., Wu, G. i Cao, G. 2004. Dependence Language Model for Information Retrieval. *Str. 170–177 u: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.

Garey, M. R. i Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman and Company, New York.

Gärtner, T., Flach, P. i Wrobel, S. 2003. On Graph Kernels: Hardness Results and Efficient Alternatives. *Str. 129–143 u: Learning Theory and Kernel Machines*. Springer.

Ge, J., Huang, X. i Wu, L. 2003. Approaches to Event-Focused Summarization Based on Named Entities and Query Words. *U: Proceedings of the 2003 Document Understanding Workshop*.

Gibson, J. J. 1975. Events are Perceivable but Time is Not. *Str. 295–301 u: The study of time II*. Springer.

Gildea, D. i Jurafsky, D. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, **28**(3), 245–288.

Glavaš, G. i Šnajder, J. 2013a. Event-Centered Information Retrieval Using Kernels on Event Graphs. *Str. 1–5 u: Graph-Based Methods for Natural Language Processing (TextGraphs-8)*.

Glavaš, G. i Šnajder, J. 2013b. Exploring Coreference Uncertainty of Generically Extracted Event Mentions. *Str. 408–422 u: Proceedings of the Conference in Intelligent Text Processing and Computational Linguistics CICLing 2013*. Springer.

Glavaš, G. i Štajner, S. 2013. Event-Centered Simplification of News Stories. *Str. 71–78 u: Proceedings of the Student Research Workshop associated with RANLP*.

Glavaš, G., Karan, M., Šarić, F., Šnajder, J., Mijić, J., Šilić, A. i Dalbelo Bašić, B. 2012. CroNER: A State-of-the-Art Named Entity Recognition and Classification for Croatian. *Str. 73–78 u: Information Society – Language Technologies Conference*.

Glavaš, G. i Šnajder, J. 2013. Recognizing Identical Events with Graph Kernels. *Str. 797–803 u: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*.

Glavaš, G. i Šnajder, J. 2014a. Constructing and Evaluating Event Graphs. *Natural Language Engineering*, **20**, U tisku.

Glavaš, G. i Šnajder, J. 2014b. Event Graphs for Information Retrieval and Multi-Document Summarization. *Expert Systems with Applications*, **41**, U recenziji.

Glavaš, G., Šnajder, J., Kolomiyets, O., Kordjamshidi, P. i Moens, M.-F. 2014. HiEve: A Corpus for Extracting Event Hierarchies from News Stories. *Str. u postupku objave u: Proceedings of 9th International Conference on Language Resources and Evaluation (LREC 2014)*.

Goodman, N. 1966. *The Structure of Appearance*. Vrin.

Gower, J. C. i Ross, G. 1969. Minimum Spanning Trees and Single Linkage Cluster Analysis. *Applied Statistics*, 54–64.

Grishman, R. i Sundheim, B. 1996. Message Understanding Conference-6: A Brief History. *Str. 466–471 u: Proceedings of International Conference on Computational Linguistics (COLING 1996)*, vol. 96.

Grishman, R. 2003. Information extraction. *The Handbook of Computational Linguistics and Natural Language Processing*, 515–530.

Grover, C., Tobin, R., Alex, B. i Byrne, K. 2010. Edinburgh-LTG: TempEval-2 System Description. *Str. 333–336 u: Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*. Association for Computational Linguistics.

Hacker, P. 1982. Events, Ontology and Grammar. *Philosophy*, **57**(222), 477–486.

Haghighi, A., Toutanova, K. i Manning, C. D. 2005. A Joint Model for Semantic Role Labeling. *Str. 173–176 u: Proceedings of the Ninth Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., i dr. 2009. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. *Str. 1–18 u: Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics.

Hammack, R., Imrich, W. i Klavžar, S. 2011. *Handbook of Product Graphs*. Discrete Mathematics and Its Applications. CRC Press.

Hatzivassiloglou, V., Gravano, L. i Maganti, A. 2000. An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering. *Str. 224–231 u: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Haussler, D. 1999. *Convolution Kernels on Discrete Structures*. Tech. rept. Technical report, Department of Computer Science, Sveučilište u Kaliforniji, Santa Cruz.

He, R., Qin, B. i Liu, T. 2012. A Novel Approach to Update Summarization Using Evolutionary Manifold-Ranking and Spectral Clustering. *Expert Systems with Applications*, **39**(3), 2375–2384.

Hiemstra, D. 2001. *Using Language Models for Information Retrieval*. Taaluitgeverij Neslia Paniculata.

Humphreys, K., Gaizauskas, R. i Azzam, S. 1997. Event Coreference for Information Extraction. *Str. 75–81 u: Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*. Association for Computational Linguistics.

Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H. i Wilks, Y. 1998. University of Sheffield: Description of the LaSIE-II System as Used for MUC-7. *U: Proceedings of the Seventh Message Understanding Conferences (MUC-7)*.

Inui, K., Fujita, A., Takahashi, T., Iida, R. i Iwakura, T. 2003. Text Simplification for Reading Assistance: A Project Note. *Str. 9–16 u: Proceedings of the second international workshop on Paraphrasing - Volume 16*. PARAPHRASE '03. Stroudsburg, PA, USA: Association for Computational Linguistics.

Jans, B., Bethard, S., Vulić, I. i Moens, M. F. 2012. Skip N-Grams and Ranking Functions for Predicting Script Events. *Str. 336–344 u: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Karttunen, L. i Zaenen, A. 2005. Veridicity. *Annotating, Extracting and Reasoning about Time and Events*.

Kawahara, D., Shinzato, K., Shibata, T. i Kurohashi, S. 2013. Precise Information Retrieval Exploiting Predicate-Argument Structures. *U: Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP 2013)*. 37–45.

Kim, J. 1966. On the Psycho-Physical Identity Theory. *American Philosophical Quarterly*, **3**(3), 227–235.

Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L. i Chissom, B. S. 1975. *Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel*. Tech. rept. DTIC Document.

Kingsbury, P. i Palmer, M. 2002. From Treebank to PropBank. *U: LREC*. Citeseer.

Kingsbury, P. i Palmer, M. 2003. PropBank: The Next Level of Treebank. *U: Proceedings of Treebanks and lexical Theories*, vol. 3.

Klein, D. i Manning, C. D. 2003. Accurate Unlexicalized Parsing. *Str. 423–430 u: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics.

Knight, K. i Marcu, D. 2002. Summarization beyond Sentence Extraction: A Probabilistic Approach to sentence compression. *Artificial Intelligence*, **139**, 91–107.

Kolomiyets, O., Bethard, S. i Moens, M. 2012. Extracting Narrative Timelines as Temporal Dependency Structures. *U: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*.

Koomen, P., Punyakanok, V., Roth, D. i Yih, W.-t. 2005. Generalized Inference with Multiple Semantic Role Labeling Systems. *Str. 181–184 u: Proceedings of the Ninth Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

Kumaran, G. i Allan, J. 2004. Text Classification and Named Entities for New Event Detection. *Str. 297–304 u: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Kurland, O. i Lee, L. 2010. PageRank Without Hyperlinks: Structural Reranking Using Links Induced by Language Models. *ACM Transactions on Information Systems (TOIS)*, **28**(4), 18.

Ladkin, P. B. 1987. Models of Axioms for Time Intervals. *Str. 234–239 u: AAAI*, vol. 87.

Lal, P. i Ruger, S. 2002. Extract-based Summarization with Simplification. *U: Proceedings of the ACL 2002 Automatic Summarization / DUC 2002 Workshop*.

Lavie, A. i Denkowski, M. J. 2009. The METEOR Metric for Automatic Evaluation of Machine Translation. *Machine Translation*, **23**(2-3), 105–115.

Lee, C., Lee, G. G. i Jang, M.-G. 2006. Dependency Structure Applied to Language Modeling for Information Retrieval. *ETRI journal*, **28**(3), 337–346.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M. i Jurafsky, D. 2011. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. *Str. 28–34 u: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics.

Lewis, D. 1986. *On the Plurality of Worlds*. Vol. 322. Cambridge Univ Press.

Li, W., Wu, M., Lu, Q., Xu, W. i Yuan, C. 2006. Extractive Summarization Using Inter-and Intra-event Relevance. *Str. 369–376 u: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Ligozat, G. 1990. Weak Representations of Interval Algebras. *Str. 715–720 u: AAAI*.

Lin, C.-H., Yen, C.-W., Hong, J.-S., Cruz-Lara, S., i dr. 2007. Event-Based Textual Document Retrieval by Using Semantic Role Labeling and Coreference Resolution. *U: IADIS International Conference WWW/Internet 2007*.

Lin, C.-Y. 2004. Rouge: A Package for Automatic Evaluation of Summaries. *Str. 74–81 u: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

Lin, C.-Y. i Hovy, E. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. *Str. 495–501 u: Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics.

Llorens, H., Saquete, E. i Navarro, B. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. *Str. 284–291 u: Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Llorens, H., Saquete, E. i Navarro-Colorado, B. 2013. Applying Semantic Knowledge to the Automatic Processing of Temporal Expressions and Events in Natural Language. *Information Processing & Management*, **49**(1), 179–197.

Lowe, S. A. 1999. The Beta-Binomial Mixture Model and Its Application to TDT Tracking and Detection. *Str. 127–131 u: Proceedings of DARPA Broadcast News Workshop*.

Mahé, P., Ueda, N., Akutsu, T., Perret, J.-L. i Vert, J.-P. 2005. Graph Kernels for Molecular Structure-Activity Relationship Analysis with Support Vector Machines. *Journal of Chemical Information and Modeling*, **45**(4), 939–951.

Maisonnasse, L., Gaussier, E. i Chevallet, J.-P. 2007. Revisiting the dependence language model for information retrieval. *Str. 695–696 u: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Makkonen, J., Ahonen-Myka, H. i Salmenkivi, M. 2004. Simple Semantics in Topic Detection and Tracking. *Information Retrieval*, **7**(3), 347–368.

Mani, I. 2001. *Automatic Summarization*. Vol. 3. John Benjamins Publishing.

Manning, C. D., Raghavan, P. i Schütze, H. 2008. *Introduction to Information Retrieval*. Vol. 1. Cambridge University Press Cambridge.

Marcus, M. P., Marcinkiewicz, M. A. i Santorini, B. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational linguistics*, **19**(2), 313–330.

McLaughlin, G. H. 1969. SMOG Grading: A New Readability Formula. *Journal of Reading*, **12**(8), 639–646.

Melli, G., Wang, Y., Liu, Y., Kashani, M. M., Shi, Z., Gu, B., Sarkar, A. i Popowich, F. 2006. Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task. *DUC*.

Menchetti, S., Costa, F. i Frasconi, P. 2005. Weighted Decomposition Kernels. *Str. 585–592 u: Proceedings of the 22nd International Conference on Machine Learning*. ACM.

Metzler, D. i Croft, W. B. 2005. A Markov Random Field Model for Term Dependencies. *Str. 472–479 u: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B. i Grishman, R. 2004. The NomBank Project: An Interim Report. *Str. 24–31 u: HLT-NAACL 2004 workshop: Frontiers in corpus annotation*.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., i dr. 2011. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, **331**(6014), 176–182.

Mihalcea, R. i Tarau, P. 2004. TextRank: Bringing Order Into Texts. *U: Proc. of the EMNLP 2004*, vol. 4. Barcelona, Spain.

Miller, G. A., Leacock, C., Tengi, R. i Bunker, R. T. 1993. A Semantic Concordance. *Str. 303–308 u: Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics.

Moens, M.-F. 2006. *Information Extraction: Algorithms and Prospects in a Retrieval Context*. Vol. 21. Springer.

Moldovan, D., Clark, C. i Harabagiu, S. 2005. Temporal Context Representation and Reasoning. *Str. 1099 u: International Joint Conference on Artificial Intelligence*, vol. 19. Citeseer.

Montague, R. 1969. On the Nature of Certain Philosophical Entities. *The Monist*, **53**(2), 159–194.

Montes-y Gómez, M., López-López, A. i Gelbukh, A. 2000. Information Retrieval with Conceptual Graph Matching. *Str. 312–321 u: Database and Expert Systems Applications*. Springer.

Moreda, P., Llorens, H., Saquete, E. i Palomar, M. 2011. Combining Semantic Information in Question Answering Systems. *Information Processing & Management*, **47**(6), 870–885.

Mourelatos, A. P. 1978. Events, Processes, and States. *Linguistics and Philosophy*, **2**(3), 415–434.

Nagel, C., Evjen, B., Glynn, J., Watson, K. i Skinner, M. 2012. *Professional C# 2012 and. Net 4.5*. John Wiley & Sons.

Navigli, R. i Lapata, M. 2007. Graph Connectivity Measures for Unsupervised Word Sense Disambiguation. *Str. 1683–1688 u: Proceedings of the 20th International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers Inc.

Nenkova, A. i Vanderwende, L. 2005. The Impact of Frequency on Summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*.

Orasan, C., R., E. i Dornescu, I. 2013. *Towards Multilingual Europe 2020: A Romanian Perspective*. Romanian Academy Publishing House, Bucharest. Chap. Text Simplification for People with Autistic Spectrum Disorders, str. 287–312.

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C. i Lioma, C. 2006. Terrier: A High Performance and Scalable Information Retrieval Platform. *Str. 18–25 u: Proceedings of the OSIR Workshop*.

Ouyang, Y., Li, W., Li, S. i Lu, Q. 2011. Applying Regression Models to Query-Focused Multi-Document Summarization. *Information Processing & Management*, **47**(2), 227–237.

Over, P. i Liggett, W. 2002. Introduction to DUC-2002: an Intrinsic Evaluation of Generic News Text Summarization Systems. *U: Proceedings of Workshop on Automatic Summarization (DUC 2002)*.

Over, P. i Yen, J. 2004. Introduction to DUC-2004: an Intrinsic Evaluation of Generic News Text Summarization Systems. *U: Proceedings of Workshop on Automatic Summarization (DUC 2004)*.

Page, L., Brin, S., Motwani, R. i Winograd, T. 1999. The PageRank Citation Ranking: Bringing Order to the Web.

Palmer, M., Gildea, D. i Xue, N. 2010. Semantic Role Labeling. *Synthesis Lectures on Human Language Technologies*, **3**(1), 1–103.

Park, J. H., Croft, W. B. i Smith, D. A. 2011. A Quasi-Synchronous Dependence Model for Information Retrieval. *Str. 17–26 u: Proc. of the 20th ACM International Conference on Information and Knowledge Management*. ACM.

Parsons, T. 1987. Underlying States in the Semantical Analysis of English. *Str. 13–30 u: Proceedings of the Aristotelian Society*, vol. 88. JSTOR.

Parsons, T. 1989. The Progressive in English: Events, States and Processes. *Linguistics and Philosophy*, **12**(2), 213–241.

Parsons, T. 1991. Tropes and Supervenience. *Philosophy and Phenomenological Research*, **51**(3), 629–632.

Pazienza, M. T. 1997. *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*. Springer.

Pianesi, F. i Varzi, A. 2000. Events and Event Talk: An Introduction. *Speaking of events*, 3–47.

Pimperton, H. i Nation, K. 2010. Suppressing Irrelevant Information from Working Memory: Evidence for Domain-Specific Deficits in Poor Comprehenders. *Journal of Memory and Language*, **62**(4), 380–391.

Ponte, J. i Croft, B. 1998. A Language Modeling Approach to Information Retrieval. *Str. 275–281 u: Proc. of the ACM SIGIR*. ACM.

Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R. i Xue, N. 2011. CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. *Str. 1–27 u: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task (CoNLL 2011)*. Association for Computational Linguistics.

Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D. i Ferro, L. 2003a. The TimeBank corpus. *Str. 40 u: Corpus Linguistics*, vol. 2003.

Pustejovsky, J., Castano, J., Ingria, R., Sauri, R., Gaizauskas, R., Setzer, A., Katz, G. i Radev, D. 2003b. TimeML: Robust Specification of Event and Temporal Expressions in Text. *New Directions in Question Answering*, **2003**, 28–34.

Quine, W. V. 1950. Identity, Ostension, and Hypostasis. *The Journal of Philosophy*, **47**(22), 621–633.

Quine, W. V. O. 1986. *Philosophy of Logic*. Harvard University Press.

Quine, W. 1985. Events and Reification. *Actions and Events*, 162–171.

Quinton, A. 1979. Objects and Events. *Mind*, **88**(1), 197–214.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D. i Manning, C. 2010. A Multi-Pass Sieve for Coreference Resolution. *Str. 492–501 u: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ramsey, F. P. i Moore, G. E. 1927. Facts and Propositions. *Proceedings of the Aristotelian Society, Supplementary Volumes*, **7**, 153–206.

Resnik, P. 1999. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, **11**, 95–130.

Robertson, S. E. i Jones, K. S. 1976. Relevance Weighting of Search Terms. *Journal of the American Society for Information science*, **27**(3), 129–146.

Ruppenhofer, J., Ellsworth, M., Petruck, M. R., Johnson, C. R. i Scheffczyk, J. 2010. *FrameNet II: Extended Theory and Practice*. Berkeley.

Saggion, H., Gómez Martínez, E., Etayo, E., Anula, A. i Bourg, L. 2011. Text Simplification in Simplext: Making Text More Accessible. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural*.

Saggion, H. i Gaizauskas, R. 2004. Multi-Document Summarization by Cluster/Profile Relevance and Redundancy Removal. *Str. 6–7 u: Proceedings of the Document Understanding Conference*.

Salton, G., Wong, A. i Yang, C. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, **18**(11), 613–620.

Salton, G. i Buckley, C. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Information processing & management*, **24**(5), 513–523.

Šarić, F., Glavaš, G., Karan, M., Šnajder, J. i Bašić, B. D. 2012. TakeLab: Systems for Measuring Semantic Text Similarity. *Str. 441–448 u: Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (*SEM 2012).*

Šarić, F., Šnajder, J. i Bašić, B. D. 2012. Optimizing Sentence Boundary Detection for Croatian. *Str. 105–111 u: Text, Speech and Dialogue*. Springer.

Saurí, R. i Pustejovsky, J. 2012. Are You Sure That This Happened? Assessing the Factuality Degree of Events in Text. *Computational Linguistics*, **38**(2), 261–299.

Schölkopf, B. i Smola, A. J. 2002. *Learning with Kernels*. MIT Press.

Schölkopf, B., Burges, C. J. i Smola, A. J. 1999. *Advances in Kernel Methods: Support Vector Learning*. MIT press.

Schultz, J. M. i Liberman, M. 1999. Topic Detection and Tracking Using IDF-Weighted Cosine Coefficient. *Str. 189–192 u: Proceedings of the DARPA broadcast news workshop*. San Francisco: Morgan Kaufmann.

Shapiro, A. i Milkes, A. 2004. Skilled Readers Make Better Use of Anaphora: A Study of the Repeated-Name Penalty on Text Comprehension. *Electronic Journal of Research in Educational Psychology*, **2**(2), 161–180.

Shinzato, K., Shibata, T., Kawahara, D. i Kurohashi, S. 2012. Tsubaki: An Open Search Engine Infrastructure for Developing Information Access Methodology. *Journal of Information Processing*, **20**(1), 216–227.

Smith, C. S. 1999. Activities: States or Events? *Linguistics and Philosophy*, **22**(5), 479–508.

Šnajder, J., Bašić, B. D. i Tadić, M. 2008. Automatic Acquisition of Inflectional Lexica for Morphological Normalisation. *Information Processing & Management*, **44**(5), 1720–1731.

Štajner, S., Evans, R., Orasan, C. i Mitkov, R. 2012. What Can Readability Measures Really Tell Us About Text Complexity? *U: Proceedings of the LREC'12 Workshop: Natural Language Processing for Improving Textual Accessibility (NLP4ITA).*

Steinberger, R., Pouliquen, B. i Van Der Goot, E. 2009. An Introduction to the European Media Monitor Family of Applications. *Str. 1–8 u: Proceedings of the Information Access in a Multilingual World-Proceedings of the SIGIR 2009 Workshop*.

Stevenson, M. i Wilks, Y. 2003. Word Sense Disambiguation. *The Oxford Handbook of Comp. Linguistics*, 249–265.

Sun, W., Rumshisky, A. i Uzuner, O. 2013. Evaluating Temporal Relations in Clinical text: 2012 i2b2 Challenge. *Journal of the American Medical Informatics Association*.

Surdeanu, M., Harabagiu, S., Williams, J. i Aarseth, P. 2003. Using Predicate-Argument Structures for Information Extraction. *Str. 8–15 u: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics.

Surdeanu, M. i Turmo, J. 2005. Semantic Role Labeling Using Complete Syntactic Analysis. *Str. 221–224 u: Proceedings of the Ninth Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L. i Nivre, J. 2008. The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. *Str. 159–177 u: Proceedings of the Twelfth Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

Taylor, R. S. 1962. The Process of Asking Questions. *American documentation*, **13**(4), 391–396.

Teh, Y. W., Jordan, M. I., Beal, M. J. i Blei, D. M. 2006. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, **101**(476).

Tong, H., Faloutsos, C., Gallagher, B. i Eliassi-Rad, T. 2007. Fast Best-Effort Pattern Matching in Large Attributed Graphs. *Str. 737–746 u: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Toutanova, K. i Manning, C. D. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. *Str. 63–70 u: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*. Association for Computational Linguistics.

UzZaman, N. i Allen, J. F. 2011. Temporal Evaluation. *Str. 351–356 u: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*.

UzZaman, N. i Allen, J. 2010. TRIPS and TRIOS system for TempEval-2: Extracting Temporal Information from Text. *Str. 276–283 u: Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

UzZaman, N., Llorens, H., Derczynski, L., Verhagen, M., Allen, J. i Pustejovsky, J. 2013. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. *U: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantcis (* SEM 2013). Association for Computational Linguistics, June*.

Van Benthem, J. F. 1983. The Logic of Time. A Model-Theoretic Investigation into the Varieties of Temporal Antology and Temporal Discourse.

van Rijsbergen, C. 1979. *Information Retrieval.* Butterworths, London.

Vendler, Z. 1957. Verbs and Times. *The Philosophical Review*, 143–160.

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G. i Pustejovsky, J. 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification. *Str. 75–80 u: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007).*

Verhagen, M., Sauri, R., Caselli, T. i Pustejovsky, J. 2010. SemEval-2010 Task 13: TempEval-2. *Str. 57–62 u: Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*. Association for Computational Linguistics.

Voorhees, E. M. 2002. The Philosophy of Information Retrieval Evaluation. *Str. 355–370 u: Evaluation of cross-language information retrieval systems*. Springer.

Walls, F., Jin, H., Sista, S. i Schwartz, R. 1999. Topic Detection in Broadcast News. *Str. 193–198 u: Proceedings of the DARPA Broadcast News Workshop*.

Wayne, C. 2000. Multilingual Topic Detection and Tracking: Successful Research Enabled by Corpora and Evaluation. *Str. 1487–1494 u: Proceedings of the Second International Conference on Language Resources and Evaluation Conference (LREC 2000)*, vol. 2000.

Wiebe, J., Bruce, R., Bell, M., Martin, M. i Wilson, T. 2001. A Corpus Study of Evaluative and Speculative Language. *Str. 1–10 u: Proceedings of the Second SIGdial Workshop on Discourse and Dialogue-Volume 16*. Association for Computational Linguistics.

Wiener, H. 1947. Structural Determination of Paraffin Boiling Points. *Journal of the American Chemical Society*, **69**(1), 17–20.

Wolf, F. i Gibson, E. 2005. Representing Discourse Coherence: A Corpus-Based Study. *Computational Linguistics*, **31**(2), 249–287.

Woodsend, K. i Lapata, M. 2011. Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming. *U: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*.

Wu, Z. i Palmer, M. 1994. Verbs Semantics and Lexical Selection. *Str. 133–138 u: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL 1994)*. Association for Computational Linguistics.

Wubben, S., van den Bosch, A. i Krahmer, E. 2012. Sentence Simplification by Monolingual Machine Translation. *Str. 1015–1024 u: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. ACL '12. Stroudsburg, PA, USA: Association for Computational Linguistics.

Yan, R., Kong, L., Huang, C., Wan, X., Li, X. i Zhang, Y. 2011. Timeline Generation through Evolutionary Trans-Temporal Summarization. *Str. 433–443 u: Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yang, Y., Carbonell, J. G., Brown, R. D., Pierce, T., Archibald, B. T. i Liu, X. 1999. Learning Approaches for Detecting and Tracking News Events. *Intelligent Systems and their Applications*, **14**(4), 32–43.

Yeh, A. 2000. More Accurate Tests for the Statistical Significance of Result Differences. *Str. 947–953 u: Proceedings of the 18th Conference on Computational linguistics*. Association for Computational Linguistics.

Zobel, J. 1998. How Reliable Are the Results of Large-Scale Information Retrieval experiments? *Str. 307–314 u: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.

Životopis

Goran Glavaš rođen je 25. kolovoza 1986. godine u Brčkom u Bosni i Hercegovini. Preddiplomski studij računarstva završio je 2008. godine na Fakultetu eletrotehnike i računarstva Sveučilišta u Zagrebu. Na istome je fakultetu (smjer računarstvo) 2010. godine s temom "Informacijski sustav praćenja morskih sisavaca" diplomirao s najvećom pohvalom. Za izvrsnost tijekom preddiplomskog i diplomskog studija četiri je puta nagrađen priznanjem "Josip Lončar".

Od svibnja 2011. godine zaposlen je na Zavodu za elektroniku, mikroelektroniku, inteligentne i računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva kao stručni suradnik, a od rujna 2012. kao znanstveni novak na projektu "Otkrivanje znanja u tekstnim podacima". Bio je uključen u nastavne aktivnosti Zavoda na predmetima Umjetna inteligencija, Skriptni jezici, Neizrazito evolucijsko i neuroračunarstvo i Strojno učenje, a asistirao je u vođenju osam diplomskih i završnih radova.

Njegovi istraživački interesi obuhvaćaju područje obrade prirodnoga jezika, pretraživanja informacija i dubinske analize teksta. U jesen 2013. bio je gostujući istraživač na Sveučilištu KU Leuven. U suautorstvu je objavio 15 radova na međunarodnim znanstvenim skupovima i četiri rada u časopisima s međunarodnom recenzijom, od kojih dva u časopisima indeksiranima u bazi CC. Član je strukovne udruge ACL (Association for Computational Linguistics). Govori engleski i talijanski jezik.

Popis objavljenih djela

Radovi u časopisima

1. Glavaš, G., Šnajder, J., "Construction and Evaluation of Event Graphs", Natural Language Engineering, Vol. 20, 2014., u tisku.
2. Glavaš, G., Šnajder, J., "Event Graphs for Information Retrieval and Multi-Document Summarization", Expert Systems with Applications, Vol. 41, 2014., u tisku.
3. Karan, M., Glavaš, G., Šarić, F., Šnajder, J., Dalbelo Bašić, B., "CroNER: Recognizing Named Entities in Croatian Using Conditional Random Fields", Informatics, Vol. 37, prosinac 2013., str. 165–172.

4. Glavaš, G., Fertalj, K., "Solving the Class Responsibility Assignment Problem Using Metaheuristic Approach", Journal of Computing and Information Technology, Vol. 19 (Number 4), prosinac 2011., str. 275–283.

Radovi na međunarodnim znanstvenim skupovima

1. Glavaš, G., Šnajder, J., Kordjamshidi, P., Moens, M.F., "HiEvents: A Corpus for Extracting Event Hierarchies from News Stories", 9th International Conference on Language Resources and Evaluation (LREC 2014), svibanj 2014., prihvaćen za objavljivanje.
2. Glavaš, G., Šnajder, J., "Event-Centered Information Retrieval Using Kernels on Event Graphs", 8th Workshop on Graph-Based Methods for Natural Language Processing in conjunction with Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), listopad 2014., str. 1–5.
3. Glavaš, G., Štajner, S., "Event-Centered Simplification of News Stories", Student Research Workshop at Conference on Recent Advances in Natural Language Processing (RANLP 2013), rujan 2013., str. 71–78, nagrada za najbolji rad.
4. Glavaš, G., Šnajder, J., "Recognizing Identical Events with Graph Kernels", 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013), kolovoz 2013., str. 797–803.
5. Glavaš, G., Korenčić, D., Šnajder, J., "Aspect-Oriented Opinion Mining from User Reviews in Croatian", 4th Workshop on Balto-Slavonic Natural Language Processing (BSNLP 2013) in conjunction with 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013), kolovoz 2013., str. 18–22.
6. Glavaš, G., Šnajder, J., "Exploring Coreference Uncertainty of Generically Extracted Event Mentions", Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2013), ožujak 2013., str. 408–422.
7. Glavaš, G., Šnajder, J., Dalbelo Bašić, B., "Are You for Real? Learning Event Factuality in Croatian Texts", Conference on Data Mining and Data Warehouses (SiKDD 2012), listopad 2012.
8. Glavaš, G., Karan, M., Šarić, F., Šnajder, J., Mijić, J., Šilić, A., Dalbelo Bašić, B., "CroNER: A State-of-the-Art Named Entity Recognition and Classification for Croatian", 8th Language Technologies Conference (IS-LTC 2012), listopad 2012., str. 73–78.
9. Marović, M., Šnajder, J., Glavaš, G., "Event and Temporal Relation Extraction from Croatian Newspaper Texts", 8th Language Technologies Conference (IS-LTC 2012), listopad 2012., str. 141–146.
10. Glavaš, G., Šnajder, J., Dalbelo Bašić, B., "Semi-Supervised Acquisition of Croatian Sentiment Lexicon", 15th International Conference on Text, Speech, and Dialogue (TSD 2012), rujan 2012., str. 166–173.

11. Glavaš, G., Fertalj, K., Šnajder, J., “From Requirements to Code: Syntax-Based Requirements Analysis for Data-Driven Application Development”, 17th International Conference on Applications of Natural Language Processing to Information Systems (NLDB 2012), lipanj 2012., str. 339–344.
12. Šarić, F., Glavaš, G., Karan, M., Šnajder, J., Dalbelo Bašić, B., “TakeLab Systems for Measuring Semantic Text Similarity”, The First Joint Conference on Lexical and Computational Semantics (*SEM 2012), lipanj 2012., str. 441–448.
13. Glavaš, G., Šnajder, J., Dalbelo Bašić, B., “Experiments on Hybrid Corpus-Based Sentiment Lexicon Acquisition”, Workshop on Innovative Hybrid Approaches to Processing Textual Data in conjunction with 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), travanj 2012., str. 1–9.
14. Glavaš, G., Fertalj, K., “Metaheuristic Approach to Class Responsibility Assignment Problem”, 33rd International Conference on Information Technology Interfaces (ITI 2011), lipanj 2011., str. 591–596.
15. Martinović, A., Glavaš, G., Juribašić, M., Sutić, D., Kalafatić, Z., “Real-Time Detection and Recognition of Traffic Signs”, 33rd International Convention MIPRO (2010), svibanj 2010., str. 247–252.

Biography

Goran Glavaš was born on August 25, 1986 in Brčko, Bosnia and Herzegovina. He received his B.Sc. in Computing from the University of Zagreb, Faculty of Electrical Engineering and Computing in 2008 and M.Sc. in Computer Science from the same university in 2010 (graduated with highest honours; thesis title: "Information Systems for Monitoring of Marine Mammals"). During his undergraduate and graduate studies he received four "Josip Lončar" Awards for excellence from the Faculty.

From May 2011 he was employed as a project associate at the Department of Electronics, Microelectronics, Computer and Intelligent Systems at the Faculty of Electrical Engineering and Computing. From September 2012 he is employed as a research associate at the same department on the project "Knowledge Discovery from Textual Data". He was involved in Department's educational activities within the courses on Artificial Intelligence; Scripting languages; Fuzzy, Evolutionary, and Neurocomputing; and Machine Learning. He also assisted in supervising four bachelor's theses.

His research interest include natural language processing, information retrieval, and text mining. In the Fall of 2013 he was a visiting scholar at KU Leuven. He has co-authored 15 conference papers and four journal papers, two of which in journals indexed in Current Contents. He is a member of the ACL (Association for Computational Linguistics). He is fluent in English and Italian.