Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Alen Nemec

Napovedovanje konceptov na podlagi toka dogodkov

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: doc. dr. Tomaž Curk

Ljubljana, 2018

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Opisi dogodkov vključujejo specifične osebe, organizacije, naravne pojave, kraje ter pojme, ki povezujejo le-te. Časovni tok povezanih dogodkov je tako tok konceptov, ki se pojavljajo v posameznih opisih. Preučite možnosti uporabe urejene zbirke novic o svetovnih dogodkih *Event Registry* za napovedovanja toka konceptov in posredno toka dogodkov. Zgrajeni napovedni model naj upošteva časovno dimenzijo podatkov o dogodkih. Poročajte o uspešnosti razvitega pristopa.

Zahvaljujem se mentorju, doc. dr. Tomažu Curku za usmerjanje, staršem in prijateljem za podporo pri pisanju diplomske naloge ter avtorjem sistema Event Registry za dostop do podatkov, brez katerega ta diplomska naloga ne bi bila izvedljiva.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
SDK	software development kit	programsko razvojno orodje
API	application programming interface	aplikacijski programski vmesnik
BP-MLL	backpropagation for multi label learning	vzvratno razširjanje napake za večznačno učenje

Povzetek

Naslov: Napovedovanje konceptov na podlagi toka dogodkov

Avtor: Alen Nemec

Možnost napovedovanja prihodnjih dogodkov in njihovih posledic je privlačna ideja, a v praksi težko izvedljiva zaradi velikega števila možnih izidov. Ta diploma predstavlja poskus napovedovanja, ki temelji na predpostavki, da se vzorci iz preteklosti ponavljajo. Svetovne dogodke modeliramo kot skupine konceptov, na podlagi katerih jih gručimo v skupine povezanih dogodkov. Iz teh zgradimo podatkovno zbirko za namene napovedovanja, kjer vhodni atributi opisujejo koncepte, ki se pojavijo v posamezni gruči znotraj danega časovnega okna, ciljne oznake pa so koncepti, ki se pojavijo naslednji teden. Na podatkih ocenimo in primerjamo uspešnost različnih modelov za napovedovanje. Naši poskusi pokažejo, da takšno modeliranje povezav med koncepti prispeva koristne informacije za namene napovedovanja prihodnjih dogodkov.

Ključne besede: novice, gručenje, podatkovno rudarjenje, napovedovanje, dogodki.

Abstract

Title: Predicting concepts based on streams of events

Author: Alen Nemec

Having the ability to predict the course of future events is an attractive idea, but difficult to achieve in practice because of the vast number of possibilities. This diploma presents an attempt at doing so based on the hypothesis that patterns of events from the past tend to repeat. We model real world events as groups of concepts. We cluster events into groups of related events. We then build a dataset from these clusters, where the attributes of each data point represent concepts that occur in a single cluster within a certain time window, and the target labels are concepts that occur the next week. We compare the effectiveness of several different prediction models. Our tests show that relationships between concepts contain useful information for predicting the future course of events.

Keywords: news, clustering, data mining, prediction, events.

Poglavje 1

Uvod

Napovedovanje prihodnjih dogodkov je relativno neraziskano področje. Preden zgradimo napovedni model, moramo odgovoriti na dve vprašanji.

Kaj napovedujemo? Potrebno je opredeliti spremenljivke, ki jih napovedujemo, saj morajo veljati tako za pretekle kot prihodnje dogodke. To so lahko splošni dogodki oziroma karakteristike dogodkov, ki ne vsebujejo informacij o kontekstu, kot so čas, lokacija ali vpletene entitete. Primeri takšnih vrednosti so "vojna," "naravna nesreča" ali "evakuacija."

Kako definirati relacije med dogodki? Za učenje modela potrebujemo predznanje o obstoječih relacijah med dogodki, ki ga lahko pridobimo iz analize besedil, ki dogodke opisujejo in konteksta, v katerem se zgodijo. Tako lahko na primer predpostavimo, da sta dva dogodka povezana, če se zgodita znotraj nekega časovnega intervala in imata podobne karakteristike, kot sta lokacija ali vpletene entitete.

Klasični pristop, ki sicer ne uporablja metod umetne inteligence ali podatkovnega rudarjenja, se zanaša na anketiranje števila posameznikov, ki so tipično strokovnjaki na svojem področju, in iskanje konsenza med njimi. Ta metoda je znana kot metoda *Delphi*. Razvita je bila v obdobju 1950-1960 [6] in se v takšni ali drugačni obliki uporablja tudi danes.

Na področju umetne inteligence je omembe vreden model avtorja Sipple [8], ki napoveduje prihajajoče dogodke glede na relacije med dogodki iz

preteklosti, pridobljene na podlagi informacij o kontekstu. Predlagan sistem deluje v realnem času in na veliki količini podatkov. Analizira prihajajoče podatke in v njih zazna pomembna dogajanja, na podlagi katerih napoveduje prihajajoče dogodke. Za modeliranje relacij med njimi uporablja statistične metode in gručenje na podlagi informacij o tipu dogodka, času in lokaciji. Vir podatkov prestavljajo spletni viri, ki govorijo o dogodkih, kot so novinarski članki, socialna omrežja in podobno.

Poglavje 2

Metode

2.1 Gručenje

S postopkom gručenja podatkov določamo obstoječo strukturo v zbirki podatkov. Gre za proces razvrščanja primerov v skupine na osnovi neke izbrane funkcije podobnosti med primeri.

V praksi je iskana struktura lahko kakršnekoli oblike, zato obstaja tudi veliko različnih pristopov. Izbira pristopa gručenja kot ocenjevanja dobljenih rezultatov potemtakem temelji na naših predpostavkah o obliki iskanih gruč v podatkih. Gručenje je oblika nenadzorovanega učenja, v primerjavi z nadzorovanim učenjem. To pomeni, da v procesu učenja število gruč ni znano, niti niso znane pravilne klasifikacije posameznih primerov v gruče. V diplomskem delu smo uporabili pristop hierarhičnega gručenja.

2.1.1 Hierarhično gručenje

Hierarhično gručenje je pristop, kjer vsak posamezen primer najprej predstavlja lastno gručo. V vsakem koraku nato postopoma združuje najbolj podobne gruče glede na izbrano mero podobnosti, dokler ustavitveni pogoj ni izpolnjen oziroma dokler ne ostane samo ena gruča. Pristop zgradi hierarhično drevo gruč. Tipično uporabljene mere razdalje med primeri za hierarhično gručenje so evklidska razdalja, manhatanska razdalja in kosinusna

razdalja, vendar pa pristop deluje s katerokoli mero razdalje. Podobnost med gručami primerov določamo na podlagi razdalje med primeri, ki pripadajo različnim gručam. Pri tem lahko uporabimo več mer:

Posamezna povezanost (ang. *single linkage*) za razdaljo med dvema gručama vzame najkrajšo razdaljo med gručama.

Celotna povezanost (ang. *complete linkage*) vzame največjo razdaljo med gručama.

Povprečna povezanost (ang. *average linkage*) uporabi povprečno razdaljo med gručama.

Povezanost *Ward* poimenovana po raziskovalcu, ki jo je predlagal, pri združevanju uporablja kriterij, ki minimizira varianco evklidske razdalje znotraj gruč [9]. Princip, na katerem temelji metoda Ward, je optimizacija kriterijske funkcije in se v teoriji lahko prilagodi tako, da upošteva poljubne kriterije.
Na podlagi razdalje med združenimi gručami v vsakem koraku lahko zgradimo vizualizacijo postopka, tako imenovan dendrogram. Takšna vizualizacija nosi koristne informacije o strukturi gruč tudi pri visokodimenzionalnih podatkih.

Evklidska razdalja

Evklidska razdalja (ang. *euclidean distance*) je razdalja med dvema točkama v evklidskem prostoru. Če sta točki q in p v evklidskem prostoru z d dimenzijami, potem je njuna razdalja definirana kot:

$$\sqrt{\sum_{i=1}^{d}(q_i - p_i)^2}$$

2.1.2 Prekletstvo dimenzionalnosti

Pojem prekletstvo dimenzionalnosti (ang. *curse of dimensionality*) se pogosto pojavlja pri problemih z visoko dimenzionalnostjo podatkov. Tipični primeri v resničnem svetu so genetski podatki in besedila.

Pojem opisuje vrsto težav, ki se pojavijo pri analizi podatkovnih zbirk z visoko dimenzionalnostjo. S povečevanjem števila dimenzij se eksponentno povečuje prostornina prostora. Posledično se povečuje tudi količina podatkov, ki so potrebni za dober opis prostora.

Določene mere razdalje lahko pri visoki dimenzionalnost izgubijo učinkovitost, saj se kontrast med razdaljami do najbližjega in najbolj oddaljenega soseda neke točke zmanjšuje. S tem se zmanjšuje tudi sposobnost diskriminacije med gručami točk [1]. Problem je bolj očiten pri zelo redkih matrikah.

Najpogostejši način reševanja problema prekletstva dimenzionalnosti je zmanjševanje dimenzionalnosti, kjer uporabimo eno od vrste metod za projekcijo podatkov v manjše število dimenzij. To je velikokrat zadostna rešitev, saj imamo pri visoki dimenzionalnosti tipično veliko atributov, ki so med seboj povezani, ali pa preprosto ne prispevajo dovolj informacije, da bi jih lahko uporabili. Takrat lahko učinkovito zmanjšamo število dimenzij in obdržimo večino informacije. Poleg tega se lahko krepko zmanjšata tudi časovna in prostorska zahtevnost algoritmov, uporabljenih za analizo teh podatkov.

2.1.3 Analiza glavnih komponent (PCA)

Analiza glavnih komponent (ang. *principal component analysis*) je metoda za zmanjševanje dimenzionalnosti. S pomočjo ortogonalne transformacije preslika podatke v manjše število dimenzij. Hkrati minimizira izgubo informacij in obdrži čim večjo varianco preslikanih točk.

2.1.4 Mere kvalitete gručenja

Zaradi široke definicije problema gručenja ima pričakovana struktura gruč v naših podatkih, kot tudi struktura podatkov samih, velik vpliv na to, katero

mero kvalitete gručenja izberemo.

V splošnem jih delimo na notranje in zunanje mere gručenja. Notranje mere ocenijo kvaliteto gručenja brez zanašanja na podatke o dejanskih oznakah, tako da preverijo strukturo gruč samih. Zunanje mere pa ocenijo gručenje glede na podatke o dejanskih oznakah. Zunanje mere so načeloma bolj točne, ampak manj uporabljene, saj se pri gručenju pogosto soočamo s primeri, za katere ne poznamo dejanskih oznak.

Silhuetna ocena je notranja mera gručenja, ki temelji na analizi kompaktnosti in ločenosti gruč [7]. Za dano množico primerov c, ki pripadajo gruči velikosti K, se *silhouette(c)* izračuna kot:

$$silhouette(c) = \frac{1}{K} \sum_{x_i \in c} \frac{b(x_i) - a(x_i)}{max\{a(x_i), b(x_i)\}},$$

kjer $a(x)$ označuje povprečno razdaljo primera x od drugih primerov znotraj iste gruče, $b(x)$ pa povprečno razdaljo primera x od vseh primerov najbližje sosednje gruče. Oceno gručenja potemtakem dobimo kot povprečje silhuetne ocene množice gruč.

Rezultat je omejen na intervalu $[-1, 1]$. Vrednost 1 pomeni, da so gruče dobro ločene, vrednost -1 pa, da so gruče razpršene in slabo definirane. Primer z visoko oceno se nahaja blizu središča gruče, primer z nizko oceno pa na robu gruče, kateri pripada. Negativne ocene označujejo primere, ki so bili dodeljeni napačni gruči, saj ležijo bližje sosednji gruči kot svoji lastni.

Povprečna kronološka povezanost Zaradi potrebe po meri, ki bi upoštevala naše predpostavke o strukturi gruč v podatkih, smo vpeljali novo mero kvalitete gručenja. Ta za vsako gručo izračuna povprečje razdalj med posameznimi dogodki v gruči, ki si sledijo po datumu, in ga deli s povprečjem vseh razdalj znotraj gruče. Če je množica dogodkov D v gruči, ki so urejeni po datumu in K število teh dogodkov, potem je:

$$chron(D) = \frac{1}{K} \sum_{d_i \in D} dist(d_i, d_{i+1}),$$

kjer $dist(d_i, d_{i+1})$ predstavlja funkcijo, ki izračuna razdaljo med dogodkom (d_i) in dogodkom, ki mu sledi (d_{i+1}). Vrednost nato dodatno delimo s povprečjem vseh razdalj v gruči.

Mera temelji na predpostavki, da imajo želene gruče splošno obliko verige, kjer so najbolj tesno povezani tisti dogodki, ki si sledijo kronološko.

2.2 Večznačna in večrazredna klasifikacija

Pri podatkovnem rudarjenju se včasih srečujemo s primeri, ki imajo več ciljnih oznak. Takrat govorimo o večrazredni klasifikaciji (ang. *multi-class*), če za vsak primer napovedujemo eno od oznak, oziroma večznačni klasifikaciji (ang. *multi-label*), kadar napovedujemo več oznak hkrati. Tipičen primer večznačnega problema je razvrščanje besedil po kategorijah.

V diplomskem delu se ukvarjamo z večznačno klasifikacijo, saj za vsak primer napovedujemo n konceptov.

2.2.1 Eden-proti-vsem

Metoda eden-proti-vsem (ang. *one-vs-rest*) je preprost pristop k večznačnim in večrazrednim primerom, kjer naučimo ločen klasifikator za vsako ciljno oznako. Zgradimo n napovednih modelov za n oznak. Tako večznačni problem prevedemo na binarno klasifikacijo, kjer nam vsak klasifikator pove, ali je pripadajoča oznaka prisotna ali ne. Pri večznačnih problemih je metoda znana kot binarna relevantnost (ang. *binary relevance*). Klasifikator, uporabljen pri tej metodi, je lahko poljuben.

Pristop je hiter in relativno učinkovit. Poleg tega lahko analiziramo klasifikator za posamezno oznako in izvemo koristne informacije o njej. Njegova slabost je ta, da ne upošteva odvisnosti med ciljnimi oznakami. Zaradi preprostosti služi kot dobro izhodišče za primerjavo z drugimi modeli.

2.2.2 Metoda linearnih podpornih vektorjev

Metoda linearnih podpornih vektorjev (ang. *linear support vector classifier*) je binarni klasifikator, ki za podatke s poljubnim številom dimenzij določi hiperravnino, ki jih razdeli na dva razreda na način, da je razdalja med ravnino in najbližjima primeroma vsakega razreda čim večja.

2.2.3 Metoda naključnih gozdov

Metoda naključnih gozdov (ang. *random forest*) je pristop, ki za napovedovanje zgradi vrsto odločitvenih dreves (ang. *decision tree*) in napove najpogostejšo vrednost. Algoritem tipično deluje po principu "bootstrap," tako da za vsako drevo iz originalne učne množice naključno (z vračanjem) izbere n vzorcev nad katerimi zgradi odločitveno drevo.

Odločitvena drevesa so zgrajena tako, da ob vsakem koraku izberemo atribut, ki množico podatkov razdeli najbolj homogeno. Pri metodi naključnih gozdov izbiramo med naključno podmnožico vseh atributov, zato da se izognemo prevelikemu prilagajanju modela (ang. *overfitting*).

Metodo naključnih gozdov je mogoče prilagoditi večznačnim problemom tako, da odločitvena drevesa vračajo n vrednosti namesto ene same, pri ločevanju množice pa upoštevamo povprečno zmanjševanje variance.

2.2.4 Nevronske mreže

Nevronske mreže so metode, ki imajo temelje v delovanju nevronov v človeških možganih. Na področju strojnega učenja se prvič pojavijo nekje v drugi polovici 20. stoletja in pozneje zamrejo zaradi nezadostne procesorske moči takratnih računalnikov in splošno slabe uspešnosti.

Ponoven razcvet doživijo na koncu 20. stoletja zaradi hitrega razvoja računalniške opreme in razvojem boljših algoritmov ter metod, kot je vzvratno razširjanje napake (ang. *backpropagation*) [10]. Danes veljajo za močno orodje umetne inteligence, ki dosega dobre rezultate na težkih problemih, kot je, na primer, klasifikacija slik.

BP-MLL

Metoda *Back Propagation for Multi Label Learning* (BP-MLL) je mera napake, prilagojena za večznačne primere (ang. *loss function*), ki upošteva korelacije med posameznimi atributi [11]. Napaka posameznega primera je definirana kot:

$$E_i = \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(k,l)\in Y_i \times \bar{Y}_i} exp(-(c_k^i - c_l^i)),$$

kjer je Y_i množica napovedanih oznak za primer i, $\bar{Y}_i$ je njena komplementarna množica, $|\cdot|$ pa označuje kardinalnost množice.

2.3 Mere uspešnosti napovedovanja

Pri evalvaciji napovednega modela obstaja vrsta mer, s katerimi lahko ocenimo pričakovano uspešnost modela na novih podatkih. Tipično za evalvacijo uporabimo več različnih mer, ki njegovo delovanje ocenijo z več zornih kotov. Pri večznačnih problemih sta informativni predvsem meri preciznosti in priklica, ki nam povesta, kolikšen odstotek napovedanih oznak je pravilen, oziroma kolikšen odstotek pravilnih oznak je bil napovedan.

Jaccardov indeks Za posamezen primer x je definiran kot razmerje med presekom in unijo napovedanih in dejanskih oznak. Za večznačne primere je mera znana kot točnost (ang. *accuracy*).

$$J(Y_{true}, Y_{pred}) = \frac{|Y_{true} \cap Y_{pred}|}{Y_{true} \cup Y_{pred}}$$

Preciznost Za večznačne probleme je definirana kot razmerje med številom pravilno napovedanih oznak in številom vseh napovedanih oznak, za vsak primer posebej. Povprečje skozi vse primere nam da oceno modela.

$$precision = \frac{|Y_{true} \cap Y_{pred}|}{Y_{pred}}$$

Visoka preciznost pomeni, da je model napovedal več pravilnih kot nepravilnih oznak.

Priklic (ang. *recall*) je definiran kot razmerje med številom pravilno napovedanih oznak in številom vseh dejanskih oznak.

$$recall = \frac{|Y_{true} \cap Y_{pred}|}{Y_{true}}$$

Visok priklic označuje model, ki napove večino pravilnih oznak.

Mera F (ang. *F-score*) je definirana kot harmonična sredina med preciznostjo in priklicem. Omejena je na intervalu $[0, 1]$.

$$F = 2 * \frac{precision \times recall}{precision + recall}$$

Poglavje 3

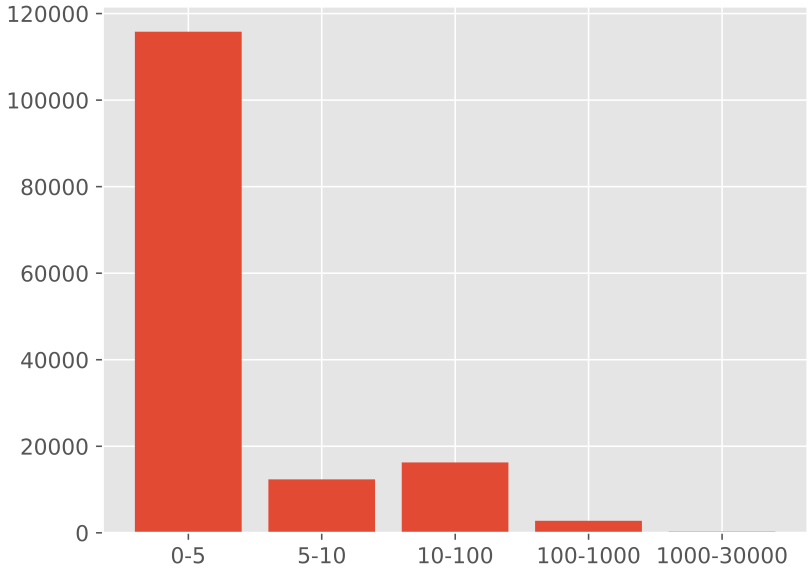
Podatki in orodja

Podatki izhajajo iz spletnih člankov, ki poročajo o aktualnih dogodkih, ki se dogajajo po svetu. Za pridobivanje podatkov smo uporabili sistem *Event Registry*, ki članke združuje in procesira s pomočjo metod tekstovnega rudarjenja.

3.1 Struktura podatkov

Podatke sprva pridobimo preko vmesnika API *Event Registry* za programski jezik Python. Od vmesnika zahtevamo vse dogodke med 1. 1. 2016 in 31. 12. 2016, ki so v angleškem jeziku in obsegajo vsaj 10 člankov. Teh je v praksi preveč za analizo in procesiranje, zato se omejimo na dogodke iz kategorije družbenih problemov (ang. *social issues*). To kategorijo smo izbrali kot primerno za analizo predvsem zato, ker zajema aktualna dogajanja po svetu, katerih posledice se lahko raztezajo dalj časa.

Dobimo 59775 različnih dogodkov, ki jih shranimo v datoteko csv, kjer vsaka vrstica predstavlja posamezen dogodek. Za vsak dogodek poznamo datum in kraj dogodka, vpletene koncepte ter oceno od 0 do 100, kako močno so povezani z dogodkom, število člankov, ki govorijo o dogodku, oceno popularnosti na družbenih omrežjih in splošne kategorije, pod katere spada. Vsak posamezen koncept spada v eno izmed štirih kategorij: organizacija, lokacija,



Slika 3.1: Distribucija konceptov glede na število pojavov v dogodkih. Os x predstavlja število pojavov, os y pa število konceptov.

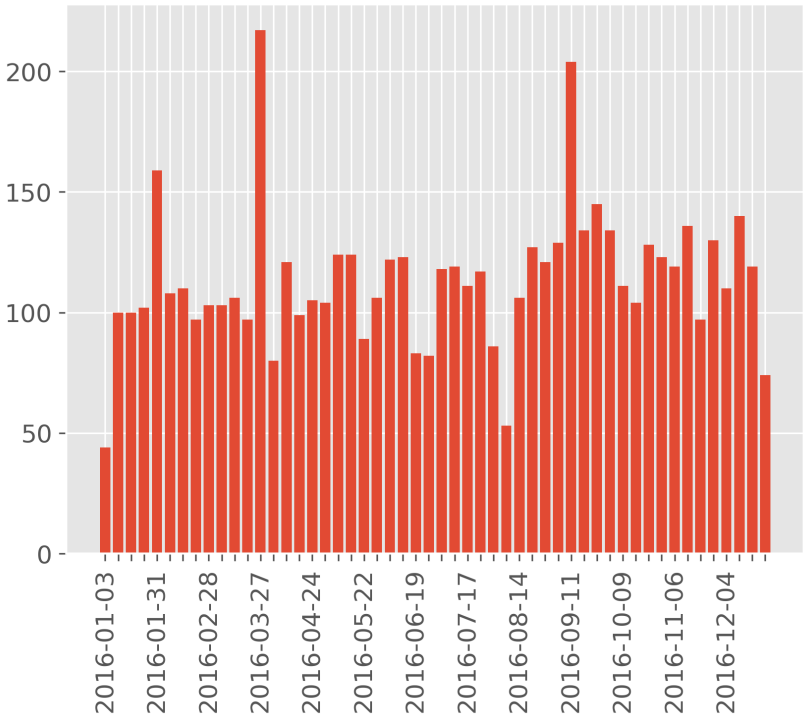
oseba ali pojem.

Vhod v model predstavlja redka matrika velikosti $n \times m$, kjer je n število dogodkov in m število konceptov. Posamezna celica matrike ima lahko vrednost od 0 do 100, ki predstavlja relevantnost koncepta v dogodku.

Podatki so visokodimenzionalni, obsegajo 147437 različnih konceptov, vendar se jih velika večina v dogodkih pojavi manj kot 5-krat, kot je to razvidno na sliki 3.1.

Kjer je bilo to mogoče, smo matriko dogodkov in njihovih konceptov obravnavali kot stisnjeno redko matriko vrstic (ang. *compressed sparse row matrix*) [2], v kateri so podatki shranjeni kot tri eno-dimenzionalne tabele, ki posamezno vsebujejo ne-nične vrednosti celic, dolžine vrstic in indekse stolpcev. Takšen zapis v delovnem pomnilniku zavzame bistveno manj prostora kot celotna matrika in še vedno podpira aritmetične operacije. To nam je omogočilo, da smo lahko s celotno matriko delali v pomnilniku.

3.2 Osnovne statistike o podatkih

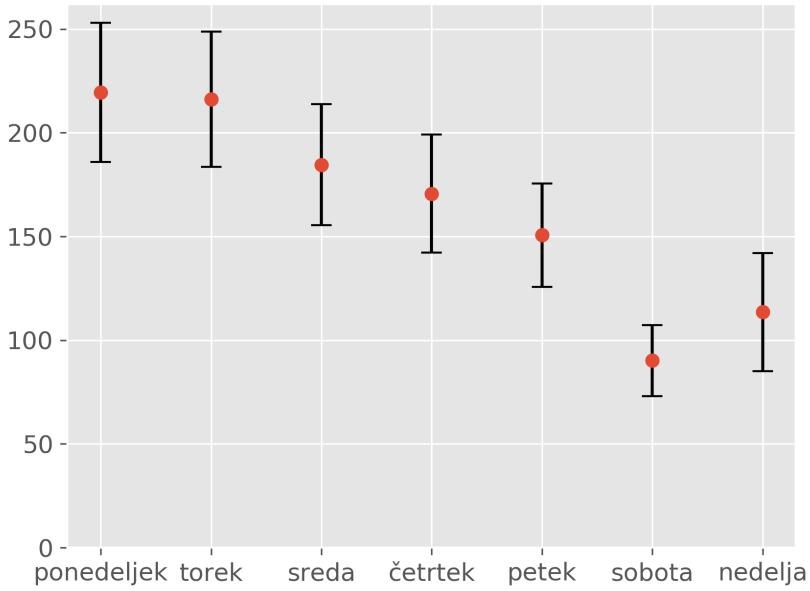


Slika 3.2: Število dogodkov na teden.

Število dogodkov ostaja relativno konstantno skozi celo leto, z nekaj izjemami, ki odražajo obdobja večje medijske aktivnosti zaradi odmevnih dogajanj. Dogodkov je načeloma manj med vikendi, kar je verjetno odraz medijskega cikla, ki je takrat manj aktiven, glej sliko 3.3.

Na sliki 3.4 je izrisana frekvenca konceptov na dan. Zaradi preglednosti je izrisanih samo najpogostejših 50 konceptov za zadnjih 100 dni v letu. Tudi tu je razvidno, da je dogodkov manj med vikendi kot pa med tednom. Omembe vredno je obdobje večje aktivnosti med datumi 11. 11. 2016 in 18. 11. 2016, kjer gre najverjetneje za poročanje o rezultatih takratnih ameriških predsedniških volitev, ki so bile 8. novembra.

Na sliki 3.5 je prikazana distribucija različnih konceptov glede na njihovo



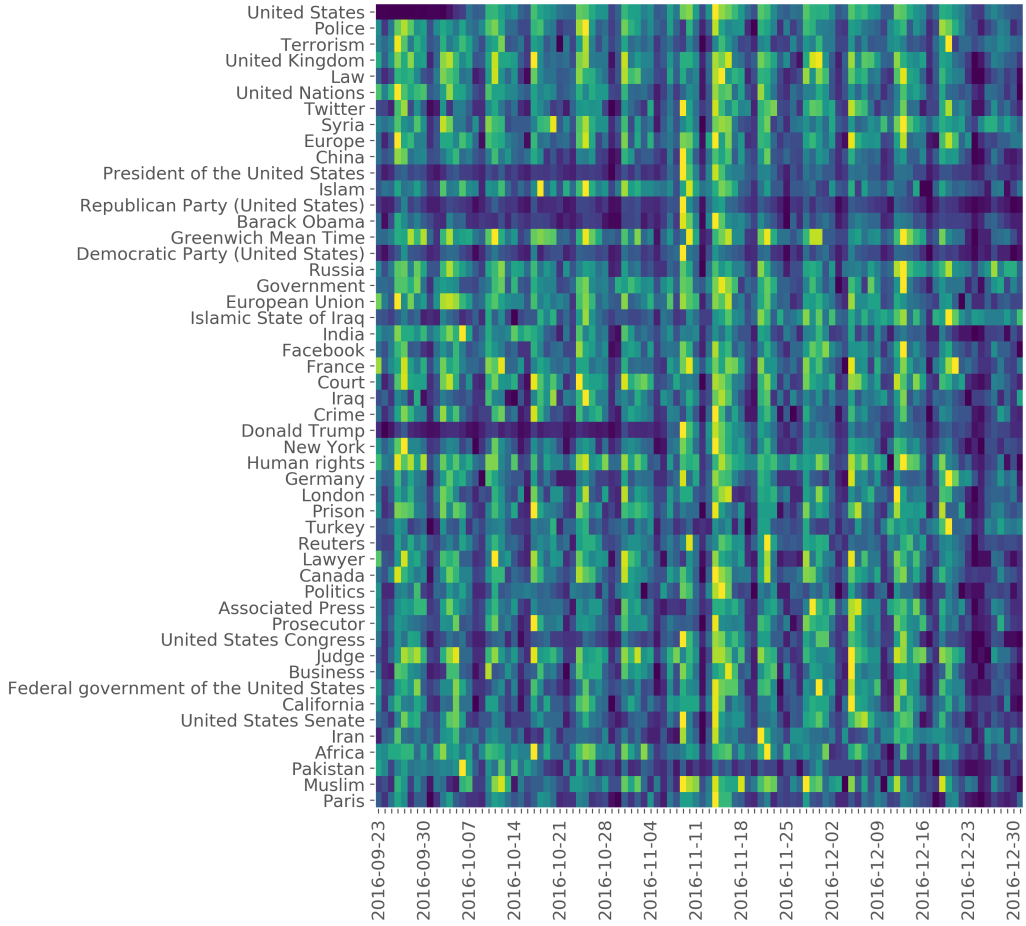
Slika 3.3: Povprečno število dogodkov za posamezni dan v tednu in standardni odklon.

klasifikacijo. Razvidno je, da je pojmov precej več kot pa oseb, lokacij ali organizacij.

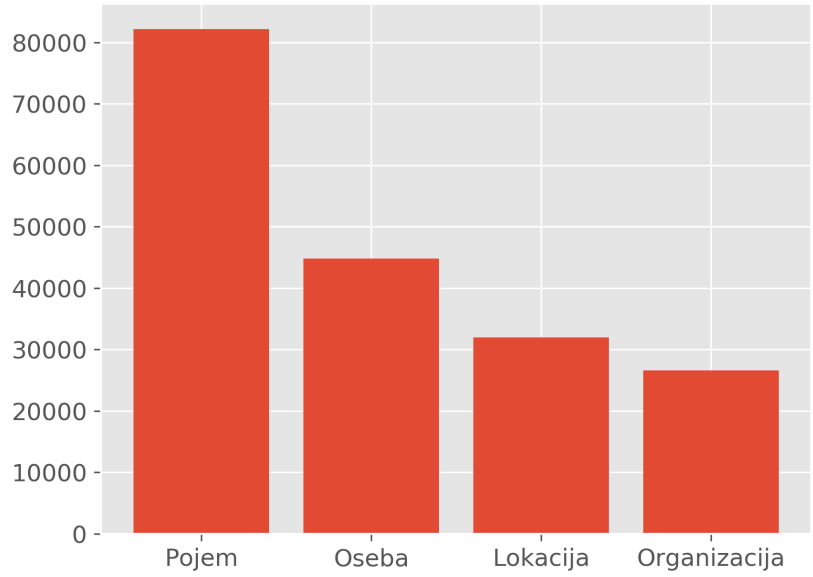
3.3 Event Registry

Event Registry je sistem, ki članke iz več kot 100000 virov v več kot 10 jezikih agregira in procesira s pomočjo tekstovnega rudarjenja [5, 4]. Sposoben je identificirati in združiti članke, ki govorijo o istem dogodku. Iz njih povzame ključne informacije o dogodkih, kot so lokacija, datum, vpletene entitete in koncepti. Omogoča analizo podatkov v podatkovni bazi preko uporabniškega vmesnika: API za pridobivanje podatkov in SDK za Python ter Node.js.

Deluje v štirih korakih. Prične s pridobivanjem člankov iz spletnih virov. Vsebino člankov nato pred-procesira z vrsto lingvističnih orodij. V njih detektira entitete in datume. V tretji fazi uporabi algoritem za gručenje, ločeno za vsakega izmed štirih najpogostejših jezikov. Vsakih n novih člankov gruče



Slika 3.4: Toplotna karta konceptov.



Slika 3.5: Število različnih konceptov po kategorijah.

ponovno analizira in jih združi oziroma razbije na več delov. Pri tem upošteva samo tiste, ki ne vsebujejo člankov starejših od k dni, zato ker te gruče najverjetneje ne opisujejo več istih dogodkov. Ime in opis dogodka izvirata iz imena in opisa članka, ki se nahaja v sami sredini gruče.

3.4 Python

Uporabili smo programski jezik Python, ki ponuja knjižnice za strojno učenje in manipulacijo podatkov: *numpy* in *pandas* za obdelavo podatkov, *sklearn* in *scipy* za splošne algoritme za strojno učenje in *keras* ter *tensorflow* za nevronske mreže.

Poglavje 4

Gradnja modela za napovedovanje konceptov

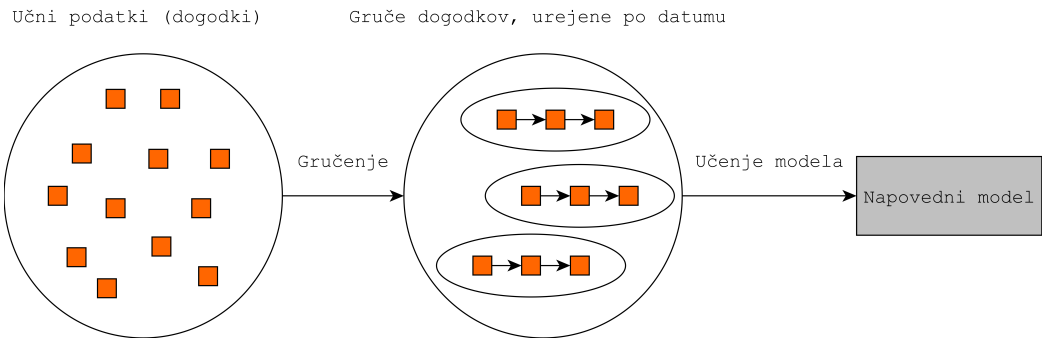
Model za napovedovanje konceptov v dogodkih zgradimo v več korakih:

1. Za vsak teden od datuma 28. 8. 2016 naprej podatkovno zbirko razdelimo na učne in validacijske podatke glede na datum. Učni podatki so dogodki, ki so se zgodili pred tem datumom, validacijski podatki pa dogodki, ki so se zgodili na ta dan ali po tem dnevu.

2. Učne dogodke gručimo z uporabo hierarhičnega gručenja.

3. Iz gruč dogodkov je sestavljena podatkovna zbirka, v kateri za vsak posamezen primer vhodni atributi predstavljajo vse koncepte znotraj gruče, ki so se zgodili znotraj časovnega okna, ciljni atributi pa so koncepti, ki so se zgodili v naslednjem tednu.

4. Na podlagi podatkovne zbirke v 3. točki zgradimo napovedni model.



Slika 4.1: Postopek gradnje napovednega modela.

4.1 Gručenje dogodkov

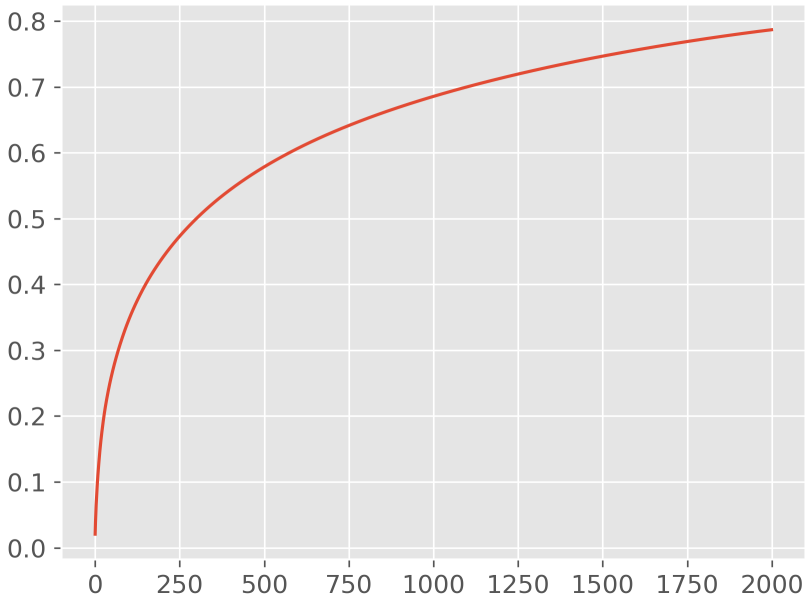
S pomočjo gručenja dogodkov dejansko želimo odkriti tok dogodkov, ki si smiselno sledijo in opisujejo neko dogajanje v svetu. Gruče nam torej predstavljajo verige dogodkov, ki implicirajo neko vzročno povezanost med dogodki.

Za gručenje smo uporabili hierarhično gručenje z metodo Ward in evklidsko razdaljo. Pri tem smo upoštevali koncepte, ki se v podatkih pojavijo v vsaj 5 dogodkih. Število dimenzij smo zmanjšali na 2000 z uporabo metode *randomized truncated SVD* [3], ki deluje na podoben način kot PCA s to razliko, da podatkov ne centrira pred izvajanjem. To pomeni, da deluje nad stisnjenimi redkimi matrikami in je časovno bolj učinkovita. Takšna projekcija obdrži 78,7 % variance v podatkih.

4.2 Napovedovanje

V koraku napovedovanja iz pridobljenih gruč najprej zgradimo podatkovno zbirko. Ta je sestavljena iz konceptov, ki so se zgodili znotraj časovnega okna, in konceptov, ki se zgodijo naslednji teden. Gre za binarne podatke, kjer 1 označuje prisotnost koncepta, 0 pa njegovo odsotnost.

Zgradimo en model na podlagi podatkov vseh gruč. Pri tem ne upošte-



Slika 4.2: Kumulativni graf razložene variance komponent SVD. Os x predstavlja komponente, os y pa odstotek pojasnjene variance.

vamo konceptov, ki so entitete (oseba, lokacija ali organizacija), ker so te specifične gručam in se ne generalizirajo na ostale podatke. Zahtevamo tudi, da se koncept pojavi v vsaj 100 dogodkih v učni množici. Tako zmanjšamo šum v podatkih in obdržimo samo tiste koncepte, ki se v podatkih pojavijo dovolj pogosto, da je iz njih mogoče razbrati vzorce.

Napovedovanje izvajamo na nivoju posameznih gruč, kar pomeni, da za vsako gručo zgradimo napoved na podlagi dogodkov znotraj časovnega okna v gruči. Nato za namene validacije napovedi dogodke iz validacijske množice umestimo v gruče iz učne množice z uporabo metode K-najbližjih sosedov (ang. *k-nearest neighbours*), ki jo naučimo na podlagi rezultatov gručenja učne množice. Pri ocenjevanju upoštevamo napovedi iz gruč, v katere spada vsaj en dogodek iz validacijske množice za trenutni teden.

Poglavje 5

Rezultati

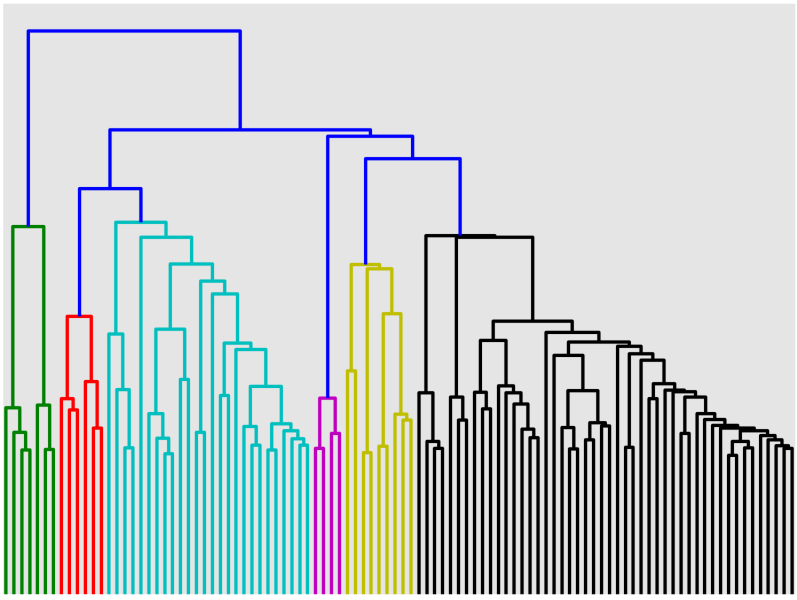
Predlagano metodo smo uporabili na podatkih iz sistema Event Registry, ki vsebujejo 59775 dogodkov in 147437 različnih konceptov. Podatke smo razdelili na učno in validacijsko množico glede na datum - dogodki, ki so se zgodili po 28. 8. 2016 so umeščeni v validacijsko množico.

5.1 Gručenje

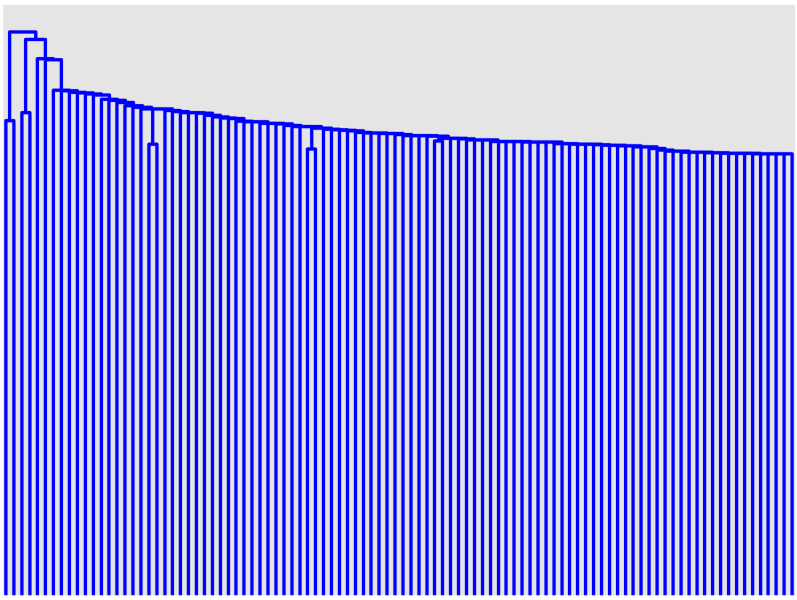
Za gručenje smo uporabili metodo Ward, ki daje najbolj enakomerne gruče, kot je prikazano na dendrogramu na sliki 5.1. Zaradi preglednosti je prikazanih samo zadnjih 100 združitev, ki jih je naredil algoritem. Ostali kriteriji za združevanje so privedli do slabo ločenih gruč. Primer je dendrogram gručenja na sliki 5.2, ki je bil zgrajen z uporabo povprečne povezanosti.

Na sliki 5.3 je razvidno, da je ena od gruč znatno večja od ostalih. Razlog za to je predvidoma visoka dimenzionalnost podatkov in posledično slabša sposobnost ločevanja med njimi.

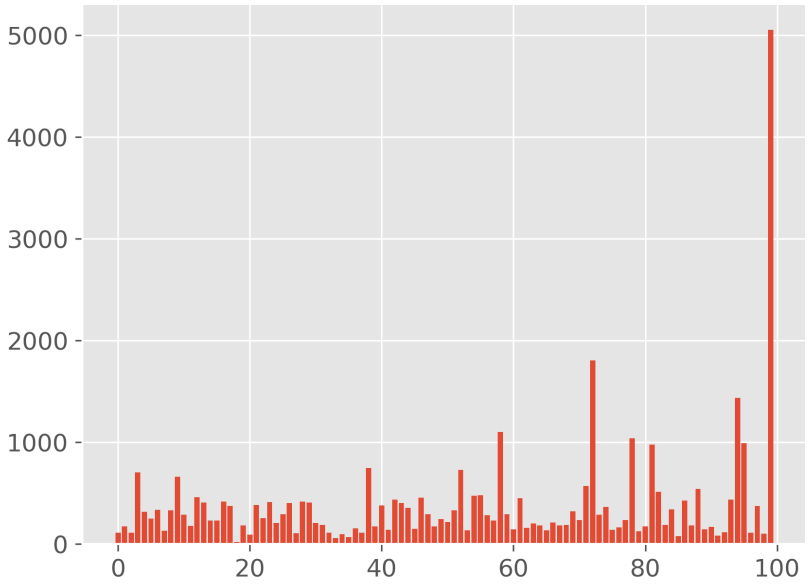
Na sliki 5.5 je razvidna slabost silhuetne ocene, saj v večini primerov boljše oceni gručenje z večjim številom gruč. S tega vidika se izkaže za bolj primerno povprečna kronološka mera, saj ne raste skupaj s številom gruč, kot je razvidno na sliki 5.4. Silhuetna ocena prav tako daje rezultate, ki so relativno blizu ničle, kar v splošnem sicer pomeni, da so gruče slabo ločene,



Slika 5.1: Dendrogram gručenja dogodkov pri uporabi metode Ward.



Slika 5.2: Dendrogram gručenja dogodkov pri uporabi povprečne povezanosti.



Slika 5.3: Velikosti gruč. Os x predstavlja gruče, os y pa število dogodkov.

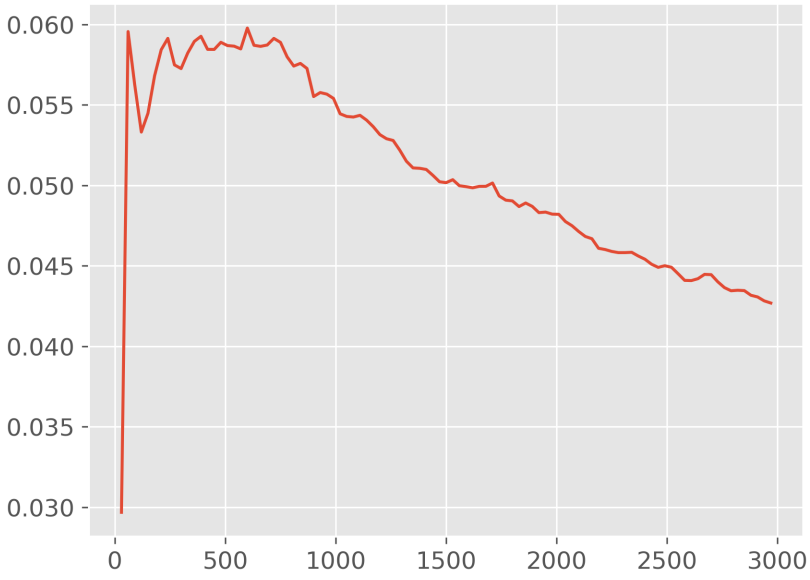
vendar to drži predvsem za gruče sferične oblike, pri katerih se ta mera najboljše odreže.

Če za določanje primernega števila gruč na podlagi grafov uporabimo metodo komolca (ang. *elbow method*), pri kateri primerno število gruč določimo glede na točko v grafu, kjer je sprememba v oceni največja, bi pri kronološki meri izbrali število med 50 in 600, pri silhuetni oceni pa število med 300 in 1700.

Pri gručenju dogodkov je vredno omeniti tudi vpliv, ki ga ima samo poročanje o dogodkih. Koncepti, s katerimi so povezani dogodki, so namreč določeni na podlagi njihove uporabe v člankih, kar posledično neposredno vpliva na razvrščanje dogodkov v gruče.

5.2 Napovedovanje

Za napovedovanje smo iz rezultatov gručenja zgradili dve podatkovni zbirki z različno dolžino časovnega okna t - 14 dni in 30 dni. Pričakovano je, da bo

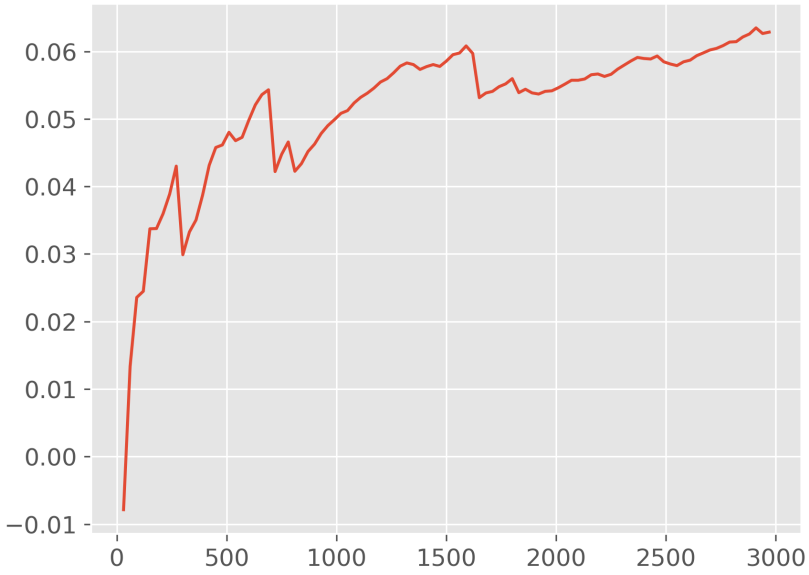


Slika 5.4: Gručenje, ocenjeno s povprečno kronološko povezanostjo. Os x prestavlja število gruč, os y pa oceno.

model z večjim t dosegel boljšo natančnost, ker ima na voljo več podatkov pri gradnji napovedi. Dogodke smo razdelili na 100 gruč. Nato smo primerjali rezultate treh konceptualno različnih metod na vsaki izmed podatkovnih zbirk. Primerjali smo metodo eden-proti-vsem z uporabo klasifikatorja linearnih podpornih vektorjev, nevronsko mrežo z vzvratnim razširjanjem napake BL-MLL, metodo naključnih gozdov in konstantni klasifikator, ki vedno napove najpogostejših 50 konceptov v učni množici.

5.3 Eden-proti-vsem z uporabo klasifikatorja linearnih podpornih vektorjev

Model eden-proti-vsem je pri obeh dolžinah časovnega okna boljši od konstantnega klasifikatorja pri vseh merah razen preciznosti. V tabeli 5.1 je razvidno, da ima večja velikost t tudi boljši priklic, medtem ko je preciznost



Slika 5.5: Gručenje, ocenjeno s silhuetno oceno. Os x prestavlja število gruč, os y pa oceno.

podobna pri vseh treh modelih.

Za klasifikator posameznega koncepta lahko izrišemo graf konceptov z največjimi koeficienti. Tako dobimo preprost vpogled v to, kateri koncepti napovedujejo njegovo prisotnost oziroma odsotnost. Vpliv določenih konceptov na sliki 5.6 za koncept "Terrorism" je smiseln, kot na primer "Illegal immigration to the United States" ali "Israel-United States relations," medtem ko je prisotnost drugih konceptov, kot je "Monopoly", nesmiselna, in

model	`točnost`	`preciznost`	`priklic`	`F`
`eden-proti-vsem` $t = 14$	0.156	0.272	0.300	0.264
`eden-proti-vsem` $t = 30$	0.159	0.265	0.322	0.268
`konstantni klasifikator`	0.116	0.268	0.202	0.240

Tabela 5.1: Uspešnost modela eden-proti-vsem z uporabo klasifikatorja linearnih podpornih vektorjev.

model	število dreves	točnost	preciznost	priklic	F
naključni gozd	10	0.201	0.711	0.228	0.316
naključni gozd	50	0.200	0.755	0.221	0.314
naključni gozd	100	0.200	0.761	0.221	0.314
naključni gozd	200	0.200	0.763	0.220	0.314

Tabela 5.2: Uspešnost modela naključnih gozdov, $t = 14$.

je verjetno posledica šuma v podatkih. Prisotna sta tudi koncepta "Ashraf Ghani" in "Hugo Chávez," ki bi praviloma morala biti definirana kot osebi in ne bi smela biti prisotna v podatkih, vendar ju je Event Registry opredelil kot pojem.

Na sliki 5.7 je za koncept "Refugees of the Syrian Civil War" razvidno, da klasifikator pogosto napove odsotnost koncepta, če se je ta pojavil znotraj časovnega okna. Zato ima v tem grafu ta koncept največji negativni vpliv.

Koncepti za časovno okno velikosti 30 dni niso prikazani, ker pri njih ni bistvene razlike.

5.4 Metoda naključnih gozdov

Kriterij za določanje atributa za vejitev drevesa je mera *Gini*. Pri vsaki vejitvi upoštevamo 10 % vseh atributov. Minimalno število primerov v listih dreves omejimo na 5. Zgradimo štiri modele z različnim številom dreves - 10, 50, 100 in 200.

V tabeli rezultatov 5.2 je razvidno, da ima metoda naključnih gozdov bistveno boljšo preciznost, vendar slabši priklic kot metoda eden-proti-vsem. Pri več kot 50 drevesih ni videti bistvene izboljšave.

Iz rezultatov v tabeli 5.3 je razvidno, da se je priklic modela povečal pri večjem časovnem oknu. Večje število dreves ni izboljšalo kvalitete napovedi.

model	število dreves	točnost	preciznost	priklic	F
naključni gozd	10	0.215	0.663	0.252	0.338
naključni gozd	50	0.214	0.703	0.244	0.335
naključni gozd	100	0.214	0.715	0.243	0.335
naključni gozd	200	0.214	0.717	0.243	0.335

Tabela 5.3: Uspešnost modela naključnih gozdov, $t = 30$.

5.5 Nevronska mreža z vzvratnim razširjanjem napake BP-MLL

Naučili smo nevronsko mrežo s funkcijo napake BP-MLL pri treh različnih številih iteracij oziroma prehodov (10, 50 in 100), skozi celotno učno množico (ang. *epoch*). Mreža ima tri skrite nivoje, vsakega z 200 nevroni. Na prvem in zadnjem nivoju je število nevronov enako številu vhodnih in izhodnih atributov.

Zadnji nivo ima sigmoidno aktivacijsko funkcijo, ki je bolj primerna za večznačne primere, ker izhodne verjetnosti modelira kot neodvisne druga od druge. V nasprotnem primeru bi se verjetnosti na izhodnih nivojih seštele v 1, česar seveda nočemo. Pri napovedovanju za nevronske mreže z binarno navzkrižno entropijo izberemo tiste oznake z verjetnostjo večjo od 0.5, pri nevronskih mrežah z BP-MLL pa ta prag postavimo na 1.

Model smo dodatno primerjali še z nevronsko mrežo, ki uporablja običajno uporabljeno funkcijo napake za večznačno klasifikacijo, binarno navzkrižno entropijo (ang. *binary cross entropy*).

Iz rezultatov je razvidno, da doseže BP-MLL občutno boljši priklic kot binarna navzkrižna entropija in podobno preciznost. Najboljši rezultat doseže pri 50 iteracijah. Povečanje velikosti časovnega okna je izboljšalo priklic in poslabšalo preciznost. Model je dosegel boljše rezultate kot metodi naključnih gozdov in eden-proti vsem.

model	iteracij	točnost	preciznost	priklic	F
BP-MLL	10	0.230	0.398	0.391	0.365
BP-MLL	50	0.262	0.363	0.522	0.406
BP-MLL	100	0.244	0.307	0.596	0.383

Tabela 5.4: Uspešnost modela nevronske mreže z BP-MLL, $t = 14$.

model	iteracij	točnost	preciznost	priklic	F
BP-MLL	10	0.238	0.359	0.468	0.377
BP-MLL	50	0.258	0.344	0.555	0.402
BP-MLL	100	0.238	0.296	0.614	0.375

Tabela 5.5: Uspešnost modela nevronske mreže z BP-MLL, $t = 30$.

model	iteracij	točnost	preciznost	priklic	F
b. n. entropija	10	0.246	0.475	0.362	0.384
b. n. entropija	50	0.204	0.355	0.355	0.329
b. n. entropija	100	0.191	0.327	0.349	0.313

Tabela 5.6: Uspešnost modela nevronske mreže z binarno navzkrižno entropijo, $t = 14$.

model	iteracij	točnost	preciznost	priklic	F
b. n. entropija	10	0.239	0.431	0.378	0.377
b. n. entropija	50	0.199	0.328	0.37	0.323
b. n. entropija	100	0.186	0.309	0.354	0.306

Tabela 5.7: Uspešnost modela nevronske mreže z binarno navzkrižno entropijo, $t = 30$.

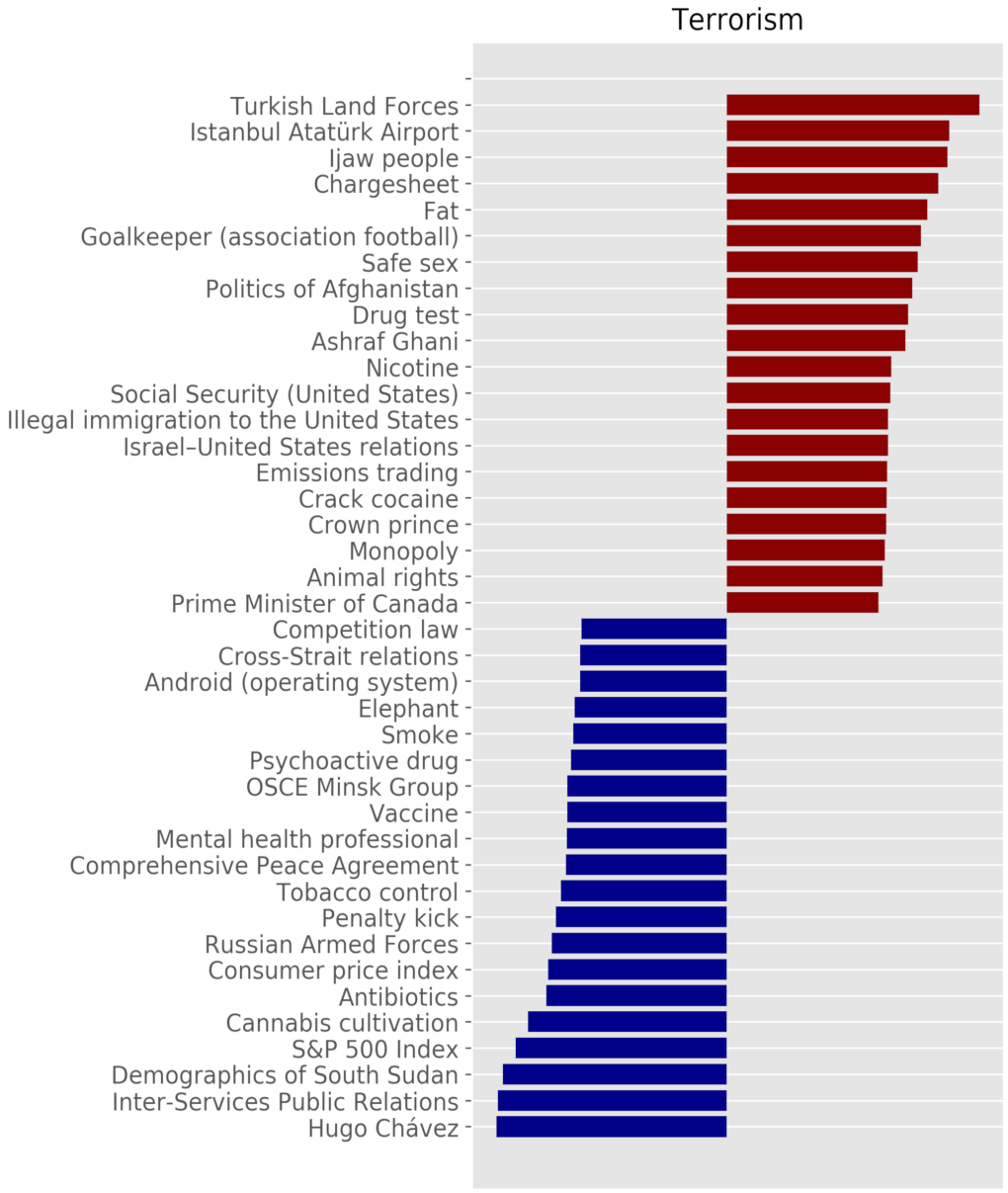
model	točnost	preciznost	priklic	F
konstantni klasifikator	0.115	0.258	0.209	0.202
eden-proti-vsem	0.146	0.227	0.327	0.250
naključni gozd, 10 dreves	0.200	0.639	0.239	0.318
nev. mreža, BP-MLL, 50 iter.	0.214	0.254	0.648	0.344

Tabela 5.8: Uspešnost modela nevronske mreže z binarno navzkrižno entropijo, na testnih podatkih, $t = 30$.

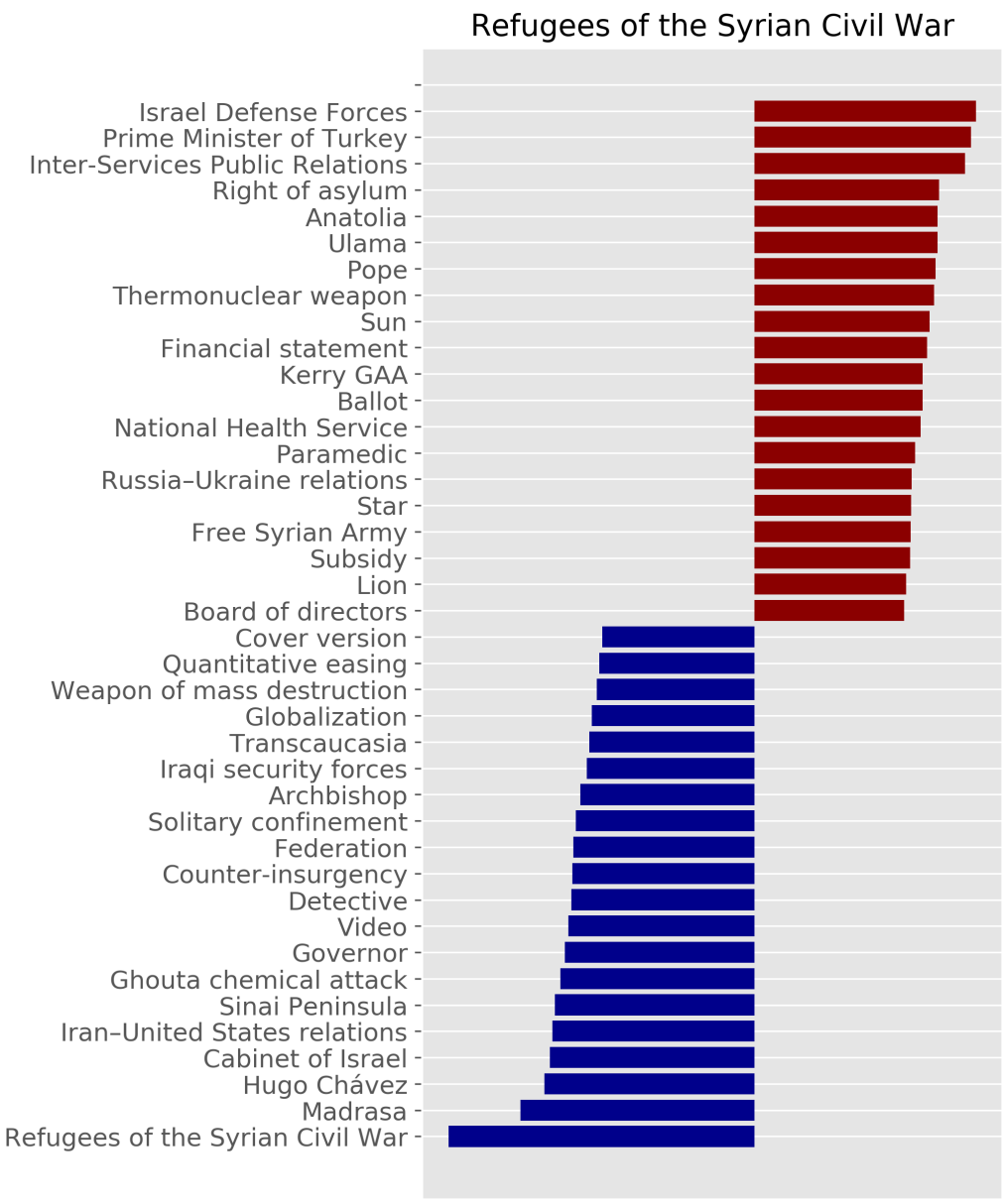
5.6 Testiranje napovednih modelov

Uspešnost modelov z najboljšo mero F smo ocenili še na testni množici. Testna množica je sestavljena iz dogodkov, ki so se zgodili med 1. 1. 2017 in 30. 4. 2017. Uporabljena je dolžina časovnega okna 30 dni, ker je v splošnem dajala boljše rezultate.

Rezultati v tabeli 5.8 so primerljivi, čeprav malce slabši od ekvivalentnih modelov ocenjenih na validacijski množici. Glede na mero F so najuspešnejše nevronske mreže.



Slika 5.6: Koncepti z največjimi koeficienti za klasifikator SVC, za koncept "terorizem."



Slika 5.7: Koncepti z največjimi koeficienti za klasifikator SVC, za koncept "begunci sirijske civilne vojne."

Poglavje 6

Zaključek

Rezultati so pokazali, da lahko z modeliranjem odnosov med dogodki kot gruče izvlečemo koristne informacije za napovedovanje karakteristik prihajajočih dogodkov. Prav tako je iz rezultatov razvidno, da lahko z večjim časovnim oknom dosežemo boljši priklic. Metoda naključnih gozdov je dosegla dobro preciznost, a slabši priklic kot metoda eden-proti-vsem. Model nevronske mreže z BP-MLL je dosegel daleč najboljši priklic. Čeprav ima slabšo preciznost od metode naključnih gozdov, doseže najboljše ravnotežje med tema dvema merama, kot je razvidno iz mere F.

Menimo, da bi bilo napovedovanje mogoče bistveno izboljšati, če bi gručenje prilagodili tako, da bolj strogo upošteva pravilo kronološke povezanosti, na primer z uporabo principa, na katerem temelji metoda *Ward* pri hierarhičnem gručenju. Pri združevanju gruč bi lashko minimizirali razdaljo med dogodki, ki si sledijo kronološko.

Na delovanje pristopa vpliva izbor pričakovanega števila gruč. Postopek smo omejili na iskanje 100 gruč, predvsem zaradi računske zahtevnosti. Ker je to parameter, ki ima lahko znaten vpliv na rezultate, tako na nivoju napovedovanja kot na nivoju validacije, bi v nadaljnjem delu bilo vredno preučiti vpliv izbora števila gruč na uspešnost modela.

Drugi pomemben parameter pri gradnji modelov je velikost časovnega okna. Upoštevali smo dve velikosti časovnega okna: 14 in 30 dni. Večja

velikost časovnega okna je v večini primerov izboljšala priklic. Verjetno je, da bi s povečanjem časovnega okna lahko dosegli še boljšo natančnost pri napovedovanju konceptov, ki se s preteklimi koncepti povezujejo preko daljšega obdobja. Glede na rezultate bi bilo vredno preveriti tudi uspešnost napovedi, ki kombinirajo ansambel modelov z različnimi velikosti časovnega okna.

V diplomskem delu smo se omejili na podatke iz obdobja enega leta. Pri uporabi večje količine podatkov za daljše obdobje bi bilo koristno prilagoditi pristop do gručenja tako, da bi omejili največji časovni razpon posamezne gruče in omogočili dinamično dodajanje gruč glede na prihajajoče podatke.

Pri obravnavanem problemu gre za sekvenčne podatke. Vrstni red, v katerem se pojavijo koncepti znotraj časovnega okna pri gradnji podatkovne zbirke za napovedovanje ne upoštevamo. Model, ki upošteva tudi vrstni red, bi lahko potencialno dosegel boljšo natančnost, na primer z uporabo rekurentnih nevronskih mrež (ang. *recurrent neural networks*).

Literatura

[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001.

[2] Aydin Buluç, Jeremy T Fineman, Matteo Frigo, John R Gilbert, and Charles E Leiserson. Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks. In *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures*, pages 233–244. ACM, 2009.

[3] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. 2009.

[4] Gregor Leban, Blaž Fortuna, Janez Brank, and Marko Grobelnik. Cross-lingual detection of world events from news articles. In *CEUR Workshop Proceedings*, volume 1272, 01 2014.

[5] Gregor Leban, Blaž Fortuna, Janez Brank, and Marko Grobelnik. Event registry: learning about world events from news. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 107–110. ACM, 2014.

[6] Nicholas Rescher. *Predicting the future: An introduction to the theory of forecasting*. SUNY press, 1998.

[7] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[8] John A Sipple. System and method for predicting events, November 18 2014. US Patent 8,892,484.

[9] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[10] PJ Werbos. New tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*, 1974.

[11] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.