

2019 < EVILKA 3 < JUL./AVG./SEP. < LETNIK XXVII < ISSN 1318-1882

03 U P O R A B N A INFORMATIKA

U P O R A B N A I N F O R M A T I K A

2019 ŠTEVILKA 3 JUL/AVG/SEP LETNIK XXVII ISSN 1318-1882

▣ Znanstveni prispevki

- Tamara Čuček, Boštjan Šumak, Maja Pušnik:
Analiza orodij za podporo optimizacij procesov 83
- Matej Ulčar, Simon Dobrišek, Marko Robnik-Šikonja:
Razpoznavanje slovenskega govora z metodami globokih nevronske mreže 96

▣ Kratki znanstveni prispevki

- Tadej Škvorc, Marko Robnik-Šikonja:
Prepoznavanje idiomatskih besednih zvez z uporabo besednih vložitev 110
- Borja Bovcon, Matej Kristan:
Primerjava segmentacijskih metod globokega učenja za detekcijo ovir na vodi 115
- Radko Polipović, Patricio Bulić:
HYLO: Hibradni logaritmični množilnik za energijsko učinkovito računanje 120

▣ Razprave

- Cene Bavec:
Analiza digitalne preobrazbe Slovenije v EU okolju 124

▣ Informacije

- Iz Islovarja 131

Ustanovitelj in izdajatelj

Slovensko društvo INFORMATIKA
Litostrojska cesta 54, 1000 Ljubljana

Predstavniki

Niko Schlamberger

Odgovorni urednik

Saša Divjak

Uredniški odbor

Andrej Kovačič, Evelin Krmac, Ivan Rozman, Jan Mendling, Jan von Knop, John Taylor, Jurij Jaklič, Lili Nemeč Zlatolas, Marko Hölbl, Mirjana Kljajić Borštnar, Mirko Vintar, Pedro Simões Coelho, Saša Divjak, Sjaak Brinkkemper, Slavko Žitnik, Tatjana Welzer Družovec, Vesna Bosilj-Vukšić, Vida Groznik, Vladislav Rajkovič

Recenzenti

Alenka Baggia, Andrej Kovačič, Bojan Rosi, Denis Trček, Franc Solina, Gregor Weiss, Igor Bernik, Janez Demšar, Jure Erjavec, Jurij Jaklič, Luka Tomat, Marjan Heričko, Marko Hölbl, Martin Vodopivec, Matevž Pesek, Matija Marolt, Mihaela Triglav Čekada, Mirjana Kljajić Borštnar, Mojca Indihar Štemberger, Monika Klun, Niko Lukač, Niko Schlamberger, Peter Trkman, Saša Divjak, Slavko Žitnik, Tomaž Dobravec, Tomaž Erjavec, Štefan Kohek, Uroš Rajkovič, Vladislav Rajkovič, Živa Rant

Tehnični urednik

Slavko Žitnik

Lektoriranje angleških izvlečkov

Marvelingua (angl.)

Oblikovanje

KOFEIN DIZAJN, d. o. o.

Prelom in tisk

Boex DTP, d. o. o., Ljubljana

Naklada

200 izvodov

Naslov uredništva

Slovensko društvo INFORMATIKA
Uredništvo revije Uporabna informatika
Litostrojska cesta 54, 1000 Ljubljana
www.uporabna-informatika.si

Revija izhaja četrtletno. Cena posamezne številke je 20,00 EUR. Letna naročnina za podjetja 85,00 EUR, za vsak nadaljnji izvod 60,00 EUR, za posameznike 35,00 EUR, za študente in seniorje 15,00 EUR. V ceno je vključen DDV.

Revija Uporabna informatika je od številke 4/VII vključena v mednarodno bazo INSPEC.

Revija Uporabna informatika je pod zaporedno številko 666 vpisana v razvid medijev, ki ga vodi Ministrstvo za kulturo RS.

Revija Uporabna informatika je vključena v Digitalno knjižnico Slovenije (dLib.si).

© Slovensko društvo INFORMATIKA

Vabilo avtorjem

V reviji Uporabna informatika objavljamo kakovostne izvirne članke domačih in tujih avtorjev z najširšega področja informatike v poslovanju podjetij, javni upravi in zasebnem življenju na znanstveni, strokovni in informativni ravni; še posebno spodbujamo objavo interdisciplinarnih člankov. Zato vabimo avtorje, da prispevke, ki ustrezajo omenjenim usmeritvam, pošljejo uredništvu revije po elektronski pošti na naslov ui@drustvo-informatika.si.

Avtorje prosimo, da pri pripravi prispevka upoštevajo navodila, objavljena v nadaljevanju ter na naslovu <http://www.uporabna-informatika.si>.

Za kakovost prispevkov skrbi mednarodni uredniški odbor. Članki so anonimno recenzirani, o objavi pa na podlagi recenzij samostojno odloča uredniški odbor. Recenzenti lahko zahtevajo, da avtorji besedilo spremenijo v skladu s priporočili in da popravljeni članek ponovno prejmejo v pregled. Uredništvo pa lahko še pred recenzijo zavrne objavo prispevka, če njegova vsebina ne ustreza vsebinski usmeritvi revije ali če članek ne ustreza kriterijem za objavo v reviji.

Pred objavo članka mora avtor podpisati izjavo o avtorstvu, s katero potrjuje originalnost članka in dovoljuje prenos materialnih avtorskih pravic. Nenaročenih prispevkov ne vračamo in ne honoriramo. Avtorji prejmejo enoletno naročnino na revijo Uporabna informatika, ki vključuje avtorski izvod revije in še nadaljnje tri zaporedne številke.

S svojim prispevkom v reviji Uporabna informatika boste prispevali k širjenju znanja na področju informatike. Želimo si čim več prispevkov z raznoliko in zanimivo tematiko in se jih že vnaprej veselimo.

Uredništvo revije

Navodila avtorjem člankov

Članke objavljamo praviloma v slovenščini, članke tujih avtorjev pa v angleščini. Besedilo naj bo jezikovno skrbno pripravljeno. Priporočamo zmernost pri uporabi tujk in – kjer je mogoče – njihovo zamenjavo s slovenskimi izrazi. V pomoč pri iskanju slovenskih ustreznih priporočamo uporabo spletnega terminološkega slovarja Slovenskega društva Informatika Islovar (www.islovar.org).

Znanstveni članek naj obsega največ 40.000 znakov, strokovni članki do 30.000 znakov, obvestila in poročila pa do 8.000 znakov.

Članek naj bo praviloma predložen v urejevalniku besedil Word (*.doc ali *.docx) v enojnem razmaku, brez posebnih znakov ali poudarjenih črk. Za ločilom na koncu stavka napravite samo en prazen prostor, pri odstavkih ne uporabljajte zamika.

Naslovu članka naj sledi za vsakega avtorja polno ime, ustanova, v kateri je zaposlen, naslov in elektronski naslov. Sledi naj povzetek v slovenščini v obsegu 8 do 10 vrstic in seznam od 5 do 8 ključnih besed, ki najbolje opredeljujejo vsebinski okvir članka. Pred povzetkom v angleščini naj bo še angleški prevod naslova, prav tako pa naj bodo dodane ključne besede v angleščini. Obratno velja v primeru predložitve članka v angleščini. Razdelki naj bodo naslovljeni in oštevilčeni z arabskimi številkami.

Slike in tabele vključite v besedilo. Opremite jih z naslovom in oštevilčite z arabskimi številkami. Vsako sliko in tabelo razložite tudi v besedilu članka. Če v članku uporabljate slike ali tabele drugih avtorjev, navedite vir pod sliko oz. tabelo. Revijo tiskamo v črno-beli tehniki, zato barvne slike ali fotografije kot original niso primerne. Slik zaslonov ne objavljamo, razen če so nujno potrebne za razumevanje besedila. Slike, grafikoni, organizacijske sheme ipd. naj imajo belo podlago. Enačbe oštevilčite v oklepajih desno od enačbe.

V besedilu se sklicujte na navedeno literaturo skladno s pravili sistema APA navajanja bibliografskih referenc, najpogosteje torej v obliki (Novak & Kovač, 2008, str. 235). Na koncu članka navedite samo v članku uporabljeno literaturo in vire v enotnem seznamu po abecednem redu avtorjev, prav tako v skladu s pravili APA. Več o sistemu APA, katerega uporabo omogoča tudi urejevalnik besedil Word 2007, najdete na strani <http://owl.english.purdue.edu/owl/resource/560/01/>.

Članku dodajte kratek življenjepis vsakega avtorja v obsegu do 8 vrstic, v katerem poudarite predvsem strokovne dosežke.

Analiza orodij za podporo optimizacij procesov

Tamara Čuček, Boštjan Šumak, Maja Pušnik
 Fakulteta za elektrotehniko, računalništvo in informatiko
 Koroška cesta 46, 2000 Maribor, Slovenija
 tamaraa.cucek@gmail.com, bostjan.sumak@um.si, maja.pusnik@um.si

Izvleček

Področje optimizacije poslovnih procesov je pomembna a slabo podprta domena v slovenskih podjetjih. Zato smo v prispevku naslovili to področje ter se osredotočili na analizo in predstavitev orodij, ki nudijo podporo optimizaciji procesov. Cilj prispevka je analizirati in predstaviti različna orodja, ki uporabnikom nudijo podporo pri razumevanju in preoblikovanju procesov. Ugotoviti smo želeli, katero izmed orodij nudi najboljšo podporo ter katere so ključne razlike med orodji. Dodatno smo želeli raziskati poznavanje analiziranih orodij in njihovo uporabo na oddelkih informatike v slovenskih podjetjih. Rezultati raziskav so pokazali, da se orodja med seboj razlikujejo glede na možnosti, ki jih ponujajo, ter notacije, ki jih podpirajo. Rezultati prav tako nakazujejo, da je poznavanje in uporaba orodij za optimizacijo poslovnih procesov v slovenskih podjetjih pomanjkljiva ter da slovenska podjetja poslovne procese redko sistematično optimizirajo.

Ključne besede: BPMN, poslovni proces, prenova procesov, simulacija, optimizacija poslovnih procesov, orodja za podporo optimizacije, orodja za modeliranje poslovnih procesov

Abstract

The domain of business process optimization is important but insufficiently supported in Slovenian companies. In this paper, we addressed the topic of optimization and focus on the analysis of tools that support process optimization. The aim of the paper was to analyse and present the various tools that provide users support in the understanding and transforming of processes, focusing on finding out which of the tools offer greatest support and what the key differences between the tools are. In addition, exploring the knowledge of analysed tools and their use in IT departments of Slovenian companies was also one of the goals. The results of the research show that the tools differ depending on the possibilities that they offer and the notations that they support. Research results also suggest that the knowledge and use of business process optimization tools in Slovenian companies is insufficient and that Slovenian companies rarely optimize business processes in a systematic way.

Keywords: BPMN, business process, process change, simulation, optimization of business processes, tools for optimization support, tools for business process modelling

1 UVOD

Poslovni procesi so pomemben del podjetij in drugih organizacij, za katera je pomembno, da poslovne procese izvajajo učinkovito in uspešno ter jih nenehno izboljšujejo in prenavljajo (DeMast & Lokkerbol, 2012, str. 604). Nenehno izboljševanje poslovnih procesov znižuje stroške organizacije, skrajša čas in izboljša kakovost poslovanja. Navedeno predstavlja konkurenčno prednost in je pomembno za preživetje podjetja na trgu (Møller et al, 2007) (Kahkesh, 2012) (Witts, 2016).

Za čim bolj učinkovito izvajanje poslovnih procesov si lahko podjetja pomagajo z informacijsko tehnologijo, ki omogoča več prednosti; predvsem večjo učinkovitost in uspešnost poslovnih procesov, ki se med drugim meri s pomočjo rezultatov porabe virov (človeški viri, finančni viri, surovine, čas) (Møller et al, 2007). Ustrezna informatizacija pri prenovi oziroma optimizaciji poslovnih procesov ter uvedba upravljanja poslovnih procesov prinašata več splošnih koristi, kot so: znižanje stroškov izvajanja procesa, dvig kakovosti, manjše število napak, krajši čas izva-

janja procesa, znižanje stroškov in krajšanje časa šolanja izvajalcev procesa, zmanjševanje števila pritožb strank, izboljšanje natančnosti načrtovanja in mnoge druge (Møller et al, 2007). V sklopu prispevka bomo analizirali, ali različna orodja pri navedenih aktivnostih dejansko ponujajo podporo.

V nadaljevanju sledi predstavitev pojma optimizacije poslovnih procesov, analiza in opis podpornih orodij ter anketa med potencialnimi uporabniki orodij ter zaključne misli.

2 POMEN OPTIMIZACIJE POSLOVNIH PROCESOV

Optimizacija je bistvena za uspeh poslovanja in kakovosti končnega izdelka oziroma storitve (Witts, 2016). Nenehen pregled in prenova procesov sta del strateškega načrta in pomembna člena upravljanja poslovnih procesov. Nekaj glavnih prednosti optimizacije je povzetih v nadaljevanju (Abdelgeer et al, 2013).

- **Optimizacija poslovnih procesov izboljšuje učinkovitost:** Skupni izziv za mnoge organizacije je pogosta neučinkovitost poslovnih procesov, kar povzroči povečanje potrat, tako časovno kot finančno. Prvi korak je ugotoviti trenutno stanje, nato sledi postopno uvajanje sprememb.
- **Optimizacija poslovnih procesov podpira prenos poslovnega znanja:** Sistematičen način zajemanja in shranjevanja pomembnih podatkov ter dokumentiranje informacij omogočata, da procesi in poslovno znanje ostanejo znotraj organizacije ter se ustrezno prenesejo.
- **Optimizacija poslovnih procesov zagotavlja okvir za nenehno izboljševanje:** Dolgoročen pristop, s katerim postopoma vpeljujemo majhne izboljšave, omogoča natančno preučevanje vpliva sprememb na proces.
- **Optimizacija poslovnih procesov odpravlja preobremenjenost:** Nerazumevanje, kaj in koliko počnejo drugi udeleženci procesa, povzroči težave, preobremenitve in podvajanja. Z optimizacijo in vidnostjo poslovnih procesov se težave prekrivajo in neenakomerne porazdelitve dela zmanjšajo.

2.1 Upravljanje in optimizacija poslovnih procesov

Upravljanje poslovnih procesov je celovit in integriran koncept vodenja, organiziranja, informatizacije in nadzora ter optimizacije. V ciklu upravljanja po-

slovnih procesov poznamo štiri faze (Ko et al., 2009; van der Aalst, ter Hofstede, & Weske, 2003):

- **Faza načrtovanja procesa:** v fazi načrtovanja so procesi (ponovno) načrtovani. V tej fazi prevladujejo grafični standardi.
- **Faza konfiguracije procesa:** ta faza konfigurira osnovno sistemsko infrastrukturo (npr. sinhronizacijo vlog in organizacijskih grafik iz računov zaposlenega v aktivnem imeniku podjetja). To fazo je težko standardizirati zaradi različnih IT struktur različnih podjetij.
- **Faza sprejetja procesa:** v tej fazi se izvedejo operativni poslovni procesi z uporabo sistema, ki je konfiguriran. Tukaj prevladujejo standardi izvrševanja.
- **Faza diagnoze:** glede na primerne analize in orodja za spremljanje se lahko ugotovijo in odpravijo ozka grla ter morebitne goljufije v poslovnih procesih. Orodja za to so vključena v diagnostične standarde.

Optimizacija poslovnih procesov je pomemben člen v sklopu upravljanja poslovnih procesov in predstavlja nadgradnjo obstoječih procesov, pri čemer obstaja več ključnih elementov, predstavljenih v nadaljevanju.

2.2 Ključni elementi optimizacije poslovnih procesov

V nadaljevanju sledi opis 6 ključnih elementov optimizacije poslovnih procesov (Rosemann & Brocke, 2010).

2.2.1 Strateško usklajevanje

Optimizacija poslovnih procesov mora biti usklajena s splošno strategijo organizacije. Strateško usklajevanje (ali sinhronizacija) je opredeljeno kot tesna povezava organizacijskih prednostnih nalog in poslovnih procesov, ki omogočajo nenehne in učinkovite ukrepe za izboljšanje poslovne uspešnosti. Procesni morajo biti zasnovani, izvedeni, upravljani in merjeni v skladu s strateškimi prednostnimi nalogami in posebnimi strateškimi situacijami. V zameno lahko specifične procesne zmožnosti (npr. konkurenčna prednost glede na čas za izvedbo ali spremembo postopka) ponudijo priložnosti za obveščanje o oblikovanju strategije, ki vodi k strategijam, ki dejansko omogočajo proces (Rosemann & Brocke, 2010).

2.2.2 Upravljanje

Upravljanje optimizacije poslovnih procesov vzpostavlja primerno in pregledno odgovornost glede vlog in odgovornosti za različne ravni optimizacije poslovnih procesov (portfelja, programa, projektov in operacij). Nadaljnji poudarek je na oblikovanju postopkov odločanja in nagrajevanja za usmerjanje ukrepov, povezanih s procesi (Rosemann & Brocke, 2010).

2.2.3 Metode

Metode, v kontekstu optimizacije poslovnih procesov, so opredeljene kot skupek orodij in tehnik, ki podpirajo in omogočajo dejavnosti v življenjskem ciklu procesa. Primeri so metode, ki olajšajo procesno modeliranje ali analizo procesov, in tehnike izboljšanja postopka (Rosemann & Brocke, 2010).

2.2.4 Informacijska tehnologija

Rešitve, ki temeljijo na IT, so pomembne za pobude optimizacij poslovnih procesov. S tradicionalnim podarkom na procesni analizi (npr. statističnem procesnem nadzoru) in podpori modeliranju procesov se informacijske rešitve, povezane z optimizacijo poslovnih procesov, vedno bolj manifestirajo v obliki procesnih informacijskih rešitev. Ozaveščanje procesov pomeni, da programska oprema izrecno razume proces, ki ga je treba izvesti (Rosemann & Brocke, 2010).

2.2.5 Ljudje

Ljudje so kot bistveni element optimizacije poslovnih procesov opredeljeni kot posamezniki in skupine, ki nenehno izboljšujejo in uporabljajo svoje procesne veščine ter znanje za izboljšanje poslovne uspešnosti. Posledično ta dejavnik zajame zmogljivosti optimizacije poslovnih procesov, ki se odražajo v človeškem kapitalu organizacije in njenega ekosistema (Rosemann & Brocke, 2010).

2.2.6 Kultura

Kultura optimizacije poslovnih procesov vključuje kolektivne vrednote in prepričanja v zvezi s procesno usmerjeno organizacijo. Namenjena je ustvarjanju olajševalnega okolja, ki dopolnjuje različne pobude optimizacij poslovnih procesov. Vpliv dejavnosti, povezanih s kulturo, ima precej daljše časovno obdobje kot dejavnosti, povezane s katerikoli drugim dejavnikom (Rosemann & Brocke, 2010).

Na trgu obstaja več plačljivih in prosto dostopnih orodij, ki nudijo podporo optimizaciji procesov ter z različnimi mehanizmi omogočajo doseganje naštetih optimizacijskih prednosti. V sklopu prispevka smo izbrali 5 najpogosteje uporabljenih oziroma pogosto omenjenih optimizacijskih pomočnikov na strokovnih forumih. Prav tako smo orodja poiskali s pomočjo ključnih besed v brskalniku in te primerjali z drugimi sorodnimi orodji. Izbrana so največkrat omenjena, prosto dostopna orodja. Optimizacijski pomočnik predstavlja orodje, ki uporabnikom nudi podporo v zaključni fazi upravljanja poslovnih procesov ter omogoča natančen vpogled v lastnosti in učinkovitost izvajanja procesa. V ta namen mora nuditi vsaj nekatere izmed navedenih funkcionalnosti: modeliranje, validacija modela, simulacija, nadzor in druge. Z njimi smo poskusili izvesti cikel optimizacije poslovnih procesov.

3 ORODJA ZA OPTIMIZACIJO POSLOVNIH PROCESOV

Na podlagi strokovne in znanstvene literature smo analizirali 5 pogosteje uporabljenih orodij za pomoč modeliranju, simuliranju in optimizaciji poslovnih procesov. Namen pregleda literature je bil raziskati, analizirati in predstaviti najpogosteje uporabljena orodja za podporo optimizacij procesov, ki so tudi (vsaj delno brezplačno) dostopna. Za izbrana orodja smo raziskali lastnosti ter ovrednotil možnosti razvoja optimizirane rešitve. Vključenih je bilo več avtorjev oziroma komercialnih spletnih strani, ki so orodja predstavili ali analizirali: (Adonis, 2019), (BOC ITC AG, 2009, 2010), (Bischoff & Van Dinther, 2016), (Bizagi, 2009, 2014, 2016), (Bork&Fill, 2014), (Delgado et al, 2016), (Sparks & Maxwell, 2010), (Ward-Dutton, 2010).

Analiza orodij je naslavljala možnosti modeliranja, optimizacije, simuliranja, validacije, nadzora in prenosa iz BPMN (Business Process Model and Notation) v izvršilno obliko BPEL (Business Process Execution Language). Primarna ugotovitev je bila, da se orodja med seboj zelo razlikujejo, čeprav so namenjena enakemu cilju. Vsem orodjem je skupno to, da ponujajo modeliranje diagramov poslovnih procesov, sicer pa so zelo raznolika. Razlogi za razlike so predstavljeni v knjigi Fundamentals of Business Process Management (Dumas & La Rosa & Mendling & Reijers, 2013). Razlike so v vrsti modelov, ki jih omogočajo, tipu notacije, ki jih ponujajo, podpori za simulacijo in možnosti prenosa iz notacije BPMN. Sledenje optimizaciji oziroma možnosti nadzora

(nadzorne plošče) ne podpira nobeno izmed analiziranih orodij. Več informacij ter primerjava orodij je predstavljena v nadaljevanju. Za vsako orodje so posebej izpostavljene lastnosti, zlasti glede možnosti modeliranja, simulacije, validacije ter prenosa iz BPMN v BPEL, če so le te funkcionalnosti podprte.

3.1 Enterprise Architect

Enterprise Architect je vizualna platforma, ki jo je razvilo podjetje Sparx Systems in je bila prvič izdana leta 2000. Namenjena je načrtovanju in izdelavi programskih sistemov, modeliranju poslovnih procesov ter modeliranju domen, ki temeljijo na industriji. Zajema vse vidike razvojnega cikla in podpira popolno sledljivost od začetne faze modeliranja do uvajanja, vzdrževanja in testiranja. Ponuja vgrajeno podporo najnovejšega zapisa modeliranja poslovnih procesov (BPMN 2.0), ki so simulirani s pomočjo vgrajenega simulacijskega pogona. S tem orodje omogoča dinamičen pregled tokov procesa. S pomočjo vgrajenega simulacijskega pogona prav tako zbira podatke o času in vire o predlaganih in starih procesih, s katerimi pomaga izboljšati poslovni proces. Ima več prednosti, navedenih v nadaljevanju (Sparks, G. & Maxwell, 2010; Sparx Systems, 2010):

- modeliranje na osnovi UML 2.5,
- vgrajeno upravljanje zahtev,
- podpora za upravljanje projektov, vključno z viri, nalogami, koledarjem in meritvami,
- vgrajeno testno upravljanje: upravljanje testnih točk, izvedba testiranja na podlagi modela, specifikacija testnega primera in podpora za JUnit in NUnit,
- prilagodljive možnosti dokumentacije: HTML, PDF in DOCX,
- podpora kode za številne jezike,
- integriran analizator vizualnih izvedb za profiliranje, odpravljanje napak in dokumentiranje izvajanja aplikacij,
- razširljivo modelirano okolje, ki lahko gostuje uporabniško določene profile in tehnologije,
- preprosto in hitro modeliranje z UML,
- obvladovanje velikih modelov in sočasnih uporabnikov.

3.1.1 Modeliranje in Simulacija

Platforma omogoča več tipov modeliranja: diagram razredov, diagram aktivnosti, diagram zaporedij, časovni diagram, diagram primerov uporabe, dia-

gram pregledov interakcij, komunikacijski diagram, diagram stanj naprav, diagram poslovnih procesov. Dodatno omogoča simulacijo in izvedbo poslovnega modela. Specifikacija BPSim omogoča konfiguracijo in dodeljevanje virov posameznim dejavnostim ali nalogam, zbiranje dogodkov, odločitev in drugih zmogljivosti v realnem svetu. Ko je model konfiguriran skladno s specifikacijo BPSim, se lahko pošlje na simulator in teče po konfiguriranih podatkih. Simulacija doprinese več prednosti:

- boljše razumevanje, kako model dejansko deluje v času izvajanja,
- preverjanje, ali vedenjski modeli opisujejo pravi len potek procesa ali dogodka,
- preverjanje pravilnosti delovanja okvirjev uporabniškega vmesnika še pred izvajanjem,
- ugotovitev morebitnih ozkih grl, neučinkovitosti in drugih težav v sistemskem modelu ali poslovnem procesu in
- zaznavanje napak zgodaj v razvojnem ciklu – preden se prične z izvajanjem sredstev.

Sparx Systems prav tako kot dodatek ponuja simulator MDG BPSim Execution Engine. Ta dodatek se integrira z modeli BPSim in BPMN, ki so definirani v Enterprise Architect in zagotavljajo zmožnost zagona in shranjevanja rezultatov različnih simulacij ter njihovo primerjavo. S pomočjo teh primerjav lahko analitiki in poslovni strategji ugotovijo, katera konfiguracija je za posamezen poslovni proces najbolj primerna (Sparx Systems, 2010).

3.1.2 Validacija

Enterprise Architect omogoča tudi validacijo BPMN modelov. Ob modelu se z desnim klikom v spustnem meniju izbere možnost »Validate Diagram«. Ob validaciji se odpre okno »System Output«, v katerem se izpiše rezultat validacije.

3.1.3 Prenos iz BPMN v BPEL

Dodatno se s platformo Enterprise Architect lahko ustvarijo skripta za izvajanje poslovnih procesov (BPEL) iz modelov BPMN. Specifična pravila validacije modela pomagajo zagotoviti, da modeli ustvarijo veljavno kodo BPEL.

3.2 Bizagi Modeler

Bizagi Modeler je orodje, ki ga je razvilo podjetje Bizagi in je bilo prvič izdano leta 2008. Omogoča vi-

zualno ustvarjanje diagramov, modeliranje in dokumentiranje poslovnih procesov v BPMN. Prav tako omogoča enostaven uvoz in izvoz v Visio ali XML ter druga orodja, ki jih potrebujemo za optimizacijo poslovnih procesov. Iz modelov se lahko generira dokumentacija v obliki formatov Word, PDF, Share-Point in Wiki¹. Vsaka datoteka se imenuje model in lahko vsebuje enega ali več diagramov. Model se lahko nanaša na celotno organizacijo, oddelek ali specifični proces. V primeru, da v modelu obstaja več diagramov, se diagrami razporedijo po posameznih zavahkih (Delgado, Calejari, & Arrigoni, 2016).

3.2.1 Modeliranje in simulacija

Bizagi Modeler omogoča simulacijo poslovnih procesov v okviru BPSim. Simulacijo se lahko izvede samo v primeru, ko je model poslovnega procesa celoten. Za popolno simulacijsko analizo se morajo izvesti štiri neodvisni nivoji. Vsak naslednji nivo vsebuje dodatne informacije, ki imajo večjo kompleksnost od prejšnjega nivoja, s čimer zagotavlja podrobno analizo poslovnih procesov. Nivoji so predstavljeni v nadaljevanju (Abdelgeer, F. M. Z., Dawood O. O. S., 2013; Bizagi, 2014; Bizagi Modeler, 2016):

- **Validacija procesa;** Prvi in najosnovnejši nivo simulacije za oceno strukture procesnega diagrama. Potrebno je oceniti deleže zaporednih tokov, ki zagotavljajo osnovo za usmerjanje. Prav tako potrebuje vrednost sprožilnega števca, ki ga vsebujejo začetni dogodki. Rezultat je prikaz vseh poti, ki so bile aktivirane med izvajanjem, in/ali so bili vsi procesi res končani. Dodatno oceni, koliko procesov je prešlo skozi vsak sekvenčni tok, vsako aktivnost do končnega dogodka.
- **Časovna analiza;** Drugi nivo simulacije za merjenje končnega časa procesa. Zraven podatkov, ki jih je potrebno vnesti v prvem nivoju, so prav tako potrebni predvideni časovni zaporedi vsake aktivnosti in čas intervala med generiranjem procesov. Ti podatki so lahko bodisi konstantni bodisi vzorci iz statističnih porazdelitev. Rezultati prikazujejo pretočne čase procesov. Predstavljeni so kot: najmanjše, najvišje, povprečje in vsota vseh časovnih obdelav.
- **Analiza virov;** predvideva, kako bo proces opravljal z različnimi ravnmi virov. Ta nivo zagotavlja zanesljivo oceno, kako se bo proces izvajal v obra-

tovanju. Ta nivo vključuje opredelitev virov in/ali vlog: koliko jih je na voljo in kje se uporabljajo. Zaradi vključitve virov je treba čas dejavnosti prilagoditi, da predstavlja dejanski čas dela (zamuda zaradi nedostopnosti osebja se navede izrecno). Struktura rezultatov je podobna časovni analizi. Prikazan je tudi porabljen čas, ki ga porabimo za vsako vrsto vira.

- **Analiza koledarja;** vključuje informacije o koledarju, ki odražajo učinkovitost procesa v dinamičnih časovnih obdobjih, kot so izmene, dnevni urniki ali tedni. Privzeto Bizagi Modeler vključuje koledar, ki deluje 24 ur na dan. Če se drugega koledarja ne določi, je privzeta nastavitev, da bodo določeni viri vedno na voljo. Struktura rezultatov je podobna analizi virov.

3.2.2 Validacija

Bizagi Modeler omogoča validacijo BPMN diagramov. Kadar validator zazna napako, se zraven osnovnega sporočila prikaže še bolj podrobno sporočilo o napaki. Podpira zgolj notacijo BPMN.

3.3 Aris Express

ARIS Express temelji na metodi ARIS in industrijskih standardih. Pametna zasnova uporabnikom omogoča hitro in enostavno zajemanje podatkov o podjetju, ki temeljijo na preglednicah. Uporabniki se lahko osredotočijo na vsebino in jim ni treba skrbeti za standarde modeliranja ali pravilno namestitve gradnikov. Model je ustvarjen takoj po vnašanju podatkov in ga je mogoče znova spremeniti. Pomoč pri modeliranju je ideja o modelnih fragmentih. Pri modeliranju procesnih modelov ali organizacijskih struktur morajo uporabniki poskrbeti za konvencionalne načine modeliranja. Zaradi tega je možna opredelitev fragmentov kot kombinacija predmetov, ki jih je mogoče ponovno uporabiti za nadaljnje modeliranje. Vključuje naslednje funkcije (Bork & Fill, 2014; Aris Community; Ward-Dutton, 2010):

- hitro modeliranje s pomočjo mini orodne vrstice za umestitev predmeta, občutljivega na kontekst,
- zajemanje podatkov na podlagi preglednic in avtomatsko generiranje modelov s pametnimi oblikovalskimi funkcijami,
- izdelava fragmentov za pogosto uporabljene kombinacije predmetov,

¹ Sestavljen je iz sintakse in ključnih besed, ki se uporablja za formatiranje strani.

- ponovna uporaba rezultatov v profesionalnih orodjih ARIS BPM.

3.3.1 Modeliranje in simulacija

Orodje ne omogoča simulacije, nudi pa podporo različnim tipom modelov: organizacijska shema, procesna pokrajina, poslovni proces, podatkovni model, diagram IT sistema, diagram systemske pokrajine in diagram poslovnih procesov. Možni tipi notacij: BPMN, EPC in ERM.

V sklopu raziskave smo uporabili brezplačno verzijo programa ARIS Express. Ta verzija je brez časovne omejitve, vendar je precej okrnjena v primerjavi z verzijo Professional ARIS Platform. Podpora za simulacijo in optimizacijo je na voljo le v verziji Professional ARIS Platform.

3.4 ProcessMaker

ProcessMaker je odprtokodno orodje za upravljanje poslovnih procesov. Orodje je razvila organizacija ProcessMaker Inc. in je bilo prvič izdano leta 2008. Z njim si lahko organizacije pomagajo pri načrtovanju in avtomatizaciji poslovnih procesov. Pomembne zmogljivosti, ki jih premore ProcessMaker, so ustvarjanje digitalnih oblik in popolno upravljanje potekov dela. Programska oprema je popolnoma spletna in dostopna preko poljubnega spletnega brskalnika, zaradi česar je enostavno upravljanje in usklajevanje poteka dela v celotni organizaciji - vključno z uporabniškimi skupinami in oddelki (Abdelgeer, F. M. Z.; Bischoff & Van Dinther, 2016; Kahkesh, 2012).

3.4.1 Modeliranje in simulacija

ProcessMaker Designer omogoča izdelavo procesov, ki so skladni z BPMN 2.0 in omogočajo standardno notacijo. ProcessMaker omogoča le modeliranje diagramov poslovnih procesov, ne podpira pa tudi simulacije. Pri povezovanju dveh elementov v procesu je naslednji razpoložljiv element obarvan z zeleno. Medtem so povezave, ki niso veljavne med elementi, obarvane z rdečo. Pri povezovanju dveh elementov, med katerima je povezava neveljavna, na primer med začetnim dogodkom in končnim dogodkom, se bo element obarval rdeče, kar onemogoča povezavo. Možen tip notacije je BPMN.

3.5 Adonis

Adonis je orodje za modeliranje poslovnih procesov, ki organizatorjem, domenskim strokovnjakom

in poslovnim analitikom pomaga izboljšati učinkovitost podjetja in zagotavlja zmogljive vmesnike za izvajanje IT. Orodje je del BOC Management Office in je bilo prvič izdano leta 1995. Omogoča naslednje funkcionalnosti (BOC Group, n.d.):

- ustvarjanje opisov in navodil procesa,
- ocenjevanje in izvajanje različnih scenarijev, kot so vodenje kakovosti, vodenje KPI, upravljanje uspešnosti ali obvladovanje tveganj,
- analiziranje in načrtovanje potreb po človeških virih,
- določevanje zahtevanih dokumentov za aplikacije, ki izvajajo poslovne procese,
- izvajanje poslovne analize za vsakodnevne poslovne procese, ki temeljijo na času, virih in stroških.

Omogoča opisovanje ključnih elementov podjetja in prikazuje, kako procesna organizacija, organizacijske strukture, izdelki, različice izdelkov in podporne informacijske tehnologije vplivajo drug na drugega. Orodje Adonis prav tako podpira osnovne dejavnosti metodologij upravljanja poslovnih procesov, vključno z zbiranjem informacij, modeliranjem in načrtovanjem, analizo, simulacijo ter vrednotenjem. Uporabljena je bila brezplačna verzija programa Adonis: Community Edition (Adonis, n.d.), ki je brez časovne omejitve, vendar je precej okrnjena v primerjavi s komercialnimi različicami. Podpora za validacijo modela je na voljo le v Commercial Edition. Podpora za simulacijo je na voljo le v Professional Edition. V sklopu raziskave je bila izbrana verzija Adonis Community Edition, kar pomeni, da ni bilo podpore za validacijo in simulacijo modela (Møller et al., 2007).

3.5.1 Modeliranje in simulacija

Omogoča modeliranje več tipov modelov: diagram primerov uporabe, diagram poslovnih procesov, podjetniški diagram procesov, diagram dokumentov, diagram IT sistema, diagram produktov, diagram delovnega okolja. Pri tem podpira več notacij: BPMS, BPMN, UML, EPC in LOVEM.

3.5.2 Prenos iz BPMN v BPEL

Iz razvitih modelov BPMN lahko Adonis ustvari jezik za izvajanje poslovnih procesov (BPEL) in WSDL, ki pomagata pri prehodu iz faze poslovne analize v fazo izvajanja.

4 RAZISKAVA

Cilj raziskave je bil ugotoviti, ali obstoječa orodja dejansko nudijo ustrezno podporo optimizaciji, in ovrednotiti, katera izmed njih nudijo več. Med osnovno podporo vključujemo možnosti optimizacije, simulacije, validacije, nadzora in prenosa iz BPMN v BPEL. Izvedli smo dve empirični raziskovalni metodi: laboratorijski eksperiment in anketo (Čuček, 2018). V sklopu eksperimenta smo izdelali preizkus, ki je vključeval ocenjevanje, učenje, analiziranje in preverjanje v izbranih orodjih. Analiziral in primerjal je pet različnih orodij (Enterprise Architect, Bizagi Modeler, Aris Express, ProcessMaker in Adonis). Izvedba primerjave je vključevala modeliranje istega procesa v vseh orodjih. Pričeli smo z orodjem Adonis in primarno ugotavljali, možnosti modeliranja. Nato se je enak proces zmodeliral še v ostalih orodjih. S pripravo modelov smo pridobili podatke o vrsti začetnih, vmesnih in končnih dogodkih ter prehodih, ki jih posamezno orodje ponuja. Vsak model smo tudi v posameznih orodjih validirali ter primerjali rezultate. Simulacija se je izvedla v primeru, če smo s pregledom literature in specifikacij ugotovili, da po-

samezno orodje to vrsto koncepta podpira. Podobno smo izvedli tudi prenos procesa iz BPMN v BPEL. Ob koncu analize vsakega orodja smo ugotovitve analizirali. Izbrali smo kriterije, ki se jih najpogosteje uporablja pri prenovi poslovnih procesov, torej: optimizacija, simulacija, validacija, nadzor in prenos iz BPMN v BPEL. Najboljšo možno oceno analize bi doseglo orodje, ki bi omogočalo vse izbrane kriterije.

Iz tabele (Tabela 1) je razvidno, da imata podporo za simulacijo le orodji Enterprise Architect in Bizagi Modeler. Orodji Adonis in ARIS imata plačljivo različico, ki ponuja podporo za simulacijo (Adonis Professional Edition in Professional ARIS Platform). Ugotovili smo, da imajo podporo za validacijo vsa orodja, razen Adonis Community Edition, ki ima to podporo v različici Commercial Edition. Prenos iz BPMN v BPEL podpirata le orodji Adonis in Enterprise Architect. Sledenje optimizaciji oziroma možnosti nadzora (nadzorne plošče) ne podpira nobeno izmed analiziranih orodij.

V tabeli (Tabela 2) so prikazane možnosti modeliranja posameznih analiziranih orodij. Diagram poslovnih procesov je edini izmed diagramov, ki ga podpirajo vsa analizirana orodja.

Tabela 1: Podpora v analiziranih orodjih.

Podpora/Orodje	Adonis	Aris	Enterprise Architect	Process Maker Designer	Bizagi Modeler
Simulacija	x	x	✓	x	✓
Validacija	x	✓	✓	✓	✓
Sledenje optimizaciji	x	x	x	x	x
BPMN v BPEL	✓	x	✓	x	x

Tabela 3 prikazuje tipe notacij, ki jih podpirajo analizirana orodja. BPMN je edina notacija, ki jo podpira-

jo vsa orodja. ProcessMaker Designer in Bizagi Modeler sta orodji, ki podpirata zgolj eno (BPMN) notacijo.

Tabela 2: Možnosti modeliranja v analiziranih orodjih

Modeliranje/Orodje	Adonis	Aris	Enterprise Architect	Process Maker Designer	Bizagi Modeler
Organizacijska shema	x	✓	x	x	x
Arhitektura procesa	x	✓	x	x	x
Podjetniški diagram procesov	✓	x	x	x	x
Diagram dokumentov	✓	x	x	x	x
Diagram IT sistema	✓	✓	x	x	x
Diagram produktov	✓	x	x	x	x

Modeliranje/Orodje	Adonis	Aris	Enterprise Architect	Process Maker Designer	Bizagi Modeler
Diagram delovnega okolja	✓	✗	✗	✗	✗
Diagram poslovnih procesov	✓	✓	✓	✓	✓
Diagram primerov uporabe	✓	✗	✓	✗	✗
Poslovni proces	✗	✓	✗	✗	✗
Podatkovni model	✗	✓	✗	✗	✗
Diagram sistemske pokrajine	✗	✓	✗	✗	✗
Diagram razredov	✗	✗	✓	✗	✗
Diagram aktivnosti	✗	✗	✓	✗	✗
Diagram zaporedij	✗	✗	✓	✗	✗
Časovni diagram	✗	✗	✓	✗	✗
Diagram pregledov interakcij	✗	✗	✓	✗	✗
Komunikacijski diagram	✗	✗	✓	✗	✗
Diagram stanj naprav	✗	✗	✓	✗	✗

Raziskavo smo razširili s pomočjo dodatne empirične raziskovalne metode (anketa v prilogi), v kateri smo želeli ugotoviti poznavanje koncepta optimizacije ter podporna orodja v slovenskih podjetjih. Anke-to je izpolnilo 31 anketirancev iz 24 različnih podjetij (Čuček, 2018).

Tabela 3: Možni tipi notacij v analiziranih orodjih

Notacija/Orodje	Adonis	Aris	Enterprise Architect	Process Maker Designer	Bizagi Modeler
BPMN	✓	✓	✓	✓	✓
BPMS	✓	✗	✗	✗	✗
UML	✓	✗	✓	✗	✗
EPC	✓	✓	✗	✗	✗
LOVEM	✓	✗	✗	✗	✗
ERM	✗	✓	✗	✗	✗
SysML	✗	✗	✓	✗	✗

4.1 Vzorec

- **Populacija:** Za anketo so bili potencialni kandidati osebe, ki so zaposlene na oddelkih informatike v slovenskih podjetjih. Starostno omejena je le spodnja meja – predvideno je namreč, da so osebe, ki so zaposlene na oddelkih informatike, stare vsaj 18 let. Številčnost populacije se je gibala med 20 in 30 podjetji.

- **Metoda vzorčenja:** Priložnostno vzorčenje – osebe, ki smo jih poznali oziroma tiste, ki so privolile v sodelovanje.
- **Vzorec:** Obseg vzorca je bil med 20 in 30 podjetji. Osebe, ki so sodelovale pri anketi, so bile v času ankete stare vsaj 18 let in so zaposlene na oddelkih informatike v slovenskih podjetjih.

Anketo smo sestavili na podlagi v uvodu zastavljenih raziskovalnih vprašanj. S pomočjo ankete smo dobili odgovor na raziskovalno vprašanje: kakšno je poznavanje orodij za optimizacijo poslovnih procesov in kolikšna je njihova uporaba na oddelkih informatike v slovenskih podjetjih (glej prilogo).

4.2 Rezultati

Anketirani so na trditve odgovarjali s 5 stopenjsko intervalno mersko lestvico (sploh se ne strinjam, se ne strinjam, se ne morem odločiti, se strinjam, se popolnoma strinjam). V nadaljevanju so predstavljena ključna vprašanja skupaj z analiziranimi odgovori (izpostavljeni so prevladujoči odgovori):

- **Ali imate v podjetju dobro dokumentirane ključne poslovne procese:** 22 % anketiranih se s to trditvijo sploh ne strinja, 19 % se s trditvijo ne strinja, 33 % anketiranih se ni moglo opredeliti glede trditve, 19% se s trditvijo strinja in 7 % anketiranih se a trditvijo popolnoma strinja.
- **Ali ste v podjetju seznanjeni s pojmom »Optimizacija poslovnih procesov«:** 16% anketiranih se

s to trditvijo sploh ne strinja, 16% anketiranih se s trditvijo ne strinja, 26% anketiranih se ni moglo odločiti, 33% anketiranih se s trditvijo strinja, 9% anketiranih se s to trditvijo popolnoma strinja.

- **V podjetju smo organizirali eno ali več predavanj na temo optimizacije poslovnih procesov:** 53% anketiranih se s trditvijo sploh ne strinja, 16% anketiranih se ne strinja, 22% anketiranih se ni moglo odločiti in 9% anketirani se strinja. Noben anketiranec se ni popolnoma strinjal s trditvijo.
- **Naše podjetje optimizira poslovne procese:** 9% se s trditvijo sploh ne strinjajo, 26% se ne strinja, 32% izmed anketiranih se ni moglo odločiti, 26% anketiranih se s to trditvijo strinja, 7% anketirana se s trditvijo popolnoma strinja.
- **Optimizaciji poslovnih procesov posvečamo veliko pozornosti:** 35% anketiranih se s trditvijo sploh ne strinja, 13% se s trditvijo ne strinja, 29% anketiranih se ni moglo odločiti, 16% se strinja, 7% se popolnoma strinja.
- **Z optimizacijo poslovnih procesov smo izboljšali poslovne rezultate (KPI):** 29% anketiranih se s to trditvijo sploh ne strinja, 13% se ne strinja, 19% anketiranih se ni moglo odločiti, 26% se jih strinja in 13% se popolnoma strinja.
- **Označite orodja, za katera ste že slišali (možnih več izbir):** največ anketiranih je izbralo orodje Visio (ki ne nudi optimizacijske podpore), natančneje 58%. 42% anketiranih je slišalo za orodje Signavio (s fakultete), 16% anketiranih pa je izbralo orodji ProcessMaker in Enterprise Architect. Orodji Bizagi Modeler in Aris je izbralo po 7% anketiranih, orodja Adonis, IBM Business Process Manager in Glify je izbralo 4% anketirancev. 23% anketiranih ni slišalo za nobeno od orodij za optimizacijo poslovnih procesov.
- **Ali ste že kdaj uporabljali katero izmed spodaj naštetih orodij (možnih več izbir):** 32% anketiranih je uporabljalo orodje Signavio, 16% je izbralo orodje Visio, 7% anketiranih pa je izbralo orodje Bizagi Modeler. Orodja Aris, IBM Business Process Manager, Glify, Enterprise Architect, ProcessMaker je izbralo po 3% anketirancev. 23% anketiranih ni uporabljalo nobenega orodja za optimizacijo poslovnih procesov. Prav tako orodje Adonis ni uporabljal noben od anketiranih.
- **Katere koncepte ste uporabljali v orodju:** 48% anketiranih je izpostavilo, da je že sodelovalo pri optimizaciji, 45% jih je orodje uporabljalo za mo-

deliranje, 42% za simulacijo, 13% za sledenje optimizaciji in 10% za prenos iz BPMN v BPEL.

Splošni podatki anketirancev (spol, starost, izobrazba, podjetje) niso vključeni med analizo rezultatov. Dodatno smo izpostavili naslednje karakteristike zaposlenih: **delovno dobo, seznanjenost z optimizacijo, poznavanje orodij za optimizacijo ter zavedanje pomembnosti optimizacijskih konceptov.** Delovna doba anketiranih je pomembna, saj je večja možnost, da je dlje zaposleni že slišal za optimizacijo poslovnih procesov, kot nekdo, ki je zaposlen leto ali manj. Prevladujejo osebe, zaposlene eno leto.

18 ZAKLJUČEK

S pregledom literature smo identificirali koncepte optimizacije poslovnih procesov (modeliranje, validacija, simulacija, sledenje optimizaciji in prenos iz BPMN v BPEL). Izbrane koncepte smo preizkusili v različnih orodjih, ki nudijo podporo optimizaciji poslovnih procesov (oziroma so na ta način predstavljeni). V okviru empirične raziskave laboratorijskega eksperimenta smo izdelali model enega procesa v vseh analiziranih orodjih ter rezultate medsebojno primerjali. Orodje Enterprise Architect smo označili kot najboljše, saj ponuja največjo podporo izmed vseh analiziranih orodij. Drugo najboljše orodje smo izbrali Bizagi Modeler. Ne glede na to, da v sklopu modeliranja omogoča le modeliranje diagramov poslovnih procesov, je zaradi podpore simulacije omogočal več od orodja Aris. Kot tretje je bilo označeno orodje Aris, ki ne omogoča simulacije in sledenja optimizaciji, vendar ima na voljo vse dogodke in prehode. Orodje ProcessMaker je bilo označeno kot četrto zaradi omejitev pri modeliranju. Najslabše je bilo označeno orodje Adonis Community Edition, saj je modeliranje procesov, brez vmesnih dogodkov, tako rekoč nemogoče.

S pomočjo ankete smo pridobili informacije o tem, kakšno je poznavanje orodij za optimizacijo poslovnih procesov in njihova uporaba na oddelkih informatike v slovenskih podjetjih. V splošnem lahko na podlagi ankete poznavanje orodij za optimizacijo poslovnih procesov v slovenskih podjetjih označimo kot pomanjkljivo oziroma je le-ti ne integrirajo v proces prenove. Več anketirancev ni poznalo nobenega od naštetih orodij, tudi sami niso dodali orodja, ki ga ni bilo med naštetimi. Na podlagi odgovorov iz ankete lahko sklepamo, da slovenska podjetja poslovnih procesov načeloma ne optimizirajo s pomočjo

podpornih orodij, kjer predhodno s pomočjo simulacije raziščejo možnosti za spremembe. S pojmom so sicer seznanjeni, vendar ga v praksi ne izvršujejo na sistematični način, kljub nekaterim pozitivnim izkušnjam glede izboljšave svojih poslovnih rezultatov.

V sklopu raziskave smo naleteli na več omejitev. Odzivnost sodelujočih je bila nizka, čeprav smo vabilo poslali več kot 100 podjetjem. Odzvalo se je le štiriindvajset podjetij, anketo pa je izpolnilo 31 posameznikov. Glede na majhno število odgovorov je rezultate težko posplošiti na populacijo vseh slovenskih podjetij. Zato previdneje sklepamo o vzorcih, ki bi lahko veljali za širšo populacijo. Raziskava je ponovljiva in se lahko ponovno izvede v drugačnem oziroma razširjenem okolju.

Na omejitve smo prav tako naleteli pri samem modeliranju poslovnega procesa, saj se je izkazalo, da vsa orodja ne podpirajo vseh gradnikov, ki smo jih potrebovali za izbran poslovni proces. Na te omejitve smo naleteli pri orodju Adonis Community Edition in ProcessMaker. Adonis ne omogoča vmesnih dogodkov, medtem ko ProcessMaker ne omogoča ekskluzivnega prehoda, ki bi podpiral več kot dve medsebojno izključni poti.

Dodatno smo naleteli na omejitve pri uporabi izbranih orodij, ki so bila plačljiva, prosto dostopne verzije pa so imele več omejitev. Enterprise Architect omogoča le trideset dnevno brezplačno uporabo orodja. Primerjavo orodij je izvedla ena oseba, kar v raziskavo vpeljuje ogroženost interne validacije.

5.1 Diskusija

V obstoječi literaturi nismo zasledili podobne analize, ki bi se fokusirala na analizo orodij za optimizacijo poslovnih procesov, in bi bila v pomoč tako v industriji kot v akademskem svetu. Rezultati raziskave so uporabni v industriji, saj iz naše raziskave izvedo, kaj katero orodje omogoča in kaj bi lahko uporabili za svoje specifične procese. Prav tako so rezultati uporabni za izobraževalne namene, predvsem priprave gradiva, dobrih praks in prednosti optimizacije poslovnih procesov.

S pomočjo anketnih rezultatov smo spoznali zrelost podjetij in pomanjkljivosti seznanjenosti zaposlenih s postopkom optimizacije. Podjetjem bi lahko izpostavili potrebo po digitalizaciji in ocenjevanje pripravljenosti na le-to.

V prihodnosti bomo v razširjeni obliki ponovili anketo, z namenom iskanj specifičnih potreb in zah-

tev slovenskih podjetij ter njihovih poslovnih strategij na področju optimizacije poslovnih procesov, ter dodatno pripravili priporočila za izbiro najprimernejšega orodja.

VIRI IN LITERATURA

- [1] Abdelgeer, F. M. Z., Dawood O. O. S., Mustafa M. M. E. (2013). Comparison of Workflow Management Systems Bizagi, ProcessMaker, and Joget. *The International Arab Conference on Information Technology*, (Interface 4), 4–8
- [2] Adonis. Adonis: Community Edition. Retrieved from <https://www.adonis-community.com/en/support/>
- [3] Bischoff, B. & Van Dinther, C. (2016). Workflow Management Systems an analysis of current open source products.
- [4] Bizagi (2014). BPMN by Example, 1–24.
- [5] BOC Group (n. d.). The Business Process Management Toolkit.
- [6] Bork, D., & Fill, H. G. (2014). Formal aspects of enterprise modeling methods: A comparison framework. *Proceedings of the Annual Hawaii International Conference on System Sciences*, (Dd), 3400–3409. <https://doi.org/10.1109/HICSS.2014.422>
- [7] Aris Community, Aris Express. Retrieved from <https://www.ariscommunity.com/aris-express/details>.
- [8] ČUČEK, Tamara, 2018, Analiza orodij za podporo optimizacij procesov, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko. Retrieved from: <https://dk.um.si/IzpisGradiva.php?lang=slv&id=72198>
- [9] Delgado, A., Alegari, D., & Arrigoni, A. (2016). Towards a Generic BPMS User Portal Definition for the Execution of Business Processes. *Electronic Notes in Theoretical Computer Science*, 329, 39–59. <https://doi.org/10.1016/j.entcs.2016.12.004>
- [10] DeMast, Jeroen, and Joran Lokkerbol. 2012. "An Analysis of the Six Sigma DMAIC Method from the Perspective of Problem Solving." *International Journal of Production Economics* 139(2): 604–14. <https://linkinghub.elsevier.com/retrieve/pii/S0925527312002277>.
- [11] Dumas, M., La Rosa, M., Mendling, J., Reijers, H. (2013). Fundamentals of Business Process Management.
- [12] Kahkesh, S. (2012). Dynamic business process management.
- [13] Ko, Ryan K.L., Stephen S.G. Lee, and Eng Wah Lee. 2009. "Business Process Management (BPM) Standards: A Survey." *Business Process Management Journal* 15(5): 744–91. <http://www.emeraldinsight.com/doi/10.1108/14637150910987937>.
- [14] Bizagi Modeler. (2016). Bizagi Process Modeler User Guide Welcome to Bizagi Modeler
- [15] Møller, C., Maack, C. J., & Tan, R. D. (2007). What is Business Process Management : A Two Stage Literature Review of an Emerging Field. *Research and Practical Issues of Enterprise Information Systems II*, 19–31. https://doi.org/10.1007/978-0-387-75902-9_3
- [16] Rosemann, M., & Brocke, J. (2010). Handbook on Business Process Management 1, 107–122. <https://doi.org/10.1007/978-3-642-00416-2>.
- [17] Sparks, G. & Maxwell, B. (2010). Project Management with Enterprise Architect Managing Editor. *Changes*. Retrieved from http://www.sparxsystems.com.au/downloads/resources/booklets/project_management_with_enterprise_architect.pdf
- [18] Sparx Systems (2010). Enterprise Architect Reviewer's Guide, Retrieved from <https://sparxsystems.com/downloads/whitepapers/EAReviewersGuide-28052013.pdf>

- [19] van der Aalst, W. M. P., ter Hofstede, A. H. M., & Weske, M. (2003). Business Process Management: A Survey, 1–12. https://doi.org/10.1007/3-540-44895-0_1.
- [20] Ward-Dutton, N. (2010). Vendor Insight, 1–14
- [21] Witts, I. (2016). 5 Reasons your Business needs Business Process Management. Retrieved from <http://blog.triaster.co.uk/blog/reasons-your-business-needs-business-process-management>

■

Tamara Čuček: Razvijalec rešitev v podjetju Nomnio d.o.o. na področju razvoja programske opreme. Končala 1. in 2. stopnjo na študijskem programu Informatika in tehnologije komuniciranja na Fakulteti za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru. Glavna strokovna področja: razvoj namiznih aplikacij in zalednih sistemov v programskem jeziku C#.

■

Boštjan Šumak: Docent na Fakulteti za elektrotehniko, računalništvo in informatiko (Univerza v Mariboru) od 2012. Leta 2011 prejel doktor znanosti s področja računalništva in informatike na Univerzi v Mariboru. Glavna strokovna področja: sodobne arhitekture, spletne tehnologije, uporabniška izkušnja, sprejetost in uporaba informacijskih tehnologij in sistemov, tehnologije za prostoročno upravljanje naprav, integracija informacijskih sistemov ter XML in povezane tehnologije.

■

Maja Pušnik: Docent na Fakulteti za elektrotehniko, računalništvo in informatiko (Univerza v Mariboru), od 2016. Leta 2014 prejela doktor znanosti s področja računalništva in informatike na Univerzi v Mariboru. Glavna strokovna področja: XML in povezane tehnologije, vrednotenje kakovosti procesov in programske opreme, optimizacija poslovnih procesov, orkestracija poslovnih procesov, operacijske raziskave in teorije odločanja.

PRILOGA: ANKETA

Splošne informacije

- Spol:
 - Moški
 - Ženska
- Starost:
 - 18-24
 - 25-34
 - 35-49
 - 50 ali več
- Izobrazba:
 - I. (nedokončana OŠ)
 - II. (OŠ)
 - III. (nižje poklicno izobraževanje (2-letno))
 - IV. (srednje poklicno izobraževanje (3-letno))
 - V. (gimnazijsko, srednje poklicno -tehniško izobraževanje, srednje tehniško oz. drugo strokovno izobraževanje)
 - VI./1. (višješolski program (do 1994), višješolski strokovni program)
 - VI./2. (visokošolski strokovni in univerzitetni program (1. bol. st.))
 - VII. (magisterij stroke (2. bol. st.))
 - VIII./1. (specializacija po univerzitetnem programu, magisterij znanosti)
 - VIII./2. (doktorat znanosti (3. bol. st.))
- Podjetje: _____
- Število let delovne dobe: _____

Poznavanje pojma »Optimizacija poslovnih procesov«

- Ali ste seznanjeni s pojmom »Optimizacija poslovnih procesov«?
 - Da
 - Ne
- Kje ste se srečali s pojmom »Optimizacija poslovnih procesov«?
 - Na fakulteti
 - V službi
 - Drugo: _____
- Ali ste že kdaj sodelovali pri optimizaciji poslovnih procesov?
 - Da
 - Ne
- Označite orodja za katera ste že slišali.
 - Bizagi Modeler
 - ProcessMaker
 - Enterprise Architect
 - Aris
 - Adonis

- Signavio
 - Visio
 - Za nobeno orodje nisem slišal/a
 - Drugo: _____
- Ali ste že kdaj uporabljali katero izmed spodaj naštetih orodij (če boste označili »Drugo«, vas prosim, da orodje, ki ste ga uporabljali, zapišete)?
- Bizagi Modeler
 - ProcessMaker
 - Enterprise Architect
 - Aris
 - Adonis
 - Signavio
 - Visio
 - Nisem uporabljal/a
 - Drugo: _____

Poznavanje orodij

Če ste pri prejšnjem vprašanju obkrožili katerega izmed orodij ali zapisali svojega, prosim nadaljujte.

- Kje ste se srečali z orodjem?
- Na fakulteti
 - V službi
 - Zasebno
 - Drugo: _____
- Za kaj ste uporabljali orodje?
- Sodeloval sem pri optimizaciji poslovnega procesa
 - Z orodjem sem se želel le seznaniti
 - Drugo: _____
- Katere koncepte ste uporabljali v orodju?
- Modeliranje
 - Optimizacija
 - Simulacija
 - Sledenje optimizaciji
 - Prenos iz BPMN v BPEL
 - Drugo: _____
- Prosim preberite spodnje trditve in označite v kolikšni meri se z njimi strinjate, pri čemer 1 pomeni »sploh se ne strinjam« in 5 »popolnoma se strinjam«.
- V podjetju imamo dobro dokumentirane ključne poslovne procese.
 - V podjetju smo seznanjeni s pojmom »Optimizacija poslovnih procesov«.
 - V podjetju so organizirali eno ali več predavanj na temo optimizacije poslovnih procesov.
 - Naše podjetje optimizira poslovne procese.
 - Optimizaciji poslovnih procesov posvečamo veliko pozornosti.
 - Z optimizacijo poslovnih procesov smo izboljšali poslovne rezultate (KPI).

■ Razpoznavanje slovenskega govora z metodami globokih nevronske mreže

Matej Ulčar¹, Simon Dobrišek², Marko Robnik-Šikonja¹

¹Univerza v Ljubljani, Fakulteta za računalništvo in informatiko,

Večna pot 113, 1000 Ljubljana

matej.ulcar@fri.uni-lj.si marko.robnik@fri.uni-lj.si

²Univerza v Ljubljani, Fakulteta za elektrotehniko,

Tržaška 25, 1000 Ljubljana

simon.dobrisek@fe.uni-lj.si

Izveček

V zadnjem času se na področju samodejnega razpoznavanja govora uveljavljajo globoke nevronske mreže, ki nadomeščajo akustično modeliranje z uporabo modelov HMM in GMM ter n-grame za jezikovni model. Za razpoznavanje govorne slovenščine smo izdelali in preizkusili več arhitektur časovno zakasnenih nevronske mreže in nevronske mreže z dolgim kratkoročnim spominom na akustičnem in jezikovnem modelu v sistemu Kaldi. Razpoznavnik smo učili na obširnem besednjaku, ki vsebuje približno milijon različnih besed. Najboljše rezultate smo dosegli s časovno zakasnenimi nevronske mreže, kjer smo dosegli 27,16 % napako po kriteriju WER. Preliminarni rezultati kažejo boljšo natančnost v primerjavi z Googlovim modelom »speech-to-text«, vendar pa je za zanesljivo primerjavo potrebno več dodatnega testiranja.

Ključne besede: strojno učenje, globoke nevronske mreže, razpoznavanje govora, govorne tehnologije, obdelava naravnega jezika

Abstract

Recently, deep neural networks have become the predominant approach to automatic speech recognition, replacing classical acoustical modelling using GMM and HMM models and n-grams for the language model. For the recognition of spoken Slovene, we have developed and tested several architectures of time-delayed neural networks and neural networks with a long short-term memory for both acoustic and language models in the Kaldi environment. We used a large lexicon containing about a million words. Time-delayed neural networks achieved the best results on continuous speech, with a 27.16% error according to the WER criterion. Preliminary results show better performance compared to Google's speech-to-text model. However, more testing is needed for a statistically valid comparison.

Keywords: Machine learning, deep neural networks, speech recognition, speech technologies, natural language processing

1 UVOD

Govor večkrat želimo zapisati kot besedilo, na primer zapisnik sestanka, zapiske s predavanj, podnapiše na televiziji v pomoč slušno prizadetim, ipd. Za zapis je potrebno govor večkrat poslušati in ga sproti zapisovati, kar je lahko časovno potratno, še posebej, ko je govor hiter in je potrebno posnetek ustavljati in ponovno predvajati.

Problem razpoznavanja govora je sestavljen iz dveh delov: akustičnega modeliranja in jezikovnega modeliranja. Akustično modeliranje se nanaša na gradnjo modelov posameznih glasov oziroma fonemov za dani govorni jezik. Fonem je osnovna enota glasu, ki razločuje pomen besed. Pri akustičnem mo-

deliranju namesto fonemov pogosto uporabljamo trifone, to je, po tri foneme združene skupaj v eno enoto. Jezikovno modeliranje pa se nanaša na modeliranje preslikave fonemov v besede in nizanja besed v besedila. Za reševanje problemov obdelave naravnih jezikov se v zadnjem času zelo uspešno uporabljajo globoke nevronske mreže (DNN – Deep Neural Networks). Z uporabo globokih nevronske mreže smo poizkusili izboljšati rezultate do sedaj najuspešnejših metod strojnega učenja pri razpoznavanju govora v slovenščini in izdelati kakovostno odprtokodno rešitev.

Članek sestavlja šest razdelkov. V drugem na kratko opišemo sorodna dela. V tretjem razdelku

opišemo uporabljene vire. V četrtem razdelku opišemo uporabljene tehnologije in predstavimo arhitekturo razpoznavalnika, oziroma različne postopke učenja, ki smo jih uporabili. V petem razdelku sledi predstavitev in analiza rezultatov. V sklepnem delu opišemo opravljeno delo ter predstavimo možnosti za izboljšave.

2 PREGLED SORODNIH DEL

Razpoznavanje govora je najbolj razvito za angleški jezik, kjer so trenutno najuspešnejši modeli, naučeni z uporabo globokih nevronske mreže. Microsoft v svojem sistemu za razpoznavanje pogovornega govora (Xiong in sod., 2017) uporablja globoke nevronske mreže za akustični in jezikovni model. Akustični model je naučen s kombinacijo konvolucijskih nevronske mreže in nevronske mreže z dolgim kratkoročnim spominom (angl. long short-term memory, LSTM). Za učenje jezikovnega modela so uporabili rekurenčne nevronske mreže (angl. recurrent neural network, RNN).

Googlova aplikacija za pametne telefone, ki uporablja glasovno upravljanje ter omogoča glasovno iskanje, uporablja za razpoznavanje nevronske mreže pri akustičnem modelu (Sak, Senior, Rao, Beaufays, Schalkwyk, 2015). Sak, Senior, Rao in Beaufays (2015) so uporabili dvosmerne globoke nevronske mreže LSTM. Za poravnavo zvočnega posnetka s transkripcijo v učni množici so namesto modela GMM-HMM (GMM – Gaussian Mixture Model - mešanica Gaussovih porazdelitev, HMM – Hidden Markov Model - prikriti Markovov model) uporabili metodo povezovalne časovne klasifikacije (angl. Connectionist Temporal Classification – CTC).

Hernandez in sod. (2018) so uporabili zbirko orodij Kaldi za učenje sistema za razpoznavanje govora v angleščini. Za govorno zbirko so uporabili zbirko predavanj TED. Akustični model so naučili s hibridnim modelom (DNN-HMM), kjer so najprej uporabili pristop GMM-HMM za učenje in poravnavo zvočnih posnetkov s transkripcijo. Nato so namesto GMM uporabili časovno zakasnjene nevronske mreže (angl. time delayed neural network, TDNN). Naučili so dva jezikovna modela. Prvega z n-grami reda 4 in drugega z uporabo nevronske mreže, kjer so uporabili tri nivoje TDNN, med njimi pa dva nivoja LSTM.

Bolka (2016) je v diplomskem delu uporabil zbirko orodij Kaldi za razpoznavanje fonemov v slo-

venščini. Uporabil je več metod učenja akustičnega modela; model zgrajen z nevronske mreže je dosegel najboljše rezultate. Nevronske mreže LSTM za prevajanje med fonemi in grafemi (tekstovnimi zapisi fonemov) predlagajo tudi Rao, Peng, Sak in Beaufays (2015). V svojem delu izdelajo model, ki napoveduje foneme glede na dane grafeme, z drugimi besedami napovedujejo izgovor besede. Uporabili so tako plitke kot globoke nevronske mreže LSTM. V našem delu obravnavamo obraten problem, kjer določamo zapis besede glede na njen izgovor. Globoke nevronske mreže LSTM vsebujejo posebne enote, imenovane spominske celice. Te so si zmožne podatke zapomniti poljubno dolgo. Pozabna vrata spominske celice skrbijo, da se podatek lahko po potrebi tudi pozabi. Zaradi teh lastnosti so globoke nevronske mreže LSTM zelo dobre pri prepoznavanju govora, saj pri učenju upoštevajo tudi kontekst (zapis besede je odvisen tudi od predhodnih glasov, ne samo od trenutnega) (Bolka, 2016; Rao in sod., 2015).

Jezikovni modeli pri razpoznavanju govora v slovenščini večinoma uporabljajo Good-Turingovo glajenje (Žgank, Donaj, Sepesy Maučec, 2014; Žgank, Verdonik, Sepesy Maučec, 2016; Donaj, 2015). Žgank in sod. (2014) so uporabili dve govorni bazi, eno z večjim deležem spontanega govora, drugo z večjim deležem branega govora. Akustični model so osnovali na zveznih prikritih Markovovih modelih. Žgank in sod. (2014) so ugotavljali predvsem vpliv velikosti uporabljenih besedilnih in govornih korpusov. Njihova analiza je pokazala, da večanje uporabljenih virov izboljša rezultate, vendar je to izboljšanje majhno, za večjo izboljšavo je potrebna tudi uporaba drugih algoritmov.

Žgank in sod. (2016) so uporabili enak razpoznavalnik kot prej opisani (Žgank in sod., 2014) za transkribiranje nove govorne baze SI TEDx-UM. Uporabili so dva jezikovna modela, enega grajenega na govorni bazi BNSI, drugega pa zgolj na besedilnem korpusu FidaPLUS. Njihovi rezultati so pokazali, da je modeliranje govorne rabe jezika pomemben del jezikovnega modela, po drugi strani pa ima tematika govora velik vpliv na natančnost razpoznavanja besed. Oba jezikovna modela sta dosegla enak rezultat (napako 50,7 %). Ker se domeni govora pri SI TEDx-UM in BNSI med seboj precej razlikujeta, so rezultati pri razpoznavanju govora na predavanjih SI TEDx-UM precej slabši od rezultatov pri razpoznavanju govora posnetkov BNSI.

3 OPIS VIROV

Učenje dobrega razpoznavalnika govora zahteva mnogo učnih podatkov. Potrebujemo posnetke govora, njihove prepise, korpus in slovar besed z izgovorjavami. Za govorne posnetke smo uporabili korpus Gos 1.0 (Zwitter Vitez in sod., 2013), Gos VideoLectures 2.0 (od tu dalje uporabljamo oznako GosVL) (Verdonik in sod., 2017) in Sofes 1.0 (Dobrišek in sod., 2017). Lastnosti govornih korpusov so opisane v tabeli 1.

Tabela 1: Lastnosti govornih korpusov

Lastnost	Gos	GosVL	Sofes
Dolžina posnetkov	120 ur	9 ur 48 minut	9 ur 52 minut
Št. vseh govorcev	1526	44	134
Št. ženskih govork	681	17	32
Št. moških govorcev	845	27	102
Število stavkov	96334	4073	12536

Korpus Sofes ima nekatere stavke podvojene, vendar so ti posneti v različnih kvalitetah. Posnetke nizke kvalitete smo pri učenju izločili. Korpusa Gos in Sofes smo razdelili na učno in validacijsko množico z namenom, da na validacijski množici izberemo dobro delujoče parametre in se s tem izognemo pretiranemu prilagajanju učni množici. Posnetki iz obeh korpusov so prisotni tako v učni kot v validacijski množici. Razdelili smo ju tako, da se nihče izmed govorcev ne pojavi v obeh množicah, vedno le v eni. Korpus GosVL smo uporabili za končno testno množico.

Korpusa Gos in Sofes imata govor prepisan na dva načina. V prvem so stavki in besede zapisani v zbornem knjižnem jeziku, v drugem pa je zapis fonetičen. Ta dva zapisa smo želeli uporabiti za izdelavo slovarja izgovorjav. Korpus Sofes je sam premajhen, vsebuje premalo različnih besed. Pri obdelavi podatkov v korpusu Gos pa smo našli neujemanja med obema transkripcijama. Zapisa nimata vedno enakega števila besed, morda tudi ne vedno enakega vrstnega reda. Rezultati so bili neuporabni, zato smo se odločili, da uporabimo leksikon besednih oblik Sloleks (Dobrovoljc in sod., 2015). Prednost je, da vsebuje več besed, oziroma besednih oblik kot korpusa Gos in Sofes. Za naš namen pripravljen leksikon Sloleks vsebuje 1.129.141 različnih besednih oblik, korpus Gos pa le 83.000. Slabost uporabe Sloleksa je, da smo morali sami določiti fonetični zapis.

Besedilni korpus smo oblikovali na podlagi korpusa ccGigafida (Logar in sod., 2013). Korpus vsebuje približno 103 milijone besed. Iz korpusa ccGigafida smo izločili vse prazne vrstice, odstranili večkratne presledke, oziroma jih nadomestili z enojnimi ter odstranili vsa ločila. Tako je učenje jezikovnega modela lažje. V nasprotnem primeru model upošteva ločila kot del besede, teh besed pa ni v slovarju. S tem bi se povečalo število besed, ki bi jih morali upoštevati, korpus bi moral biti mnogo večji, učenje pa bi bilo zahtevnejše in počasnejše.

3.1 Obdelava podatkov

Govorni korpusi imajo transkripcijo zapisano v datotekah *.trs, ki so oblika formata XML. Za učenje z uporabo orodij Kaldi moramo najprej iz teh datotek izluščiti potrebne podatke. V ta namen smo napisali skripta, ki preberejo datoteke *.trs in vsakemu izreku pripišejo unikaten identifikator (ID), ID govorca, spol govorca ter datoteko zvočnega posnetka. Ti podatki se shranijo v različne datoteke. Datoteka *text* vsebuje v vsaki vrstici najprej identifikator izreka, nato sam izrek. Datoteka *utt2spk* vsebuje v vsaki vrstici identifikator izreka in identifikator govorca. Datoteka *spk2gender* vsebuje identifikator govorca in spol govorca (m za moški spol, f za ženski). Datoteka *wav.scp* vsebuje identifikator izreka in polno pot do zvočne datoteke. V določenih primerih je lahko v isti zvočni datoteki več izrekov. Vsi uporabljeni govorni korpusi imajo sicer zvočne posnetke razdeljene glede na posamezne izreke, vendar se pri korpusu GosVL ti ne ujemajo popolnoma. Število izrekov ni enako številu zvočnih posnetkov. Ročno ugotavljanje, kje pride do neujemanj, je zelo zamudno. Korpus GosVL ima zvočne posnetke razdeljene tudi na posamezna predavanja. Datoteke .trs vsebujejo podatke o časovni poziciji vsakega izreka znotraj zvočnega posnetka celotnega predavanja. Zato smo uporabili zvočne posnetke celotnih predavanj ter v datoteko *segments* za vsak izrek zapisali začetno in končno mesto (v sekundah) v zvočnem posnetku.

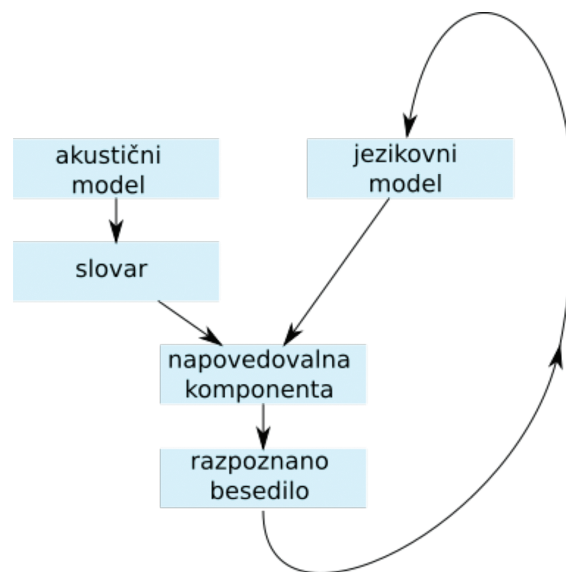
Korpus Gos ima pri nekaterih izrekih spol govorca označen kot »nedoločen«. V teh primerih smo najprej preverili ID govorca. Zadnja črka ID govorca namreč označuje spol govorca (»m« za moškega, »f« za žensko). V večini primerov, ko je spol označen kot nedoločen, je zadnja črka ID govorca »n«. Takrat smo se odločili, da govorcu pripišemo ženski spol »f«, ker so to večinoma posnetki otrok, ki so po višini glasu

bolj podobni ženskam, kot moškim. Nekateri izreki imajo več govorcev. Na primer, en govorec začne stavek, drugi ga dokonča. Ker ne vemo točno, kdaj govori kateri izmed govorcev, lahko pa se zgodi celo, da govorita hkrati, smo cel izrek pripisali enemu govorniku, ki je v transkripcijski datoteki zapisan prvi.

Sloleks vsebuje nekaj več kot 100.000 lem, oziroma skupaj 2.791.919 besednih oblik. Odstranili smo podvojene vnose, na primer samostalnik »miza« ima enako obliko v rodilniku ednine ter imenovalniku in tožilniku množine (»mize«). V Sloleksu bi to bili trije vnosi, potrebujemo pa le enega, saj ne uporabljamo podatkov o spolu, številu, sklonu in podobno. Končno imamo 1.129.141 različnih besednih oblik. Izgovarjave teh besed smo tvorili s pomočjo pravil zapisa in pravil izgovarjave v Slovenskem pravopisu. Dodali pa smo še nekaj svojih pravil, ki bolje opisujejo pogovorni jezik, ne zgolj zborni knjižni jezik. Primer takega pravila je izgovarjava končnice »-el« z glasom »-u« (ne »-ew«). Z znakom »w« smo tu označili vse oblike dvoustničnega u.

4 Arhitektura in učenje razpoznavalnika

Sistem za razpoznavanje govora lahko v grobem razdelimo na dva dela, akustični model in jezikovni model. Akustični model naučimo na značilkah pridobljenih iz zvočnih posnetkov govora in pripadajočih transkripcijah govora. Akustični model zvočnemu signalu pripiše pripadajoč fonem. Jezikovni model naučimo iz besedilnega korpusa. Ta model na podlagi predhodnih besed predlaga najbolj verjetno naslednjo besedo. Akustični model prek slovarja, s pomočjo katerega napovedanim fonemom pripišemo najverjetnejšo besedo, povežemo z jezikovnim modelom (slika 1). Napovedovalna komponenta vrača najbolj verjetno naslednjo besedo z uteženim povprečjem napovedi akustičnega in jezikovnega modela. Razpoznavalnik smo naučili z uporabo orodja Kaldi, ki je odprtokodna zbirka orodij za učenje razpoznavanja govora (Povey in sod., 2011).



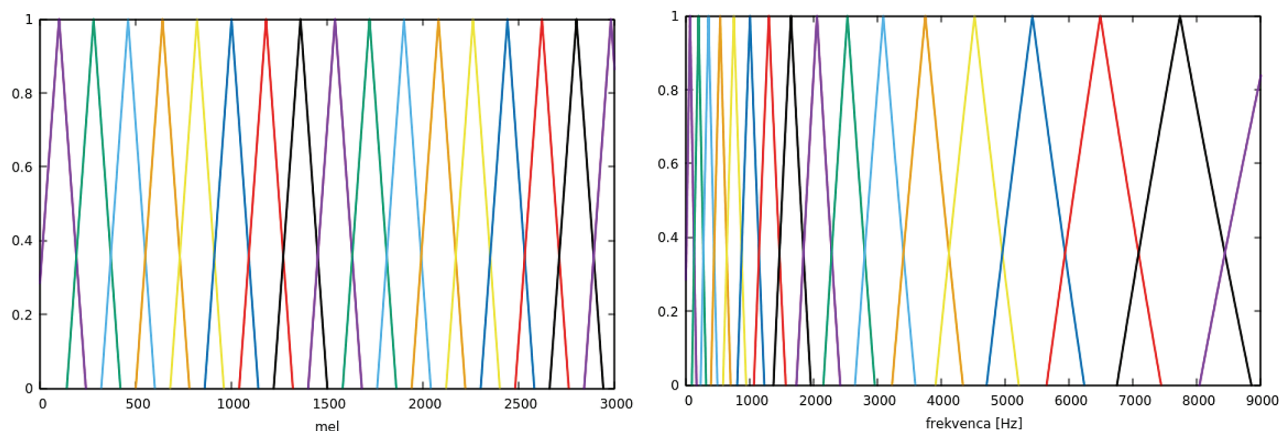
Slika 1: Groba shema posameznih komponent, oziroma modelov in povezav med njimi.

4.1 Akustični model

Učenje akustičnega modela razpoznavalnika govora poteka v več zaporednih povezanih fazah. V vsaki fazi smo uporabili drug model učenja, ki je kompleksnejši od prejšnjega. Zvočne posnetke smo najprej razrezali na kratke odseke, oziroma okna, dolga 25 ms, razdalja med sosednjima oknomoma pa je 10 ms. Signal v vsakem oknu smo transformirali s Fourierjevo transformacijo in nato izračunali značilke MFCC (mel-frekvenčni kepstralni koeficienti) (Davis, Mermelstein, 1980, Charan in sod., 2017). Značilke MFCC dobimo tako, da v mel frekvenčni skali definiramo filtre (slika 2), ki se deloma prekrivajo med seboj. Mel-frekvence (m) dobimo iz frekvenc v hertzih (f) z enačbo:

$$m(f) = 1125 \ln \left(1 + \frac{f}{700} \right) \quad (1).$$

Za vsak filter zmnožimo spekter signala s filtrom in izračunamo logaritem spektralne energije znotraj filtra. Nad dobljenim izračunamo kosinusno transformacijo. Prvih 13 koeficientov obdržimo za naše značilke. Na teh značilkah smo učili naš sistem, v kasnejših fazah pa smo jim dodali še druge značilke, predvsem delta in delta-delta značilke, ki predstavljajo prvi, oziroma drugi časovni odvod značilk MFCC, torej njihovo spremembo v času.



Slika 2: Primer mel-frekvenčnih filtrov, prikazanih v mel skali na levi in isti filtri v normalni

Osnovo akustičnega modela predstavlja prikriti Markov model. Ta modelira spremembe zvočnega signala v času, to je prehode iz enega fonema v naslednjega. Fonemi predstavljajo skrita stanja v HMM, za njihov zapis pa moramo poznati izgovorjavo vsake besede. Za to uporabimo slovar, v katerem vsakemu geslu (besedi) pripišemo izgovor. Z drugimi besedami, zapišemo besedo in njen fonetični zapis. Opazovana stanja HMM so mešanica Gaussovih porazdelitev (GMM), ki opisujejo spekter posameznega časovnega izseka govornega signala.

Linearna diskriminantna analiza (LDA) je način, s katerim zmanjšamo število dimenzij v vektorjih značilk, obenem pa ohranimo diskriminantne značilnosti množice značilk. Rezultat so nižjedimenzionalni vektorji značilk, ki so manj korelirani in dobro razlikujejo med posameznimi razredi. Tako je učenje akustičnega modela lažje, oziroma hitreje.

Linearna transformacija z največjim verjetjem (angl. maximum likelihood linear transform - MLLT), ki je poseben primer »delno povezane kovariance« (angl. semi-tied covariance - STC), se pogosto uporablja v kombinaciji z linearno diskriminantno analizo (Gales, 1999). Če imamo opazovana stanja v HMM predstavljena z GMM, bi morali za vsako komponento, torej za vsako Gaussovo porazdelitev, izračunati celotno kovariančno matriko Σ_{jm} . Namesto tega opišemo kovariančno matriko, kot da je sestavljena iz dveh delov. Prvi del predstavlja diagonalna matrika, ki je specifična za vsako komponento (Σ_{jm}^{diag}). Drugi del je delno povezana matrika $\mathbf{H}^{(r)}$, ki ni specifična za vsako komponento, ampak za vsak razred komponent. Delno povezana matrika $\mathbf{H}^{(r)}$ lahko predstavlja poljubno število

komponent (Stuttle, 2003). Kovariančno matriko zapišemo kot:

$$\Sigma_{jm} = \mathbf{H}^{(r)} \Sigma_{jm}^{(diag)} \mathbf{H}^{(r)T} \quad (2).$$

Za od govornika neodvisen sistem za razpoznavanje govora nujno potrebujemo veliko število govorcev v učni množici. Akustični modeli naučeni na taki učni množici zato vsebujejo tudi parametre, ki razlikujejo med posameznimi govorniki. Naš cilj pa je razlikovati med različnimi besedami, ne glede na govornika. Problem rešujemo z učenjem s prilagajanjem govorniku (angl. speaker adaptive training - SAT). Začnemo z modelom neodvisnim od govornika, definiramo afino transformacijo

$$\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{b} \quad (3),$$

kjer postavimo začetna $\mathbf{A}=\mathbf{I}$ in $\mathbf{b}=\mathbf{0}$. S podatki le enega govornika ($\mathbf{o}(t)$), kjer značilke transformiramo z dano transformacijo (3), učimo akustični model eno iteracijo. Shranimo povprečja (\mathbf{P}), variance (μ) in posteriorne verjetnosti (γ) za vsako Gaussovo porazdelitev (m) in vsak časovni okvir (t). Ocenimo novi vrednosti \mathbf{A} in \mathbf{b} , z maksimizacijo spodnjega izraza (4).

$$K - \frac{1}{2} \sum \sum \gamma_m(t) \left(K^{(m)} + \log(|P^{(m)}|) - \log(|A|^2) + (A\mathbf{o}(t) + \mathbf{b} - \mu^{(m)})^T P^{(m)-1} (A\mathbf{o}(t) + \mathbf{b} - \mu^{(m)}) \right) \quad (4)$$

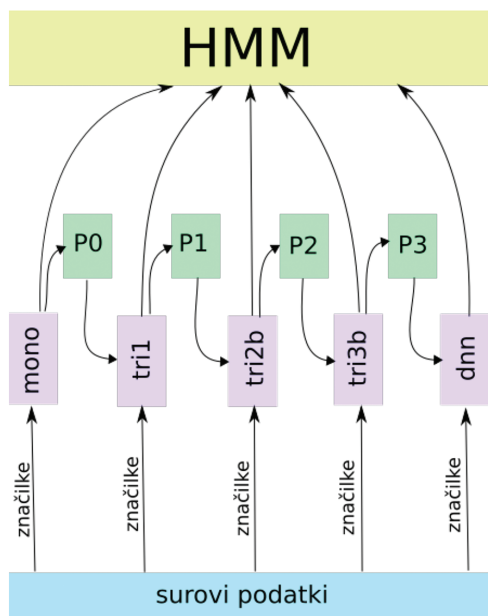
Ponovimo učenje z novimi vrednostmi \mathbf{A} in \mathbf{b} , in ponavljamo opisani postopek dokler vrednosti ne konvergirajo (Gales, 1998). Konstanta K v izrazu (4) je

odvisna le od verjetnosti prehodov med stanji HMM, $K^{(m)}$ pa je normalizacijska konstanta za vsak m . A in b sta različna za vsakega govorca, tako izraz (3) predstavlja transformacijo med kanoničnim modelom in modelom specifičnim za posameznega govorca.

Najprej smo uporabili različne modele GMM-HMM, povzete v tabeli 1. Uporabili smo monofonski model učenja (mono), trifonski model z delta in delta-delta značilkami (tri1), trifonski model z LDA in MLLT značilkami (tri2b), trifonski model z LDA, MLLT in dodanim prilagajanjem govorcu (SAT) (tri3b). Rezultate modela GMM-HMM smo uporabili za osnovo učenja hibridnega modela DNN-HMM. Skrita stanja v HMM in verjetnosti za prehode med njimi smo zamrzili. Za napovedovanje fonemov smo, namesto GMM, uporabili globoke nevronske mreže (DNN). Uporabili smo več različnih konfiguracij globokih nevronske mreže.

GMM-HMM

Monofonski model smo učili le na podmnožici učnih podatkov. Vzeli smo le 20.000 najkrajših izrekov. Razlog za učenje na krajših izrekih je, da smo zagotovili boljše ujemanje med zapisom besed in zvočnim posnetkom le-teh. V učnih podatkih nimamo zapisa, kje natančno znotraj zvočnega posnetka se nahajajo po-



Slika 3: Shema posameznih faz učenja, za vsako izluščimo značilke in učimo verjetnosti skritih stanj v HMM. Hkrati določimo poravnave fonemov z značilkami (P_n), s katerimi poravnamo foneme in značilke pred učenjem naslednje faze, kjer na novo učimo verjetnosti istega HMM.

samezne besede, niti kje natančno so posamezni fonemi znotraj besede. Za dober model moramo vsak fonem opisati zgolj s tistim delom posnetka, kjer je fonem izrečen. To skušamo ugotoviti med samim učenjem, kar je bolj zanesljivo pri krajših posnetkih. Po vsaki fazi učenja smo z Viterbijevim algoritmom čim bolje poravnali zvočne posnetke s fonemi, oziroma v naslednjih fazah s trifoni. Te poravnave uporabimo kot osnovo pri učenju naslednje faze (slika 3).

Pri preostalih fazah učenja akustičnega modela smo uporabili celotno učno množico. V učni množici ne nastopajo vsi možni trifoni, prav tako je teh veliko, zato jih uredimo v odločitveno drevo, glede na lastnosti posameznih fonemov v trifonu (samoglasnik ali soglasnik, in podobno). Z vsako naslednjo fazo smo povečali število Gaussovih porazdelitev (komponent v GMM) in število stanj, oziroma listov v našem odločitvenem drevesu, saj so modeli čedalje kompleksnejši. Poizkusili smo več različnih parametrov in primerjali rezultate na validacijski množici. Rezultati so bili zelo podobni, občutno izboljšanje smo dosegli le pri močno povečanem številu stanj, predvsem pa številu komponent v GMM. Zaradi bojzani, da bi s tem model prenaučili, smo se odločili za manjše število Gaussovih porazdelitev in manj listov v drevesu. Uporabljeni parametri so navedeni v tabeli 2. Pri monofonskem

Tabela 2: Parametri odločitvenega drevesa pri različnih akustičnih modelih

Učni model	Število komponent v GMM	Število listov	Razmerje med št. komponent in št. listov
mono	1000	/	/
tri1	12000	2000	6,00
tri2b	20000	3000	6,67
tri3b	30000	3500	8,57

modelu je odločitveno drevo trivialno, navedemo le število Gaussovih porazdelitev za primerjavo.

DNN-HMM

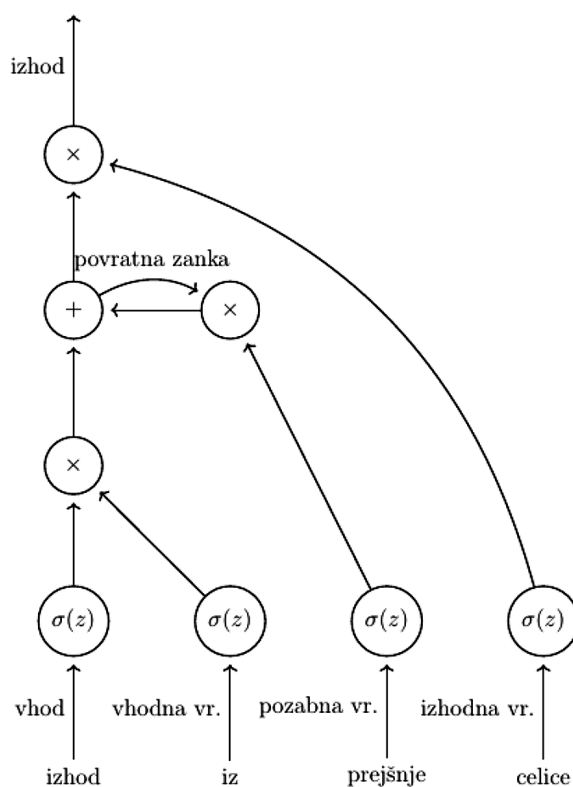
Pred učenjem globoke nevronske mreže smo umetno povečali količino učnih podatkov tako, da smo spremenili hitrost posnetkov. Originalno hitrost smo pomnožili s faktorji 0,9, 1,0 in 1,1 ter dobili trikrat toliko podatkov. Dodatno smo zmanjšali izhodno vzorčno frekvenco modela na tretjino, tako da je vsak vzorec sestavljen iz treh podvzorcev. Efektivno

izhod nevronske mreže ovrednotimo na vsakem tretjem (originalnem) vzorcu oziroma časovnem oknu. Nato zamaknemo podatke za en podvzorec in učimo ponovno. Končno zamaknemo podatke za še en podvzorec (skupno za dva podvzorca) in spet učimo. Nevronske mreže smo učili 5 dob, kar z zamikanjem vzorcev skupno nanese 15 dob. Za učenje nevronske mreže smo povečali spektralno ločljivost. Število značilnik MFCC smo dvignili s 13 na 40. Vse mreže imajo tako vhodni nivo dimenzije 40. Napovedujemo stanja v HMM, ki predstavljajo trifone, oziroma konkretnije liste odločitvenega drevesa. Teh stanj je po zadnji GMM-HMM fazi 2848, zato je taka tudi dimenzija izhodnega nivoja. Za optimizacijski algoritem smo uporabili stohastični gradientni spust, za funkcijo izgube pa logaritem verjetnosti pravilne sekvence fonemov. Za potrebe regularizacije smo dodali še en izhodni nivo enake dimenzije, kjer smo uporabili križno entropijo za funkcijo izgube. Pri učenju nastopata oba izhodna nivoja, pri napovedovanju pa zgolj prvi.

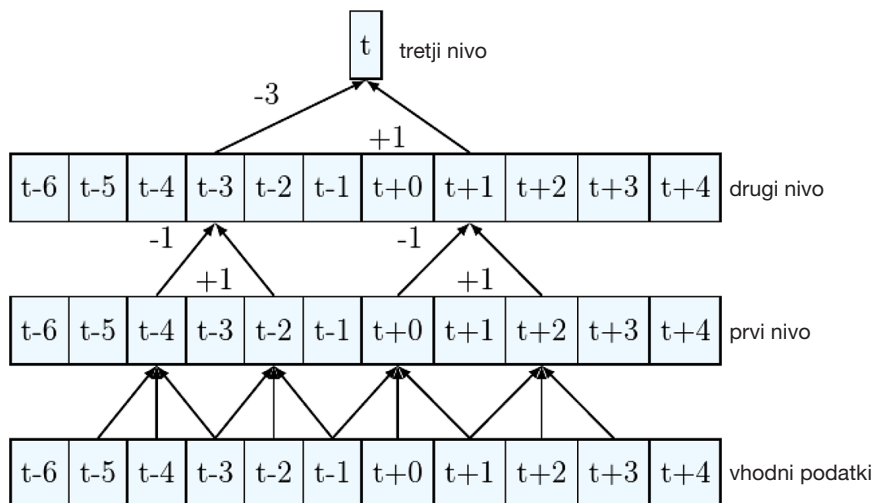
V zadnjem času so se v praksi uveljavile mreže LSTM, ki rešujejo težavo izginjajočega gradienta (Hochreiter & Schmidhuber, 1997). Mreža LSTM je

sestavljena iz posameznih celic, ki so med seboj povezane s povezavami naprej in s povratnimi povezavami. Vsaka posamezna celica je sestavljena iz več enot (slika 4). Poleg vhoda, aktivacijske funkcije in izhoda ima celica tudi povratno zanko znotraj celice ter troje vrat, ki utežujejo posamezne dele celice. Vhodna vrata dajo utež vhodnim podatkom celice. Izhodna vrata dajo utež izhodu iz celice. Pozabna vrata utežijo povratno zanko znotraj celice. Vsa vrata imajo sigmoidno aktivacijsko funkcijo, podatki na vhodu pa imajo poljubno nelinearno aktivacijsko funkcijo (Goodfellow, Bengio, & Courville, 2016); v našem primeru smo uporabili hiperbolični tangens (\tanh).

TDNN so vrsta nevronske mreže s povezavami naprej, vendar se učijo na časovno širšem kontekstu. Skriti nivoji v globoki nevronske mreže združujejo informacijo iz prejšnjega nivoja, tako da nevroni v vsakem naslednjem nivoju upoštevajo večji časovni razpon. Na primer, če na vhodnem nivoju vsak nevron predstavlja eno časovno okno, bo nevron na prvem skitem nivoju predstavljal pet časovnih oken. Združil bo informacijo enega okna s po dvema predhodnima in dvema naslednjima oknom (slika 5). Neuron na drugem skitem nivoju na primer združi



Slika 4: Sestava ene celice mreže LSTM.

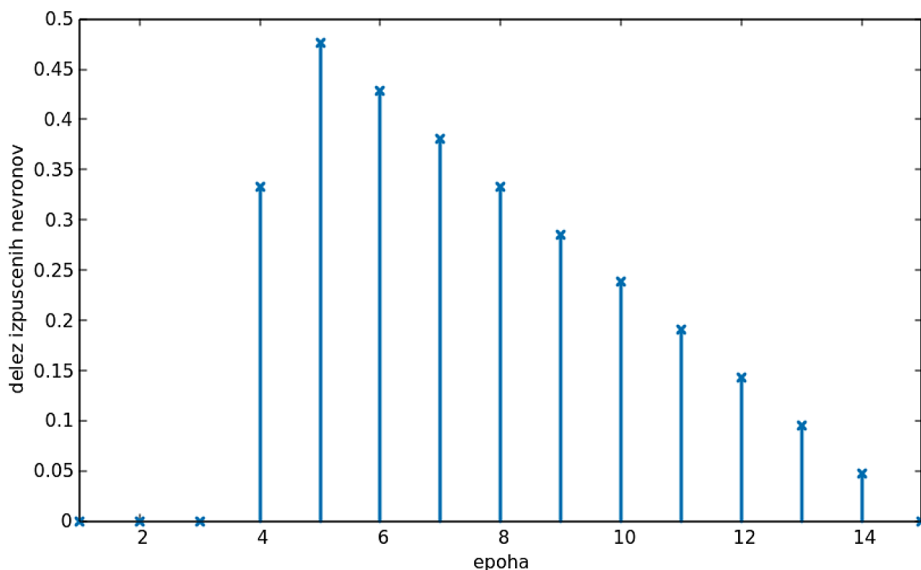


Slika 5: Primer časovnega združevanja v TDNN. Na prvem nivoju se združi informacija pri časovnih indeksih $t-1$ in $t+1$, glede na vhod. Na drugem nivoju se združi informacija pri časovnih indeksih $t-1$ in $t+1$ prvega nivoja. Na tretjem nivoju pri indeksih $t-3$ in $t+1$. Vsak nevron v tretjem nivoju vsebuje informacije devetih časovnih okvirjev vhodnih podatkov (med $t-5$ in $t+3$).

informacijo štirih nevronov iz prvega skritega nivoja: enega pri istem časovnem indeksu ter še enega z večjim in dva z manjšim časovnim indeksom (ali obratno) (Peddinti, Povey, Khudanpur, 2015).

Preizkusili smo več različnih konfiguracij mrež z različnim številom skritih nivojev in z različno povezanimi nivoji. Osredotočili smo se na arhitekturi TDNN in LSTM. Prva časovno zakasnjena mreža (*tdnn_1a*) na prvem skitem nivoju (nivo *lda*) združi časovni kontekst petih okvirjev pri časovnih indeksih, ki se razlikujejo za -2 , -1 , 0 , 1 in 2 glede na opazovan časovni indeks. Nivo je polno povezan in je dimen-

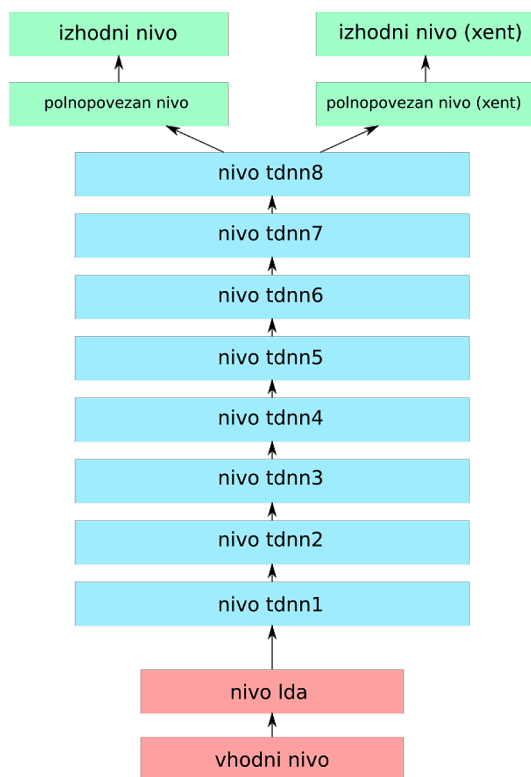
zije 40. Sledi osem polno povezanih nivojev (nivoji *tdnn1-8*) dimenzije 512, kjer uporabimo izpuščanje nevronov (angl. dropout) (Hinton in sod., 2012). Za dano moč izpuščanja α definiramo masko dimenzije 512, kjer ima vsak element maske naključno vrednost iz intervala $[1-2\alpha, 1+2\alpha]$. V vsakem časovnem okvirju pomnožimo izhode nevronov z enako masko (Povey, 2018). Moč izpuščanja α se spreminja tekom učenja in je prikazana na sliki 6. Izpuščanje nevronov je prisotno v nivojih *tdnn1-8*. Za aktivacijsko funkcijo uporabimo ReLU. V petih izmed osmih nivojev TDNN združimo časovni kontekst treh različnih okvirjev iz



Slika 6: Spreminjanje »deleža izpusčenih nevronov« oziroma vrednosti α tekom učenja.

Tabela 3: Združevanje po času pri mreži *tdnn_1a* - upoštevani so naštetih časovni indeksi prejšnjega nivoja, glede na trenutni časovni indeks *t*

Nivo	Časovni indeksi
lda	t-2, t-1, t+0, t+1, t+2
tdnn2	t-1, t+0, t+1
tdnn4	t-1, t+0, t+1
tdnn6	t-3, t+0, t+3
tdnn7	t-3, t+0, t+3
tdnn8	t-6, t-3, t+0

Slika 7: Shema konfiguracija *tdnn_1a* na dvema izhodoma pri učenju, *xent* pomeni, da ta veja uporablja križno entropijo za funkcijo izgube.

prejšnjega nivoja, kot je prikazano v tabeli 3. Sledita dva vzporedna polno povezana nivoja dimenzije 512 z aktivacijsko funkcijo ReLU, brez izpuščanja in nato izhodna nivoja (slika 7). Konfiguracijo smo povzeli po Kaldijevem projektu *vystadial_cz* za češki jezik (Denisov, 2018).

Časovno zakasnjena mreža *tdnn_1c* je podobna mreži *tdnn_1a*, le da vsebuje šest nivojev TDNN. Časovno združevanje je enako, le da pri tej konfiguraciji poteka pri nivojih *lda*, *tdnn2*, *tdnn3*, *tdnn4*, *tdnn5* in *tdnn6*. Časovno zakasnjena mreža *tdnn_1b* je identična mreži *tdnn_1c*, le da ne uporabljamo izpušča-

nja nevronov. Časovno zakasnjena mreži *tdnn_1d* in *tdnn_1e* vsebujeta 10 nivojev TDNN, kjer uporabljamo izpuščanje nevronov. Pri mrežah *tdnn_1d* in *tdnn_1e* smo preizkusili asimetrično združevanje podatkov po času še v dveh nivojih poleg zadnjega (glej tabelo 4). Nevronske mreže *tdnn_1a*, *tdnn_1b*, *tdnn_1c* in *tdnn_1d* smo učili 5 dob. Uporabili smo privzete Kaldijeve parametre paralelizacije, nastavili smo le število začetnih vzporednih nalog na 2 in število končnih nalog na 12. Pri nevronske mreži *tdnn_1e* smo oba parametra števila nalog nastavili na 1, kar naj bi bilo bolj optimalno, saj smo za učenje uporabljali le eno grafično kartico. Število dob smo zmanjšali na 3, tako da smo približno ohranili število iteracij in čas učenja. Število iteracij je bilo pri konfiguraciji *tdnn_1e* nekoliko večje kot pri konfiguraciji *tdnn_1d*, čas učenja pa daljši kljub manjšemu številu dob in enakim skritim nivojem.

Tabela 4: Združevanje po času pri mrežah *tdnn_1d* in *tdnn_1e* - upoštevani so naštetih časovni indeksi prejšnjega nivoja, glede na trenutni časovni indeks *t*.

Nivo	Časovni indeksi
lda	t-1, t+0, t+1
tdnn2	t-2, t+0, t+1
tdnn4	t-1, t+0, t+2
tdnn6	t-3, t+0, t+3
tdnn8	t-3, t+0, t+3
tdnn10	t-6, t-3, t+0

Prva mreža LSTM (*lstm_1b*) ima enak vhodni nivo kot mreže TDNN, prav tako ima enak prvi skriti nivo (*lda*). Sledijo štiri nivoji tipa LSTM, dimenzije 1024, in izhodni nivo. Nivo LSTM se od LSTM razlikuje v tem, da izhod nivoja ne pelje nazaj na vhod istega nivoja, ampak sta vmes dva vzporedna projekcijska nivoja. Izhod enega projekcijskega nivoja pelje na vhod nivoja LSTM, izhod drugega projekcijskega nivoja pa na vhod naslednjega skritega nivoja. Dimenzija projekcijskih nivojev je 256. Druga mreža LSTM (*lstm_1c*) ima po prvem skitem nivoju šest nivojev LSTM (ne LSTM) dimenzije 512. Pri *lstm_1b* in *lstm_1c* ni izpuščanja nevronov. Obe mreži LSTM smo učili 4 dobe.

Globoke nevronske mreže smo učili na grafični kartici Nvidia Tesla P100, ostale modele pa na procesorju z 8 nitmi. Učenje posamezne TDNN je trajalo približno 16 ur, LSTM približno 20 ur, vseh predhodnih faz na procesorju pa skupno 2-3 dni.

4.2 Jezikovni model

Jezikovni modeli so v obliki končnih pretvornikov (angl. finite-state transducer - FST) (Mohri, 1997). Naučimo jih z uporabo n-gramov, to so skupine n besed, ki se zaporedno pojavijo v besedilu. Na primer, v stavku: »Članek potrebuje pozornega bralca«, imamo štiri unigrame (posamezne besede), tri bigrame

(»članek potrebuje«, »potrebuje pozornega«, »pozornega bralca«), dva trigrami in en štirigram.

N-gramski jezikovni model ocenjuje verjetnost, da neka beseda w_i sledi danim n-1 predhodnim besedam. Uporabili smo 3-grame z Witten-Bellovim glajenjem (Witten, Bell, 1991). Witten-Bellovo glajenje ocenjuje verjetnost pojavitve besede w_i glede na predhodnih n-1 besed v n-gramu, $p(w_i | w_{i-n+1}^{i-1})$ z enačbo:

$$p(w_i | w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i) + N_{1+}(w_{i-n+1}^{i-1} \bullet) p(w_i | w_{i-n+2}^{i-1})}{\sum_{w_i} c(w_{i-n+1}^i) + N_{1+}(w_{i-n+1}^{i-1} \bullet)} \quad (5),$$

kjer je $c(w_{i-n+1}^{i-1})$ število pojavitev besed w_{i-n+1}^{i-1} , \bullet poljubna beseda, N_{1+} pa število besed \bullet , ki se v n-

gramu w_{i-n+1}^{i-1} \bullet pojavi vsaj enkrat (Chen, Goodman, 1998). Slednje lahko drugače zapišemo kot:

$$N_{1+}(w_{i-n+1}^{i-1} \bullet) = |\{w_i : c(w_{i-n+1}^i w_i) > 0\}| \quad (6),$$

torej velikost množice besed w_i , za katere velja, da je število pojavitev n-grama $w_{i-n+1}^i w_i$ (tj. n-1-gram w_{i-n+1}^i , ki mu sledi beseda w_i) večje od nič.

Uteži končnih pretvornikov ponovno ocenimo z uporabo globokih nevronske mreže. Preizkusili smo več različnih konfiguracij časovno zakasnjene mreže in mreže LSTM. Model s 3-grami smo naučili na besedilnem korpusu ccGigafida, modele z nevronske mreže pa na besedilnem korpusu ccKres (Logar in sod., 2013)). Za korpus ccKres smo se odločili, ker so različni viri besedil (internet, revije, časopisi, leposlovje, ...) bolj enakomerno zastopani kot pri korpusu ccGigafida. Prav tako je korpus ccKres manj obširen (približno 10 milijonov besed), kar je pohitrilo učenje z več kot en dan na nekaj ur.

V razdelku 5 ovrednotimo tri konfiguracije nevronske mreže, ki smo jih uporabili za učenje jezikovnega modela. Konfiguracijo *rnnlm_1a* sestavlja pet skritih nivojev, od tega trije nivoji TDNN in dva nivoja LSTM. Dimenzija vsakega nivoja je 800. Prvi skriti nivo je TDNN, kjer združimo časovna indeksa t in t-1. Sledi nivo LSTM. Dimenzija projekcijskih nivojev je 200. Tretji skriti nivo je TDNN, kjer združimo časovna indeksa t in t-2, sledi še en enak nivo LSTM. Zadnji skriti nivo je TDNN z združevanjem časovnih indeksov t in t-1. Opisano nevronske mreže smo učili 10 dob.

Konfiguracijo *rnnlm_1b* sestavlja trije skriti nivoji tipa TDNN. V prvem in drugem skitem nivoju

združimo časovna indeksa t in t-1, v tretjem skitem nivoju pa časovna indeksa t in t-2. Vsi nivoji so dimenzije 800. Nevronske mreže smo učili 15 dob. Konfiguracijo *rnnlm_1c* sestavljajo trije skriti nivoji tipa LSTM. Vsi nivoji so dimenzije 512. Nevronske mreže smo učili 15 dob.

Nevronske jezikovne modele smo učili na enaki grafični kartici kot akustične. Učenje je trajalo 3 do 5 ur.

5 REZULTATI IN ANALIZA

Uspešnost modelov pri razpoznavanju govora smo izmerili na validacijski in testni množici. Uspešnost podajamo kot delež besed, ki jih moramo dejanskemu besedilu dodati (vriniti), izbrisati in zamenjati, da bi dobili razpoznano besedilo. Ta mera za uspešnost se imenuje WER (besedna stopnja napak, angl. word error rate) in jo izračunamo kot:

$$WER = (V + I + Z) / B, \quad (7)$$

kjer V predstavlja število vrivanj, I število izbrisov, Z število zamenjav in B število vseh izgovorjenih besed v dejanskem besedilu.

Validacijska množica je sestavljena iz dveh korpusov, Gos (dev_gos) in Sofes (dev_sofes). Zaradi posebnosti obeh validacijskih množic, rezultate ločeno predstavljamo za vsak korpus posebej. Množica

dev_sofes vsebuje veliko tujih lastnih imen, predvsem imen letališč in mest, zaradi česar pričakujemo slabše rezultate. Množica dev_gos vsebuje nekaj izrekov, kjer nastopata dva govorca hkrati. Transkripcija korpusa Gos zapiše dva ločena stavka, ki pa se navezujeta na isti zvočni posnetek. Ločena stavka moramo združiti v en izrek, pri čemer ne vemo, v kakšnem vrstnem redu sta stavka izrečena. Možno je tudi, da govorca govorita hkrati. V teh primerih se transkripcija ne ujema z zvočnim posnetkom. Primer takega izreka je v tabeli 5. Rezultati v teh primerih

Tabela 5: Primerjava treh transkripcij izreka iz korpusa Gos, brez ločil. Primerjamo transkripcijo zapisano v korpusu Gos (T1), našo ročno transkripcijo (T2) in transkripcijo modela *tdnn_1a* (T3)

T1	»to je to je še pred volitvami bilo ja in sem takrat deloval kot državni sekretar na Ministrstvu za notranje zadeve kar v bistvu ni bila politična funkcija«
T2	»in sem takrat deloval kot državni sekretar na ministrstvu za notranje zadeve kar TO TO JE ŠE PRED VOLITVAMI JA politična funkcija«
T3	»in sem takrat deloval kot državni sekretar na ministrstvu za notranje zadeve kako je biti še pred volitvami je policija«

prikažejo slabše stanje od dejanskega.

Če ovrednotimo transkripcijo T3 iz tabele 5 glede na transkripcijo T1 dobimo napako WER=85,0%, glede na transkripcijo T2 pa WER=35,0%. Pri tem nismo upoštevali malih in velikih začetnic.

Metode, opisane v razdelku 4, smo ovrednotili na omenjenih validacijskih množicah in na testni množici. Primerjali smo rezultate pri različnih utežeh med jezikovnim in akustičnim modelom. Z večanjem uteži jezikovnega modela, narašča število vrtan, vendar pada število izbrisov. Število zamenjav se pri množici dev_sofes ne spreminja dosti v odvisnosti od uteži, pri množici dev_gos pa rahlo pada. Rezultate navajamo pri uteži, ki da najboljše rezultate na posamezni validacijski množici, rezultate na testni množici pa pri uteži, ki da najboljše rezultate na uniji obeh validacijskih množic. Najboljši rezultat je dosegla kombinacija akustičnega modela *tdnn_1a* in jezikovnega modela *rnnlm_1a*, kjer na testni množici dosežemo WER=27,16%.

V primerjavi z GMM-HMM, so vsi akustični modeli, naučeni z nevronske mreže, dosegli boljše rezultate, razen konfiguracije *tdnn_1e*, kjer je rezultat na validacijski množici dev_sofes slabši. Glede na to, da je zgradba nevronske mreže v *tdnn_1e* enaka tisti

v *tdnn_1d*, sklepamo, da je nismo dovolj dolgo učili, čeprav je bil čas učenja daljši. Mreža *tdnn_1d* doseže za 8,21% boljši rezultat na množici dev_gos kot mreža *tdnn_1e*. Najboljša časovno zakasnjena mreža je mreža *tdnn_1a*, ki na validacijskih množicah pravilno razpozna 7,42%, oziroma 17,39% več besed kot model GMM-HMM. Na testni množici je izboljšanje 14,41% (tabela 6). Mreže LSTM so prinesle manjše izboljšanje od časovno zakasnenih mrež. Najboljša mreža LSTM, v primerjavi z najboljšo mrežo TDNN, pravilno razpozna 2,95% manj besed na množici dev_gos in 4,16% manj besed na testni množici. Pravilno pa razpozna 0,46% več besed na množici dev_sofes (tabela 6). Tu je razlika zelo majhna, zato ne moremo sklepati o prednosti mreže LSTM za praktično rabo.

Razlike v uspešnosti med jezikovnimi modeli naučenimi z nevronske mreže so zelo majhne. Pri obeh validacijskih množicah in pri testni množici so razlike v deležu pravilno razpoznanih besed med nevronske jezikovnimi modeli manjše od 0,5%. Najboljši nevronske jezikovni model *rnnlm_1a* v primerjavi s 3-gramskim jezikovnim modelom pravilno razpozna 1,96% več besed na množici dev_gos, 4,71% več besed na množici dev_sofes in 2,61% več besed na testni množici.

Tabela 6: Rezultati (WER) akustičnih modelov GMM-HMM (*tri3b*), najboljše mreže TDNN (*tdnn_1a*) in najboljše mreže LSTM (*lstm_1b*) pri 3-gramskem jezikovnem modelu ter akustičnega modela TDNN pri najboljšem nevronske jezikovnem modelu (*rnnlm_1a*).

model	dev_gos	dev_sofes	test
<i>tri3b</i>	70,69 %	36,72 %	44,18 %
<i>tdnn_1a</i>	53,30 %	29,30 %	29,77 %
<i>lstm_1b</i>	56,25 %	28,84 %	33,93 %
<i>tdn_1a+rnnlm_1a</i>	51,34 %	24,59 %	27,16 %

Rezultate našega sistema smo primerjali z Googlovim vmesnikom Google Cloud speech-to-text (2018). Naključno smo izbrali pet izrekov iz testne množice, jih prepisali z Googlovim vmesnikom in transkripcije primerjali z transkripcijami našega razpoznavalnika. Napaka pri izbranih izrekih je WER=33,33% za Googlov sistem in WER=21,28% za naš sistem, kjer smo uporabili akustični model *tdnn_1a* in jezikovni model *rnnlm_1a*. Primerjava ni ravno reprezentativna, ker smo uporabili le 5 izrekov iz testne množice. Preliminarni rezultati kažejo, da naš model deluje bolje, vendar je potrebno več dodatnega testiranja, da

bi dobili statistično zanesljive rezultate. Glede na trenutno videno ima Googlov sistem manj vrivanj kot naš sistem, kar je prednost pri obotavljanju govorca. Mašila, kot sta »hmm« ali »eee« in prekinjene besede Googlov razpoznavalnik izpusti, medtem ko jih naš razpoznavalnik napačno zazna kot neke druge besede. Prednost našega razpoznavalnika v primerjavi z Googlovim je, da ima manjše število izbrisov in zamenjav. Torej zazna večje število izgovorjenih besed in te tudi bolj pravilno prepozna.

Podrobneje je Googlov vmesnik preizkusil David Čefarin (2016) v svojem diplomskem delu. Ugotovil je, da sistem dosega 61,72% točnost, oziroma WER=38,28% pri prostem govoru. Naš razpoznavalnik ima napako WER=27,16%, vendar omenjenih točnosti, oziroma napak Googlovega in našega sistema tudi tu ni moč neposredno primerjati med seboj, saj smo našega ovrednotili na drugi testni množici.

Naš razpoznavalnik je bil izdelan z namenom razpoznavanja splošnega, vsakdanjega govora. Pri učenju smo uporabili široko besedišče in veliko število govorcev, tako da razpoznavalnik ni specializiran za specifično področje uporabe (npr. za medicino ali za glasovno upravljanje nekega programa) niti ni prilagojen na posameznega govorca. Prednost tega je, da razpoznavalnik dosega podobne rezultate ne glede na vsebino govora ali na to, čigav govor razpoznavamo. Slabost te splošnosti pa je, da dosega slabše rezultate od specializiranih razpoznavalnikov, ki so omejeni zgolj na ozko besedišče ali enega govorca. Pred razpoznavanjem moramo govor shraniti v zvočno datoteko, obdelava posameznih datotek pa je daljša, kot če bi sproti brali zvočni signal z mikrofona.

6 SKLEPNE UGOTOVITVE

Predstavili smo sistem za razpoznavanje slovenskega govora z metodami globokih nevronske mreže. Za akustični model smo uporabili hibridni sistem DNN-HMM, kjer uporabimo globoke nevronske mreže za napovedovanje skritih stanj v HMM. Časovno zakasnjene nevronske mreže so se izkazale kot bolj uporabne od mrež z dolgim kratkoročnim spominom, saj dosejajo nekoliko boljše rezultate, obenem pa sta učenje in prepis hitrejša. Jezikovni model smo naučili z metodo n-gramov in z različnimi globokimi nevronske mreže. Model z n-grami dosega slabšo natančnost, vendar je mnogo hitrejši pri prepisu. Napovedovanje (s procesorjem, brez uporabe grafične kartice) z akustičnim nevronske modelov

in n-gramskim jezikovnim modelom traja približno tako dolgo, kot je dolžina posnetka, ki ga prepisemo. Če dodamo nevronske jezikovni model, se čas napovedovanja podaljša za do 100%. Med modeli z globokimi nevronske mreže ni velikih razlik v uspešnosti. Naš sistem dosega boljše rezultate pri razpoznavanju tekočega govora od primerljivih razpoznavalnikov za slovenščino.

Trenutna verzija razpoznavalnika je uporabna na vseh področjih, kjer visoka pravilnost razpoznavanja ni ključnega pomena. Pri učenju smo uporabili široko besedišče, zato je uporaben pri različnih tematikah, tako strokovnih kot pri vsakdanjem govoru. Slabost razpoznavalnika je njegova hitrost, saj nima možnosti sprotnega razpoznavanja govora.

Kljub uspešnemu razpoznavanju je možnosti za izboljšave še precej. Na točnost razpoznavanja vplivata velikost učnih podatkov in arhitektura razpoznavalnika. Smiselno bi bilo vključiti več govornih in besedilnih korpusov, prav tako bi lahko drugače obravnavali težave korpusa Gos. Problematične izreke bi lahko popolnoma izpustili iz učne množice ali jih transkribirali na novo. Pri učenju jezikovnega modela bi uspešnost lahko izboljšali z upoštevanjem morfoloških podatkov in z uporabo večjega besedilnega korpusa. Obstaja veliko različnih možnih konfiguracij nevronske mreže. V našem delu gotovo nismo našli najbolj optimalne, saj je možnosti preveč, učenje pa dolgotrajno.

Naš sistem bi lahko izboljšali z uporabo značilnik iVector, ki nekoliko izboljšajo točnost razpoznavanja. Največja prednost teh značilnik je, da se uporabljajo pri tekočem razpoznavanju govora. To pomeni, da ni potrebno vnaprej posneti celotnega zvočnega posnetka, ampak govor razpoznavamo sproti, med snemanjem.

Razpoznavalnik bi lahko razširili s sistemom, ki v besedilu avtomatsko postavi ločila. Tako besedilo bi bilo lažje berljivo in manj dvoumno, če bi tak sistem dobro naučili. Prepisan govor bi potreboval manj ročnega urejanja, saj moramo pri uporabi našega razpoznavalnika ločila ročno dodati.

Uporabniško izkušnjo bi lahko izboljšali z aplikacijo, ki z mikrofonom snema govor, sproti shranjuje posnetke in jih obdela za uporabo razpoznavalnika govora. Aplikacijo bi lahko uporabili tudi za že obstoječe posnetke govora.

Programska koda našega razpoznavalnika je dostopna na spletnem naslovu <https://github.com/>

MatejUlcar/kaldi/tree/slovenscina/egs/slovenscina. Načrtujemo, da bomo vsaj nekatere od zgornjih možnosti za izboljšave implementirali v sistem in izboljšano verzijo skupaj z najboljšimi naučenimi modeli objavili na repozitoriju Clarin.si.

LITERATURA

- [1] Alam, J., Kinnunen, T., Kenny, P., Ouellet, P., O'Shaughnessy, D. (2013). Multitaper MFCC and PLP features for speaker verification using i-vectors. *Speech Communications*, 55:237–251.
- [2] Bolka, A. (2016). *Samodejno razpoznavanje fonemov slovenskega govora z uporabo zbirke orodij Kaldi*. Diplomsko delo, Univerza v Ljubljani, Fakulteta za elektrotehniko.
- [3] Chen, S. F., Goodman, J. (1998). An Empirical Study of Smoothing Techniques for Language Modeling. *Harvard Computer Science Group Technical Report TR-10-98*.
- [4] Charan, R., Manisha, A., Karthik, R., Kumar, M. R. (2017). A Text-independent Speaker Verification Model: A Comparative Analysis. *2017 International Conference on Intelligent Computing and Control*.
- [5] Čefarin, D. (2016). *Preizkus Googlovega govornega programskega vmesnika za slovenski govorni jezik*. Diplomsko delo, Univerza v Ljubljani, Fakulteta za elektrotehniko.
- [6] Davis, S., Mermelstein, P. (1980). Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), strani 357–366.
- [7] Denisov, P. (2018). https://github.com/kaldi-asr/kaldi/blob/master/egs/vystadial_cz/s5b/local/chain/tuning/run_tdn_1a.sh, dostopano 6. 8. 2018.
- [8] Dobrišek, Simon; Žganec Gros, Jerneja; Žibert, Janez; Mihelič, France and Pavešič, Nikola, 2017, *Speech Database of Spoken Flight Information Enquiries SOFES 1.0*, Slovenian language resource repository CLARIN.SI, <http://hdl.handle.net/11356/1125>.
- [9] Dobrovoljc, Kaja; Krek, Simon; Holozan, Peter; Erjavec, Tomaž and Romih, Miro, 2015, *Morphological lexicon Sloleks 1.2*, Slovenian language resource repository CLARIN.SI, <http://hdl.handle.net/11356/1039>.
- [10] Donaj, G. (2015). *Avtomatsko razpoznavanje govora za pregibni jezik z uporabo morfoloških jezikovnih modelov s kontekstno odvisno strukturo*. Doktorska disertacija, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko.
- [11] Gales, M. (1998). Maximum Likelihood Linear Transformations for HMM-based Speech Recognition. *Computer Speech and Language*, vol. 12, strani 75–98.
- [12] Gales, M. (1999). Semi-tied Covariance Matrices for Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing*, vol. 7, strani 272–281.
- [13] Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing. *Journal of Artificial Intelligence Research*, 57:345–420.
- [14] Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- [15] Google. *Google cloud text-to-speech*. <https://cloud.google.com/speech-to-text/>, dostopano 18. 9. 2018.
- [16] Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., Estève, Y. (2018). TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. *20th International Conference, SPECOM 2018*, strani 198–208.
- [17] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R. (2012). Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. *ArXiv preprint arXiv:1207.0580*.
- [18] Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural computation*, 9, strani 1735–1780.
- [19] Logar, Nataša; Erjavec, Tomaž; Krek, Simon; Grčar, Miha and Holozan, Peter, 2013, *Written corpus ccGigafida 1.0*, Slovenian language resource repository CLARIN.SI, <http://hdl.handle.net/11356/1035>.
- [20] Logar, Nataša; Erjavec, Tomaž; Krek, Simon; Grčar, Miha and Holozan, Peter, 2013, *Written corpus ccKres 1.0*, Slovenian language resource repository CLARIN.SI, <http://hdl.handle.net/11356/1034>.
- [21] Mohri, M. (1997). Finite-state Transducers in Language and Speech Processing. *Computational Linguistics*, vol. 23, issue 2, strani 269–311.
- [22] Peddinti, V., Povey, D., and Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. *Sixteenth Annual Conference of the International Speech Communication Association*, strani 3214–3219.
- [23] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K. (2011). The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Catalog No.: CFP11SRW-USB.
- [24] Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohamadi, M., Khudanpur, S. (2018). Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. *Interspeech 2018* 3743–3747.
- [25] Rao, K., Peng, F., Sak, H., Beaufays, F. (2015). Grapheme-to-Phoneme Conversion Using Long Short-Term Memory Recurrent Neural Networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, strani 4225–4229.
- [26] Sak, H., Senior, A., Rao, K., Beaufays (2015a). Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition. *16th Annual Conference of the International Speech Communication Association (INTERSPEECH 2015)*, strani 1468–1473.
- [27] Sak, H., Senior, A., Rao, K., Beaufays, F., Schalkwyk, J. (2015b). Google voice search: faster and more accurate. *Google AI Blog*. <https://ai.googleblog.com/2015/09/google-voice-search-faster-and-more.html>, dostopano 10. 9. 2018.
- [28] Stuttle, M. N. (2003). *A Gaussian Mixture Model Spectral Representation for Speech Recognition*. Doktorska disertacija, Hughes Hall and Cambridge University Engineering Department.
- [29] Verdonik, Darinka; Potočnik, Tomaž; Sepesy Maučec, Mirjam and Erjavec, Tomaž, 2017, *Spoken corpus Gos VideoLectures 2.0 (transcription)*, Slovenian language resource repository CLARIN.SI, <http://hdl.handle.net/11356/1158>.
- [30] Witten, I. H., Bell, T. C. (1991). The Zero-Frequency Problem: Estimating the probabilities of Novel Events in Adaptive Text Compression. *IEEE Transactions on Information Theory*, vol. 37, no. 4, julij 1991.
- [31] Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., Zweig, G. (2017). The Microsoft 2016 conversational speech recognition system. *IEEE International Conference on Acoustics, Speech and Signal Processing*, strani 5255–5259.
- [32] Zwitter Vitez, Ana; Zemljarič Miklavčič, Jana; Krek, Simon; Stabej, Marko and Erjavec, Tomaž, 2013, *Spoken corpus*

- Gos 1.0, Slovenian language resource repository CLARIN.SI, <http://hdl.handle.net/11356/1040>.
- [33] Žgank, A., Donaj, G., Sepesy Maučec, M. (2014). Razpoznavnik tekočega govora UMB Broadcast News 2014: kakšno vlogo igra velikost učnih virov? V: *Zbornik 9. konference Jezikovne tehnologije, Informacijska družba - IS 2014*, strani 147-150.
- [34] Žgank, A., Verdonik, D., Sepesy Maučec, M. (2016). Razpoznavanje tekočega govora v slovenščini z bazo predavanj SI TEDx-UM. V: *Zbornik konference Jezikovne tehnologije in digitalna humanistika*, strani 186-189.

■

Matej Ulčar je leta 2018 magistriral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je zaposlen kot raziskovalec. Ukvarja se z razpoznavanjem govora in medjezikovnimi tehnologijami, predvsem z vektorskimi vložitvami besed.

■

Simon Dobrišek je izredni profesor in predstojnik Laboratorija za strojno inteligenco na Fakulteti za elektrotehniko Univerze v Ljubljani. Raziskovalno deluje na širšem področju tehnologij govorjenega jezika, razpoznavanja vzorcev, biometrije in umetnih inteligentnih sistemov. Posveča se tudi interdisciplinarnim raziskavam, ki segajo na področje jezikoslovja in glasoslovja ter varstva zasebnosti pri uporabi informacijskih in komunikacijskih tehnologij. V zadnjih letih je sodeloval pri več nacionalnih in mednarodnih raziskovalnih projektih s področja razvoja biometričnih tehnologij in tehnologij govorjenega jezika ter s področja etike in pravnega urejanja uporabe nadzornih tehnologij.

■

Marko Rognik-Šikonja je redni profesor in predstojnik Katedre za umetno inteligenco Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Raziskovalno se ukvarja s področji umetne inteligence, strojnega učenja, obdelave naravnega jezika in analize omrežij. Je avtor več kot 100 znanstvenih publikacij, ki so bile citirane več kot 4000-krat.

Prepoznavanje idiomatskih besednih zvez z uporabo besednih vložitev

Tadej Škvorc^{1, 2}, Marko Robnik Šikonja¹

¹Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana.

²Institut Jožef Stefan, Jamova cesta 39, 1000 Ljubljana.

tadej.skvorc@fri.uni-lj.si, marko.robnik@fri.uni-lj.si

Izvleček

Prisotnost idiomov v besedilu povzroča probleme številnim pristopom na področju obdelave naravnega jezika, saj jih računalniki težko prepoznajo. Strojno prepoznavanje takšnih izrazov še ni rešen problem. V zadnjih letih so razvili številne metode, ki lahko prepoznajo različne pomene besed glede na njihovo okolico in na podlagi tega zgradijo kontekstne vektorske vložitve besed. Takšne vložitve bi morale biti primerne za zaznavanje idiomov. Trenutni pristopi ali ne uporabljajo vektorskih vložitev ali pa uporabljajo ne-kontekstne vložitve. V delu pokažemo, kako lahko uporabimo kontekstne vložitve besed za ločevanje med dobessedno in idiomatsko rabo besed. Pokažemo, da lahko z različnimi značilkami (npr., s kontekstualnimi vektorji in razdaljami do srednjih kontekstualnih vektorjev za vsako besedo) zaznamo idiome prisotne v korpusu angleških besedil GloWbE.

Ključne besede: Večbesedni izrazi, obdelava naravnega jezika, besedilno rudarjenje, vektorske vložitve besed.

Abstract

The presence of idioms presents problems for many tasks in natural language processing, as they can be hard for computers to detect. Detecting such expressions and correctly determining their meanings has not yet been fully solved. In recent years, several methods for constructing contextual word embeddings have been proposed, which are capable of detecting the different meanings of the same word based on context. Such embeddings should be well suited to detecting idioms. Current approaches either do not use embeddings or use non-contextual embeddings. We have demonstrated that we can use contextual word embeddings to differentiate between literal and idiomatic word use. We have extracted various features (e.g. the contextual vectors and distance to the mean contextual vector for each word) and shown that they can be useful for detecting idiomatic word expressions present in the GloWbE corpus of English texts.

Keywords: Multi-word expression, natural language processing, text mining, word embeddings

1 UVOD

Prisotnost idiomov in ostalih večbesednih zvez oteži mnoge naloge obdelave naravnega jezika, kot so strojno prevajanje, analiza sentimenta in samodejno povzemanje, saj je lahko prenesen pomen besed različen od dobessednega (npr. "vreči puško v koruzo", ki pomeni obupati ali odnehati).

Idiome lahko zaznamo z uporabo slovarjev, vendar te v strojni obliki niso na voljo v vseh jezikih in niso primerni za pristope, ki morajo delovati na več jezikih hkrati. Nekateri trenutni pristopi temeljijo na uporabi modelov strojnega učenja. Savary et al.

[Savary et al., 2017] predstavijo pregled trenutnih pristopov in njihove rezultate na raziskovalnem izzivu PARSEME o samodejnem zaznavanju glagolskih večbesednih zvez. Takšni pristopi lahko poleg idiomov prepoznajo tudi druge večbesedne zveze. Na izzivu je najbolje deloval sistem TRANSITION [Constant and Nivre, 2016], ki je dosegel najboljše rezultate na številu različnih jezikov. Pozneje so Stodden et al. [Stodden et al., 2018] pristop izboljšali tako, da so model strojnega učenja zamenjali s konvolucijsko nevronske mrežo. Dobre rezultate je dosegel tudi sistem MUMULS [Klyueva et al., 2017].

Tabela 1: Izluščeni idiomi in število pojavitev v korpusu.

Idiom	Število izluščenih povedi	Število vseh povedi
Take place	1999	47522
Under the weather	200	472
The last straw	1000	1108
Bent out of shape	200	381
Hang in there	1000	2082
Break the ice	500	658
Live and learn	200	472

Zgoraj navedeni pristopi temeljijo na strojnem učenju in potrebujejo velike korpuse besedil z ročno označenimi idiomi, ki v veliko jezikih niso na voljo. Mnogim opravičeno bi bil v pomoč pristop, ki se ne zanaša na zunanje vire za zaznavanje idiomov. To lahko dosežemo z uporabo vektorskih vložitev besed. Namesto, da besede ponazorimo z znaki jih lahko ponazorimo z vektorji na takšen način, da ti vektorji odražajo pomen besed in sorodnosti med njimi. Takšne vložitve lahko pridobimo z nenadzorovanim strojnem učenjem, ki ne potrebuje ročno-označenih besedil. Za izgradnjo vektorskih vložitev obstajajo mnoge metode

[Mikolov et al., 2013, Pennington et al., 2014]. Številni avtorji so pokazali, da lahko tako pridobljeni vektorji kodirajo pomenske podobnosti in sorodnosti med besedami [Bojanowski et al., 2017, Mikolov et al., 2013]. Poleg tega lahko na podlagi pridobljenih vektorjev odgovorimo na vprašanja tipa »*A* je proti *B*, kot je *C* proti«, če poznamo vektorje besed *A*, *B* in *C*.

Vektorske vložitve lahko uporabimo za zaznavanje idiomov. Ker se besede v idiomski rabi pojavijo v drugačnih kontekstih kot v dobesedni rabi, lahko na podlagi vektorjev besed v kontekstu prepoznamo ali gre za idiomsko ali za dobesedno rabo. Gharbieh et al. [Gharbieh et al., 2016] predstavijo metodo za zaznavanje idiomov oblike glagol-samostalnik na podlagi povprečja vektorjev v okolici dvoumnih besed. Peng in Feldman [Peng and Feldman, 2015] predstavita dva podobna pristopa. Prvi temelji na skalarnem produktu vektorjev besed v kontekstu, drugi pa na njihovih kovariančnih matrikah. Oba pristopa za izgradnjo vektorjev uporabita metodo word2vec [Mikolov et al., 2013].

Pred kratkim so začeli razvijati kontekstne vektorske vložitve [Peters et al., 2018], s katerimi želimo upoštevati več možnih pomenov posamezne besede. Namesto, da vsaki besedi priredimo le en vektor,

priredimo vektor vsaki pojavitvi besede, pri čimer so vektorji odvisni od okolice pojavitve besede. Glavas et al. [Glavas et al., 2019] pokažejo, da lahko z uporabo takšnih vektorjev izboljšamo delovanje številnih pristopov na področju obdelave naravnega jezika.

V našem delu predstavimo, kako lahko kontekstne vložitve besed uporabimo za zaznavanje idiomov. Pri kontekstnih vložitvah za zaznavanje idiomov ne potrebujemo vektorjev besed konteksta, saj je informacija o kontekstu vsebovana v vektorju vsake pojavitve besede. Za razliko od obstoječih metod, ki delujejo na ne-kontekstnih vektorskih vložitvah, ne potrebujemo dodatnih metod, s katerimi iz vektorjev konteksta izluščimo koristne informacije. Namesto tega lahko neposredno uporabimo vektorje posameznih besed za zaznavanje idiomov. V našem delu uporabimo ELMo kontekstne vložitve [Peters et al., 2018], naučene na korpusu "1 Billion Words Benchmark" [Chelba et al., 2013], s katerimi izračunamo vložitve za besede iz sedmih različnih idiomov, ki so prisotni v korpusu "Global Web-based English Corpus" (GloWbE) [Davies and Fuchs, 2015]. Z vizualizacijo pokažemo, da so pridobljene vložitve zmožne razlikovati med prenesenimi in dobesednimi pomeni besed. Poleg tega pokažemo, da lahko z uporabo vložitev dosežemo visoko točnost pri strojnem zaznavanju idiomov. Glavna novost našega pristopa je, da ne potrebuje velikih, ročno-označenih korpusov, ki so potrebni za ostale podobne pristope.

2 ZAZNAVANJE IDIOMOV S KONTEKSTNIMI VLOŽITVAMI BESED

Za analizo zaznavanja idiomov potrebujemo veliko podatkovno množico z označenimi idiomi. Obstoječe množice (npr. množica za raziskovalni izziv PARSEME) so majhne in vsebujejo majhno število idiomov. Problem smo rešili tako, da smo zgradili lastno podatkovno množico iz besedil prisotnih v korpusu

“Global Web-based English Corpus (GloWbE)”, ki vsebuje 1.9 milijard besed iz spletnih besedil. Iz besedil smo izluščili povedi, v katerih se je pojavil eden izmed sedmih različnih angleških idiomov. Idiomi, število izluščenih povedi in število vseh povedi ki vsebujejo idiom, so prikazani v Tabeli 1.

V podatkovno množico smo dodatno vključili povedi z dobesednimi pomeni besed v idiomih. Iz korpusa smo naključno izbrali toliko povedi, da smo dobili uravnoteženo podatkovno množico (53,7% vsebuje idiome).

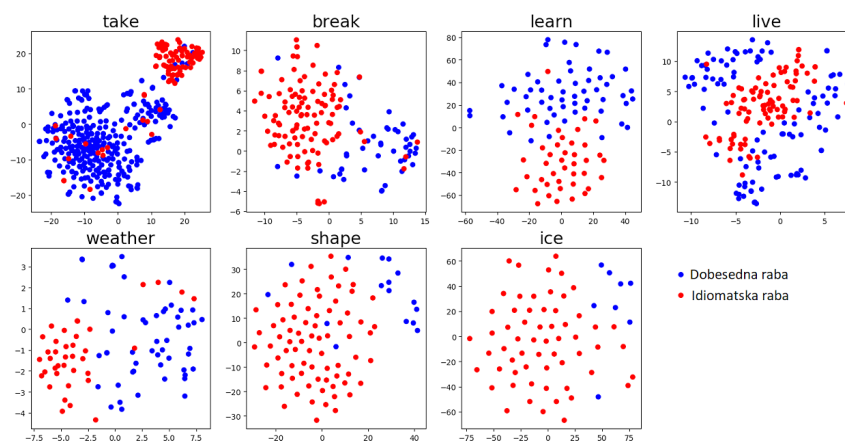
Tabela 2: Srednje vrednosti in standardni odkloni kosinusnih razdalj do srednje vrednosti vektorjev besede.

Beseda	Razdalja do srednje vrednosti (besedna raba)	Razdalja do srednje vrednosti (idiom)
Take	0.807 (0.070)	0.722 (0.073)
Weather	0.765 (0.055)	0.731 (0.035)
Straw	0.770 (0.010)	0.785 (0.029)
Bent	0.562 (0.011)	0.671 (0.039)
Shape	0.699 (0.126)	0.762 (0.032)
Hang	0.838 (0.055)	0.887 (0.023)
Ice	0.795 (0.014)	0.819 (0.033)
Live	0.769 (0.093)	0.845 (0.040)
Learn	0.877 (0.024)	0.839 (0.038)

Za izračun vložitev ELMo smo uporabili nevronska mrežo, ki je bila vnaprej naučena¹ na problemu jezikovnega modeliranja na korpusu “1 Billion Word Benchmark dataset [Chelba et al., 2013]”. Izračunali smo vektorje dimenzije 1024, kjer smo uporabili celotno poved kot kontekst besede. Vektorje smo vizualizirali z metodo t-SNE [Maaten and Hinton, 2008]. Poleg tega smo na podlagi vektorjev izvedli klasifikacijo med idiomatskimi in dobesednimi rabami besed. Preverili smo tudi, ali je razdalja do srednjega vektorja besede koristna za zaznavanje idiomov. Rezultate predstavimo v razdelku 3.

3 REZULTATI

Za vizualizacijo smo izračunali vektorje devetih različnih besed, ki so se pojavile tako v idiomih kot dobesedno. Vektorje smo pretvorili v dve dimenziji z metodo t-SNE. S tem lahko vidimo, kako se vektorji idiomov razlikujejo od vektorjev besed v njihovi dobesedni rabi. Pri analizi vidimo, da lahko z vektorji ELMo v nekaterih primerih ločim med idiomom in dobesedno rabo besed. Na Sliki 1 so vektorji besede “take” v idiomu “take place” jasno ločeni od ostalih rab besede. Vektorji, ki se prekrivajo verjetno predstavljajo ostale prenesene pomene besede “take”. Podobno velja za ostale besede, čeprav ločitev ni tako jasno razvidna.



Slika 1: Vizualizacija vektorjev besede “take”, “break”, “learn”, “live”, “weather”, “shape” in “ice”

Prav tako smo preverili, ali je razdalja posameznega vektorja do srednje vrednosti vseh vektorjev dovolj za zaznavanje idiomov. Statistično so dobesedne rabe besed bolj pogoste kot idiomske, torej bi morale biti bližje srednji vrednosti vseh vektorjev. Ta lastnost ne velja za našo podatkovno množico, saj smo jo zgradili tako, da je uravnotežena. Zaradi

tega smo za ta poskus srednjo vrednost vektorjev izračunali iz novega naključnega vzorca vseh besed v korpusu. Nato smo za 100 dobesednih in 100 idiomatskih rab besed iz naše podatkovne množice izračunali kosinusno razdaljo do srednjega vektorja. Povprečne razdalje in standardni odkloni so prikazani v Tabeli 2.

Tabela 3: Rezultati klasifikacije z 10-kratnim prečnim preverjanjem.

Klasifikacijski model	Povprečna klasifikacijska točnost	95 % interval zaupanja
Naključni gozdovi	0.928	+/- 0.045
Metoda podpornih vektorjev	0.926	+/- 0.060
Extreme gradient boosting	0.940	+/- 0.049

Iz tabele je razvidno, da se razdalje razlikujejo glede na rabo besede. Pri nekaterih besedah se izkaže, da so dobesedne rabe besed bližje srednji vrednosti vektorjev kot idiomske rabe. To nasprotuje naši hipotezi, vendar prisotnost razlik nakazuje da bi razdalje še vedno lahko bile koristne pri zaznavanju idiomov. Ena možna razlaga je, da so v nekaterih primerih srednje vrednosti obeh skupin blizu druga drugi, v katerem primeru bo razdalja odvisna predvsem od standardnega odklona distribucij vektorjev obeh skupin. Če so vektorji idiomatskih rab skoncentrirani skupaj, vektorji dobesednih rab pa daleč narazen bodo idiomatske rabe bližje srednji vrednosti vseh vektorjev.

Preverili smo tudi, ali lahko uporabimo klasifikacijske modele na vektorskih vložitvah za zaznavanje idiomov. Uporabili smo sledeče modele: naključni gozdovi [Breiman, 2001], metoda podpornih vektorjev [Suykens and Vandewalle, 1999] in extreme gradient boosting [Chen et al., 2015]. Rezultati 10-kratnega prečnega preverjanja so prikazani v Tabeli 3.

Najboljše rezultate smo dosegli z metodo extreme gradient boosting. Z vsemi metodami smo dosegli visoke rezultate. Čeprav je naša podatkovna množica je preprosta, zaradi česar bi bili rezultati na zahtevnejših primerih verjetno slabši, to nakazuje da so kontekstne vektorske vložitve primerne za zaznavanje idiomov.

4 ZAKLJUČEK

Zaznavanje idiomov je pomembna za veliko opravil na področju obdelave naravnega jezika. Pokazali

smo, da lahko kontekstne vektorske vložitve uporabimo za zaznavanje idiomov, pri čemer ne potrebujemo ročno označenih korpusov. Na umetni podatkovni množici z vektorji lahko ustrezno ločimo dobesedne in idiomske rabe in s klasifikacijo dosežemo 94 % točnost.

V nadaljnjem delu nameravamo razširiti evaluacijo z uporabo večje podatkovne množice in preveriti, ali lahko z vektorskimi vložitvami izboljšamo delovanje trenutnih pristopov. Trenutna evaluacija je omejena na le sedem idiomov, zaradi česar je težko vedeti, kako dobro naš pristop deluje na vseh idiomih, predvsem na takšnih, ki se v besedilih redko pojavijo.

ZAHVALA

Raziskovalno delo je bilo sofinancirano s strani Javne agencije za raziskovalno dejavnost Republike Slovenije, št. projekta P2-0209 in P2-0103.

LITERATURA

- [1] [Bojanowski et al., 2017] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- [2] [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- [3] [Chelba et al., 2013] Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.
- [4] [Chen et al., 2015] Chen, T., He, T., Benesty, M., Khotilovich, V., and Tang, Y. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, pages 1–4.
- [5] [Constant and Nivre, 2016] Constant, M. and Nivre, J. (2016). A transition-based system for joint lexical and syntactic analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 161–171.

¹ <https://tfhub.dev/google/elmo/2>

- [6] [Davies and Fuchs, 2015] Davies, M. and Fuchs, R. (2015). Expanding horizons in the study of World Englishes with the 1.9 billion word Global Web-based English Corpus (GloWbE). *English World-Wide*, 36(1):1–28.
- [7] [Gharbieh et al., 2016] Gharbieh, W., Bhavsar, V., and Cook, P. (2016). A word embedding approach to identifying verb-noun idiomatic combinations. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 112–118, Berlin, Germany. Association for Computational Linguistics.
- [8] [Glavas et al., 2019] Glavas, G., Litschko, R., Ruder, S., and Vulic, I. (2019). How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. *arXiv preprint arXiv:1902.00508*.
- [9] [Klyueva et al., 2017] Klyueva, N., Doucet, A., and Straka, M. (2017). Neural networks for multi-word expression detection. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 60–65.
- [10] [Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
- [11] [Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- [12] [Peng and Feldman, 2015] Peng, J. and Feldman, A. (2015). Automatic idiom recognition with word embeddings. In *Information Management and Big Data*, pages 17–29. Springer.
- [13] [Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- [14] [Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAA-CL*.
- [15] [Savary et al., 2017] Savary, A., Ramisch, C., Cordeiro, S., Sangati, F., Vincze, V., QasemiZadeh, B., Candito, M., Cap, F., Giouli, V., and Stoyanova, I. (2017). The parseme shared task on automatic identification of verbal multiword expressions.
- [16] [Stodden et al., 2018] Stodden, R., QasemiZadeh, B., and Kallmeyer, L. (2018). Trapacc and trapaccs at parseme shared task 2018: Neural transition tagging of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 268–274.
- [17] [Suykens and Vandewalle, 1999] Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

■

Tadej Škvorc je mladi raziskovalec in doktorski študent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Ukvarja se s področjem obdelave naravnega jezika in umetne inteligence.

■

Marko Robnik-Šikonja je redni profesor in predstojnik Katedre za umetno inteligenco Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Raziskovalno se ukvarja s področji umetne inteligence, strojnega učenja, obdelave naravnega jezika in analize omrežij. Je avtor več kot 100 znanstvenih publikacij, ki so bile citirane več kot 4000-krat.

Primerjava segmentacijskih metod globokega učenja za detekcijo ovir na vodi

Borja Bovcon, Matej Kristan

University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, Ljubljana, Slovenia

borja.bovcon@fri.uni-lj.si, matej.kristan@fri.uni-lj.si

Izvleček

Razvoj segmentacijskih metod globokega učenja za detekcijo ovir na vodi je v precejšnjem zaostanku v primerjavi z razvojem na sorodni domeni avtonomnih vozil (AGV). Do nedavnega je bil glavni razlog za to pomanjkanje ustreznih podatkovnih zbirk ter dejstvo, da metode razvite za AGV niso primerne za aplikacijo na vodno okolje zaradi domenskih specifik. Trenutno ni jasno katere arhitekture so najprimernejše za vodno domeno. Zato smo izbrali tri popularne arhitekture metod globoke semantične segmentacije (U-Net, PSPNet, DeepLab2), jih učili na podatkovni zbirki MaSTr1325 ter evalvirali njihovo delovanje na ločeni podatkovni zbirki Modd2. Eksperimentalna analiza je pokazala pomembnost globokega ogrodja arhitekture, ki omogoča učenje različnih izgledov vode. Kot zahteven problem so se izkazali artefakti v vodi (morska pena, odbleski in odsevi), ki prožijo lažne alarme in s tem motijo postopek avtonomne navigacije.

Ključne besede: detekcija ovir, globoko učenje, morski okoliš, primerjava metod, semantična segmentacija

Abstract

Development of deep segmentation methods for water surface obstacle detection lags far behind the development on a related domain of autonomous ground vehicles (AGVs). The main reasons is that much fewer curated training datasets exist and that methods developed for the AGVs rely on domain-specific assumptions and cannot be readily applied to the marine environment. It is currently unclear which architectures are most appropriate for marine environments. To address this, we selected three common state-of-the-art deep segmentation architectures (U-Net, PSPNet, DeepLab2), trained them on MaSTr1325 dataset and evaluated their performance on a separate test set Modd2. Experimental analysis shows a significant importance of the capacity of the deep backbone architecture, required to cope with the varying water appearance. Artefacts, such as foam fragments, glitter and reflections, present a leading cause of false positive detections which disturb autonomous navigation of the USV.

Keywords: benchmark, deep learning, marine environment, obstacle detection, semantic segmentation

1 UVOD

Razvoj morske robotike je privedel do vzpostavitve novega razreda avtonomnih plovil (USV), t.j. robotskih čolnov. Ta plovila so cenovno dostopna in prenosljiva za plovbo v plitvih vodah in ozkih marinah. Uporabljajo se predvsem za priobalno patroljiranje in oddaljeni pregled težko dostopnih okolji, ki lahko predstavljajo nevarnost za človeka. To zahteva visoko stopnjo avtonomije, ki je predvsem odvisna od pravočasnega odkrivanja in izogibanja bližnjim oviram ter plavajočim odpadkom. Za odkrivanje ovir ter boljše razumevanje okolice in njenega plovnega območja lahko uporabimo številne robotske senzorje

(RADAR [Almeida et al., 2009, Onunka and Bright, 2010], SONAR [Heidarsson and Sukhatme, 2011], LIDAR). Omejitve plovnosti ter napajanja USV-jev pa omejujejo namestitev potencialnih senzorjev. Zato se kamere kot lahki in informativni senzorji, vedno bolj uveljavljajo kot primerni mehanizmi za odkrivanje ovir.

Na sorodni domeni avtonomnih vozil (AGV) je bilo razvitih že veliko algoritmov računalniškega vida za detekcijo ovir s pomočjo kamere. Kot najobsebnije so se izkazale metode semantične segmentacije, ki temeljijo na globokem učenju [Alvarez et al., 2012b, Alvarez et al., 2012a, Levi et al., 2015, Olivei-

ra et al., 2016]. Metode globokega učenja zahtevajo ogromne količine natančnih in reprezentativnih učnih vzorcev. Domena AGV ima na voljo več takšnih ustreznih podatkovnih zbirk kot so KITTI [Geiger et al., 2012], Cityscapes [Cordts et al., 2016] in BDD100k [Yu et al., 2018]. Razvoj metod semantične segmentacije na področju USV-jev pa precej zaostaja. Zaradi pomanjkanja učnih podatkovnih zbirk pomorskega okoliša, je bilo izvedenih le nekaj poskusov uporabe globokega učenja. Lee et al. [Lee et al., 2018] so sestavili lastno podatkovno zbirko sedmih tipov ladij ter uporabili Faster R-CNN [Ren et al., 2015] za detekcijo le-teh. Vendar njihova metoda ne omogoča detekcije poljubnih ovir, ki niso bile videne v učni fazi. Cane et al. [Cane and Ferryman, 2018] so preizkusili delovanje treh metod globoke semantične segmentacije na pomorski domeni. Metode so učili na vsesplošni podatkovni zbirki ADE20k, ki pa vizualno ni reprezentativna za okolje, v katerem USV deluje. Rezultat je relativno slaba segmentacija.

V tem članku evalviramo tri popularne arhitekture metod globoke semantične segmentacije, ki so se izkazale uspešne na področju AGV-jev na podatkovni zbirki, ki je reprezentativna za robotska plovila.

2 METODE SEMANTIČNE SEGMENTACIJE

Izbrali smo tri popularne arhitekture metod globoke semantične segmentacije, ki dosegajo izjemne rezultate in predstavljajo osnovo velikemu številu naprednih metod semantične segmentacije:

- 1) *Arhitektura U-Net* [Ronneberger et al., 2015] je bila uspešno aplicirana na širok spekter različnih problemov. Sestavljata jo kodirnik, ki zajema kontekst, ter simetrični dekodirnik, ki poskrbi za natančno lokalizacijo. Korespondenčni nivoji kodirnika in dekodirnika so med seboj povezani, kar preprečuje problem izginjanja gradienta tekom učenja.
- 2) *Arhitektura PSPNet* [Zhao et al., 2017] je dosegla izjemne rezultate segmentacije urbanega okoliša. Ogradje arhitekture temelji na konceptu popularne mreže ResNet [He et al., 2016] z razširjenimi konvolucijami. Mreža vsebuje tudi modul piramidnega združevanja (*Pyramid-Pooling Module*) za povečavo vizualnega konteksta.
- 3) *Arhitektura DeepLab2* [Chen et al., 2018a] podobno temelji na ogradi ResNet [He et al., 2016] z razširjenimi konvolucijami. Kombinirana je z modulom *Atrous Spatial Pyramid Pooling* (ASPP), ki omogoča zajem konteksta na različnih skalah. Pogojno slu-

čajno polje (Conditional Random Field - CRF) poskrbi za izpopolnitev segmentacije. DeepLab2 implementira dve različici arhitekture — enoslojno in večslojno. Večslojna različica sprejme kot vhod sliko na različnih skalah ter vsako vzporedno in neodvisno segmentira. Rezultate segmentacij sešteje ter jih izpopolni z modulom CRF. V nadaljnjem bomo označili enoslojno različico DeepLab2 z DL2s, večslojno pa z DL2m.

3 EVALVACIJA

3.1 Podatkovno zbirka in podrobnosti implementacija

Izbrane arhitekture (Poglavje 2) smo učili na podatkovni zbirki MaSTr1325 [Bovcon et al., 2019], ki vključuje 1325 reprezentativnih, pomorskih slik, zajetih z USV-jem. Za zajem slik je bil uporabljen stereo sistem kamer *Vrmagic VRmMFC*, ki ga sestavljata dva senzorja CCD *Vrmagic VRmS-14/C-COB* z medsebojno oddaljenostjo 0,3 metra, lečo *Thorlabs MVL-4WA* z goriščno razdaljo 3,5 milimetra ter vidnim kotom 132,1°. Slike podatkovne zbirke so označene natančno do posameznega slikovnega elementa za tri semantične komponente — nebo, morje ter okoliš z ovirami. Pred učenjem smo izvedli povečevanje zbirke z naključnimi rotacijami (5, 15 stopinj), zrcaljenjem preko vertikalne osi ter ustreznim barvnim prenosom [Reinhard et al., 2001]. Po povečevanju je učna množica vsebovala 53000 učnih slik.

Za učenje metod smo uporabili cenilno funkcijo *softmax cross-entropy* ter optimizator *momentum*. Parameter začetne stopnje učenja smo nastavili na 1-4 ter nanj aplicirali funkcijo polinomskega upada s parametrom 0,9. Za inicializacijo ograda metod PSPNet in DeepLab2 smo uporabili uteži, pred naučene na podatkovni zbirki ImageNet [Deng et al., 2009]. Postopek finega učenja metod na podatkovni zbirki MaSTr1325 smo izvajali 13 epoh.

Evalvacijo smo opravili na ločeni podatkovni zbirki Modd2 [Bovcon et al., 2018], ki je bila zajeta z enakim USV-jem kot učna podatkovna zbirka. To je trenutno najzahtevnejša in največja testna podatkovna zbirka morskega okoliša. Vsebuje 28 raznolikih sekvenc z zahtevnimi vremenskimi pogoji kot so megla, nizko ležeče sonce, itd. Slike v podatkovni zbirki so sinhronizirane z meritvami senzorja inercij (IMU). Za evalvacijo kakovosti segmentacije smo uporabili dva tipa metrik: (i) klasične segmentacijske metrike

(*mean pixel accuracy, mean intersection-over-union in frequency-weighted intersection-over-union*) ter (ii) metriko za detekcijo ovir, kot so jo predstavili Kristan *et al.* [Kristan *et al.*, 2016].

Ovire v neposredni bližini USV-ja predstavljajo večjo nevarnost kot oddaljene. Za natančnejšo evaluacijo bližnje nevarnosti, smo s pomočjo meritev IMU v slikah testne množice določili t.i. kritični pas. To

je krožno območje z radijem 15 metrov, centrirano na trenutni lokaciji USV-ja. Radij krožnega območja smo določili pod predpostavko, da se USV giblje s povprečno hitrostjo 1.5 m/s. Za izogibanje oviram, ki so oddaljene več kot 15 metrov, je na voljo potemtakem vsaj deset sekund, kar ne predstavlja nevarnosti. Oviram, ki so bližje od 15 metrov, pa se moramo izmakniti v manj kot deset sekundah.

Tabela 1: Klasične segmentacijske metrike (levo) in metrike za detekcijo ovir (desno). Napaka približne ocenitve roba morja je označena z *ped* in merjena v pikslih, število pozitivnih detekcij s TP (*true positives*), število lažnih detekcij s FP (*false positives*), število zgrešenih ovir s FN (*false negatives*) ter splošna F-mera, merjena v procentih. Število detekcij TP je obratno sorazmerno s številom detekcij FN, F-mera pa je definirana kot harmonična sredina priklica ($TP/(TP+FN)$) in natančnosti ($TP/(TP+FP)$), izračunana z uporabo enačbe $F = 2TP / (2TP + FP + FN)$. V oglatih oklepajih so zapisane detekcije znotraj nevarnostnega območja.

Architecture	Mean PA	Mean IOU	fw-IOU	TP	FP	FN	F-mera
U-Net (Ronneberger <i>et al.</i> , 2015)	91,96	96,80	90,65	592 (208)	3706 (3337)	87 (43)	23,8 (11,0)
PSPNet (Zhao <i>et al.</i> , 2017)	90,98	94,50	88,22	346 (126)	54 (39)	333 (125)	64,1 (60,6)
DL2s (Cheb <i>et al.</i> , 2018a)	92,65	97,49	91,48	369 (167)	108 (36)	310 (84)	63,8 (73,6)
DL2m (Chen <i>et al.</i> , 2018b)	92,61	97,41	91,38	304 (147)	65 (14)	375 (104)	58,2 (71,4)

3.2 Analiza rezultatov

Rezultati, pridobljeni s klasičnimi segmentacijskimi metriki, se nahajajo v Tabeli 1. Opazimo, da DL2s doseže konsistentno najboljše rezultate v vseh treh segmentacijskih metrikah. Zgolj na podlagi rezultatov klasičnih segmentacijskih metrik ne moremo sklepati katera od metod bi bila najprimernejša za avtonomno navigacijo. Dodatno potrebno informacijo nam priskrbi metrika za detekcijo ovir (Tabela 1). Rob morja najboljše ocenita DL2s in DL2m, sledita jima pa U-Net ter PSPNet, pri čemer je napaka ocene roba morja metode PSPNet dvakrat večja. To je razvidno na spodnji vrstici Slike 1. Največje število pravih detekcij, ter s tem posledično tudi najmanjše število detekcij FN, doseže U-Net. Po natančnem vizualnem pregledu smo ugotovili, da je metoda U-Net zelo občutljiva na artefakte v vodi (Slika 1 srednja vrstica). Rezultat je veliko pravih detekcij, ampak še večje število lažnih detekcij, ki so posledica morske pene, valov, odbleskov ter odsevov. Najboljši rezultat F-mere doseže PSPNet, kateremu sledita DL2s in DL2m. Znotraj kritičnega območja se najboljše obnese DL2s, sledijo pa mu DL2m, PSPNet in U-Net. Podrobnejša analiza je pokazala, da večstopenjska različica DL2 zabeleži manj detekcij, saj se manjše ovire in artefakti izgubijo na skalah nižje ločljivosti (Slika 1 zgornja vrstica). To posledično privede do zmanjšanja števila detekcij FP in TP ter istočasno do povečanja števila detekcij FN,

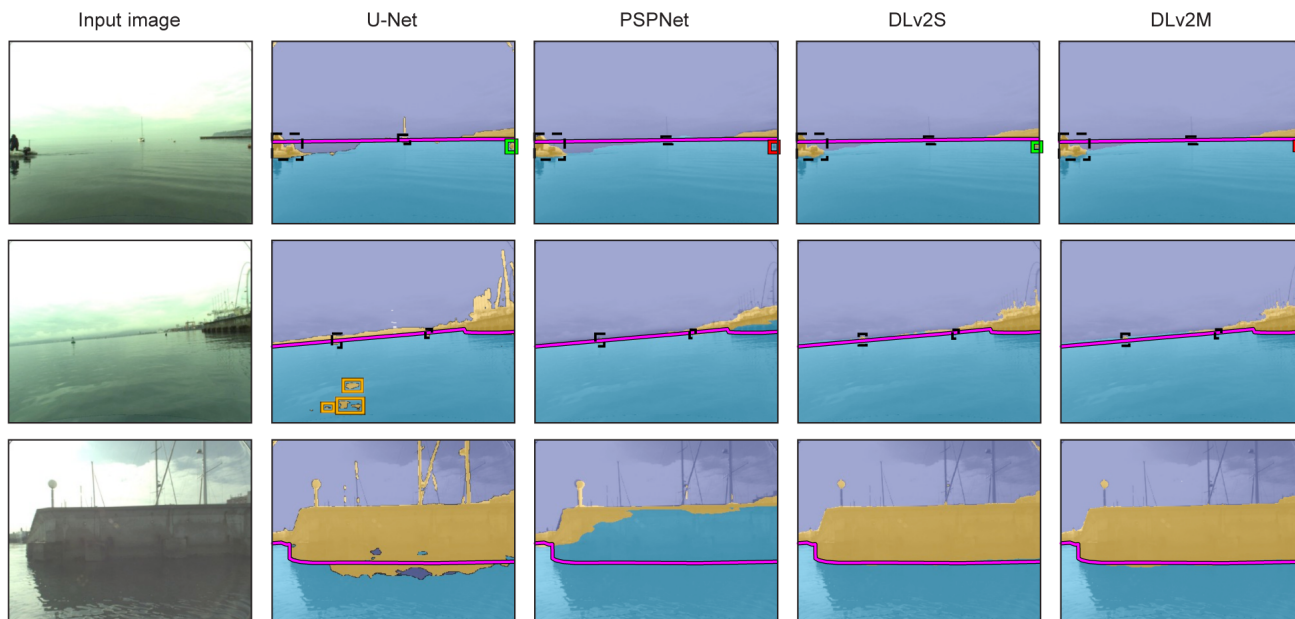
kar predstavlja nevarnost plovilu. Na podlagi izpostavljenih rezultatov doseže DL2s najboljše razmerje med natančnostjo ocene roba morja, številom pravih detekcij ter številom lažnih alarmov.

4 SKLEP

Predstavili smo primerjavo treh popularnih arhitektur za globoko semantično segmentacijo. Izbrane metode smo učili na podatkovni zbirki MaStr1325 morskega okoliša, njihovo kakovost detektiranja ovir pa testirali na ločeni podatkovni zbirki Modd2.

Rezultati so pokazali, da je problem segmentacije vode zelo težak, saj njen vizualni izgled zelo variira glede na vremenske pogoje in morsko dno. Za uspešno segmentacijo je potrebno globoko ogrodje arhitekture, kot je ResNet, ki omogoča mreži naučiti se različnih izgledov vode.

Empirični rezultati kažejo, da lažne detekcije, ki jih povzročajo artefakti (morska pena, odbleski in odsevi), predstavljajo zahteven problem za avtonomno navigacijo, saj plovilu prožijo lažne alarme, ki ovirajo nemoteno plovbo. Nevarnejši problem predstavljajo zgrešene detekcije, saj lahko vodijo do trka in poškodbe avtonomnega plovila. V prihodnjem delu nameravamo nasloviti problem lažnih detekcij z vpeljavo časovne komponente, problem zgrešenih detekcij pa z ustreznimi spremembami arhitekture mreže.



Slika 1: Kvalitativna primerjava delovanja metod. Nebo, ovire in voda so označene s temno-plavo, rumeno ter cian barvo. Zlati standard roba morja je označen z roza črto, zlati standard ovir pa s črkanim pravokotnikom. Lažne detekcije označujejo rdeči pravokotniki, pravilne detekcije pa zeleni pravokotniki.

LITERATURA

- [1] [Almeida et al., 2009] Almeida, C., Franco, T., Ferreira, H., Martins, A., Santos, R., Almeida, J. M., Carvalho, J., and Silva, E. (2009). Radar based collision detection developments on USV ROAZ II. In *OCEANS - EU*, pages 1–6.
- [2] [Alvarez et al., 2012a] Alvarez, J. M., Gevers, T., LeCun, Y., and Lopez, A. M. (2012a). Road scene segmentation from a single image. In *European Conference on Computer Vision*, pages 376–389. Springer.
- [3] [Alvarez et al., 2012b] Alvarez, J. M., LeCun, Y., Gevers, T., and Lopez, A. M. (2012b). Semantic road segmentation via multi-scale ensembles of learned features. In *European Conference on Computer Vision*, pages 586–595. Springer.
- [4] [Bovcon et al., 2019] Bovcon, B., Muhovič, J., Perš, J., and Kristan, M. (2019). The mastr1325 dataset for training deep usv obstacle detection models. In *IROS*.
- [5] [Bovcon et al., 2018] Bovcon, B., Perš, J., Kristan, M., et al. (2018). Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation. *Robotics and Autonomous Systems*, 104:1–13.
- [6] [Cane and Ferryman, 2018] Cane, T. and Ferryman, J. (2018). Evaluating deep semantic segmentation networks for object detection in maritime surveillance. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE.
- [7] [Chen et al., 2018a] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848.
- [8] [Chen et al., 2018b] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018b). Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*.
- [9] [Cordts et al., 2016] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223.
- [10] [Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.
- [11] [Geiger et al., 2012] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE.
- [12] [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [13] [Heidarsson and Sukhatme, 2011] Heidarsson, H. K. and Sukhatme, G. S. (2011). Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar. In *ICRA 2011*, pages 731–736.
- [14] [Kristan et al., 2016] Kristan, M., Kenk, V. S., Kovačič, S., and Perš, J. (2016). Fast image-based obstacle detection for unmanned surface vehicles. *IEEE TCYB*, 46(3):641–654.
- [15] [Lee et al., 2018] Lee, S.-J., Roh, M.-I., Lee, H.-W., Ha, J.-S., Woo, I.-G., et al. (2018). Image-based ship detection and classification for unmanned surface vehicle using real-time object detection neural networks. In *The 28th International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers.
- [16] [Levi et al., 2015] Levi, D., Garnett, N., Fetaya, E., and Herzlyia, I. (2015). Stixelnet: A deep convolutional network for obstacle detection and road segmentation. In *BMVC*, pages 109–11.
- [17] [Oliveira et al., 2016] Oliveira, G. L., Burgard, W., and Brox, T. (2016). Efficient deep models for monocular road segmentati-

- on. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4885–4891. IEEE.
- [18] [Onunka and Bright, 2010] Onunka, C. and Bright, G. (2010). Autonomous marine craft navigation: On the study of radar obstacle detection. In *ICCAR 2010*, pages 567–572.
- [19] [Reinhard et al., 2001] Reinhard, E., Adhikhmin, M., Gooch, B., and Shirley, P. (2001). Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41.
- [20] [Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.
- [21] [Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer.
- [13] [Yu et al., 2018] Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., and Darrell, T. (2018). Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*.
- [14] [Zhao et al., 2017] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *EEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890.

■

Borja Bovcon je magistriral iz računalništva in matematike na Univerzi v Ljubljani, Fakulteti za matematiko in fiziko ter Univerzi v Ljubljani, Fakulteti za računalništvo in informatiko leta 2017. Trenutno obiskuje doktorski študij na Univerzi v Ljubljani, Fakulteti za računalništvo in informatiko ter dela kot raziskovalec v Laboratoriju za umetne vizualne spoznavne sisteme (LUVSS). Njegovo raziskovalno področje obsega računalniški vid, razpoznavanje vzorcev, detekcijo ovir ter avtonomne sisteme.

■

Matej Kristan, izr. prof. dr., je doktoriral leta 2008 na Univerzi v Ljubljani, Fakulteti za elektrotehniko. Trenutno je član Laboratorija za umetne vizualne spoznavne sisteme (LUVSS) ter izredni profesor na Univerzi v Ljubljani, Fakulteti za računalništvo in informatiko. Njegovo raziskovalno področje obsega verjetnostne metode za računalniški vid s poudarkom na vizualnemu sledenju, dinamičnih modelih, sprotnemu učenju in strojnem vidu za mobilno robotiko.

HYLO: Hibridni logaritmični množilnik za energijsko učinkovito računanje

Ratko Pilipović, Patricio Bulić

University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia

ratko.pilipovic@fri.uni-lj.si, patricio.bulic@fri.uni-lj.si

Izveček

V članku predstavimo hibridni logaritmični množilnik, HYLO, ki združuje Radix-4 Boothovo kodiranje in logaritmično aproksimacijo za generiranje delnih produktov. HYLO tako prinaša prednosti dveh glavnih strategij v domeni približnih množilnikov, rezanje delnih produktov in logaritemsko aproksimacijo množenja. Zaradi manjšega števila delnih produktov, HYLO množilnik ima manjšo porabo toka, zato ga lahko uporabimo v široki paleti aplikacij, ki so imune na računsko napako, brez degradacije natančnosti ali performans.

Ključne besede: Logaritmični množilnik, načrtovanje aritmetičnih vezj, približni množilniki, približno računanje

Abstract

We propose a novel hybrid logarithmic approximate multiplier HYLO, which combines Radix-4 Booth encoding and logarithm product approximation for partial product generation. With this, HYLO brings the advantages of two major strategies: truncated multipliers and logarithm product approximation. Due to the smaller number of partial products, the described multiplier is power-efficient and can be employed in a wide area of error-resilient applications without the decrease in performance or quality.

Keywords: Approximate computing, approximate multiplier, arithmetic circuit design, logarithmic multipliers

1 INTRODUCTION

The explosion of the volume of complex and noisy data [Agrawal et al., 2016], produced by low-power devices and scientific data centres, stretches the capabilities of modern computing platforms and raises concern on power-efficient processing in embedded systems and high-performance computing platforms. Due to the emergence of error-tolerant applications [Jerger and Miguel, 2018], approximate computing surfaced as the solution for achieving power-efficient processing [Mittal, 2016]. Approximate computing allows small inaccuracies in computing in order to achieve more efficient processing. Recent progress in the area of approximate computing [Jerger and Miguel, 2018, Eeckhout, 2018] indicates the popularity of this approach.

Traditional arithmetic circuits are encountering difficulties when it comes to design improvement [Azizi et al., 2010]. This made the approximate computing a popular strategy for arithmetic circuit design [Han and Orshansky, 2013] where energy-efficient design is more important than accuracy and

the challenge is to achieve the best trade-off between accuracy and design efficiency. Multipliers represent complex arithmetic circuits [Parhami, 1999] and at the same time are indispensable components for many systems. Considering this, more efficient design of a multiplier is needed for more energy-efficient processing. With approximate computing techniques, we can achieve efficient multiplier design and improve the performance of error-resilient systems.

Two design paradigms dominate in the field of approximate multipliers: logarithmic and truncated multipliers. Logarithmic multipliers rely on the addition of binary logarithms to approximate multiplication. They employ the property that multiplication can be substituted with addition in the logarithmic domain. Mitchell's multiplier [Mitchell, 1962] represents the first logarithmic multiplier, where the binary logarithm is used to approximate multiplication. In order to improve Mitchell's multiplier design, Babic et al. developed the iterative logarithmic multiplier (ILM) [Babić et al., 2011], that achieves arbitrary accuracy through an iterative procedure.

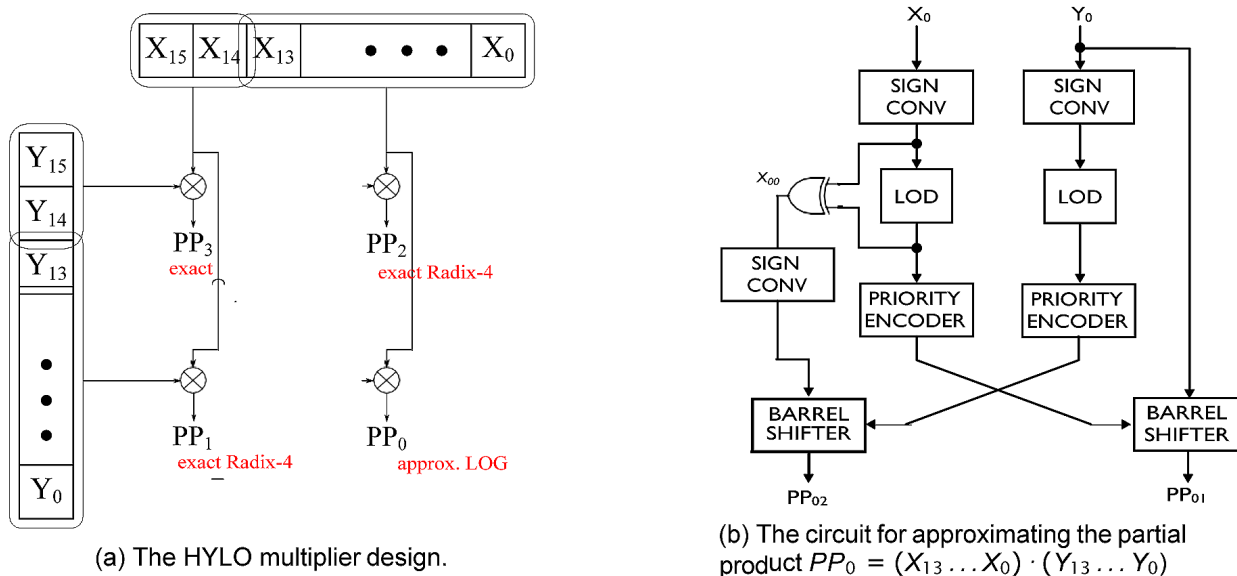


Figure 1: 16-bit HYLO multiplier. (a) The proposed multiplier generates four partial products, PP₃, PP₂, PP₁, and PP₀. The partial products PP₃, PP₂ and PP₁ are exact and simple to generate using the Radix-4 Booth encoding. (b) The partial product PP₀ is generated using the logarithmic multiplier approximation strategy.

Recently, the number rounding approach became a popular approach for logarithmic multiplier design. Zendagani et al. [Zendagani et al., 2017] developed the rounding based approximate (ROBA) multiplier, where the input operands are rounded to the nearest value of the power of two.

The error of the truncated multipliers emerges from simplifying two main stages of a multiplier: the partial product addition and the partial product generation. Simplifications in the partial product addition stage rely on employment of approximate compressors. [Yang et al., 2015]. In order to deliver efficient compressor, Esposito et al. [Esposito et al., 2018] developed XOR-free compressors that achieve smaller error and area usage. Simplifications in PP generation could lead to more efficient design than simplifications in the partial product addition stage. The goal is to produce less partial products and at the same time produce a small error. Following this, Zervakis et al. [Zervakis et al., 2015] introduced the technique for the partial product omission in Booth encoding. The omission of partial products leads to the simpler partial products addition stage but leads to a significant decrease in accuracy.

Previous work in the field of approximate multipliers indicates its importance and necessity for power efficient processing. We can also see that there is no clear winner in this field and that the search for

an efficient approximate multiplier is far from being finished. In this work, we introduce the Hybrid Logarithm (HYLO) multiplier which utilizes the concept of operand decomposition to generate fewer partial products compared to the exact multiplier. HYLO incorporates Radix-4 Booth encoding to encode two partial products from the two most significant bits and a logarithmic multiplier to generate one partial product from the other bits. The proposed multiplier combines advantages from both worlds: simple design from logarithmic multipliers and high accuracy of truncated multipliers.

The rest of paper is organized as follows: Section 2 describes the HYLO product approximation and Section 3 present error and design evaluation of HYLO multiplier. Finally, The Section 4 concludes the paper.

2 THE PROPOSED HYLO MULTIPLIER

HYLO multiplier addresses two important issues. The first issue refers to the design of the efficient partial product (PP) generation stage. By combining Radix-4 Booth encoding [Ercegovic and Lang, 2004] and logarithmic multiplier approximation strategy, HYLO tries to deliver good trade-off between low-power design and small error. The second issue refers to the challenge of increasing the accuracy of logarithmic multiplier without a significant increase in area utilization and power consumption. HYLO multiplier uti-

Table 1: **Synthesis results and MRE.**

Multiplier	Delay (ns)	Power (mW)	Area (μm_2)	Energy (ρJ)	PADP ($mJ \cdot \mu m_2$)	MRE (%)
Radix-4	4.05	1.526	6.23e+04	6.18	0.385	0.0
HYLO	6.43	0.796	5.56e+04	5.11	0.261	4.13

lizes the concept of operand decomposition to address previously mentioned problems. Illustration of HYLO multiplier design is presented in Figure 1a.

Let X and Y be the 16-bit signed numbers in two's complement representation. To obtain the product $X \cdot Y$, both inputs X and Y are divided into two segments: most significant segments (MSS) - X_1, Y_1 and least significant segments (LSS) - X_0, Y_0 . The MSS consists of two leftmost bits while other bits belong to LSS. Then, segments are multiplied with each other and added at the end:

$$X \cdot Y = X_1 \cdot Y_1 \cdot 2^{28} + (X_1 \cdot Y_0 + X_0 \cdot Y_1) \cdot 2^{14} + X_0 \cdot Y_0 \quad (1)$$

The circuit for generation of partial product $X_1 Y_1$ is implemented through employment of 2-bit signed multiplier. For designing 2-bit signed multiplier we employed Carnaugh's minimization procedure. Partial products $X_1 \cdot Y_0$ and $Y_1 \cdot X_0$ are obtained through employment of Radix-4 Booth encoding:

$$X_1 \cdot Y_0 = \hat{x}^{R4} \cdot Y_0 \quad \text{and} \quad Y_1 \cdot X_0 = \hat{y}^{R4} \cdot X_0 \quad (2)$$

where:

$$x^{R4} = -2 \cdot x_{15} + x_{14} + x_{13} \quad \text{and} \quad y^{R4} = -2 \cdot y_{15} + y_{14} + y_{13} \quad (3)$$

The partial products $X_1 Y_0$ and $Y_1 Y_0$ are easy to generate without multiplication due to the use of radix-4 Booth encoding. The only multiplication in Eq. (1) is required by the term $X_0 Y_0$. Note that X_0 and Y_0 are 14-bit signed integer numbers. According to study [Babić et al., 2011], we approximate partial product $X_0 \cdot Y_0$ as:

$$X_0 \cdot Y_0 \approx \text{sign}(X_0)Y_0 2^{k_{x_0}} + \text{sign}(Y_0)X_0 2^{k_{y_0}} = PP_{01} + PP_{02} \quad (4)$$

where $k_{y_0} = \lfloor \log_2 |Y_0| \rfloor$, $k_{x_0} = \lfloor \log_2 |X_0| \rfloor$, $X_{00} = |X_0| - 2^{k_{x_0}}$ and $\text{sign}(X_0 \cdot Y_0)$ denotes the sign of product $X_0 \cdot Y_0$.

The circuit for partial product generation is shown in Figure 1b. Sign conversion blocks (SIGN CONV) are employed to calculate the absolute value of input operands and sign of partial products. Next, Leading-one-detectors (LOD) together with Priority encoders calculate the k_{x_0} and k_{y_0} . In the end, Barrel shifters produce partial products. Finally, all partial products are added by employing the Wallace tree methodology [Wallace, 1964]. For the final addition of partial products, we employ Carry-Look-Ahead adder [Ercegovic and Lang, 2004].

3 RESULTS AND DISCUSSION

This section presents the evaluation of proposed multiplier in terms of hardware (power, area, delay, energy and power-area-delay product (PADP)) and mean relative error (MRE). We have compared HYLO with an accurate radix-4 multiplier. The multipliers have been implemented in Verilog and synthesized using TSMC 180-nm standard cell library. After place-and-route, the cell area and critical path delay were reported. The power consumption is estimated after the layout extraction using the IRSIM simulator with randomly generated 10000 input test vectors. Table 1 contains the synthesis results (delay, power, area, energy and power-area-delay product) as well as MRE for both multipliers.

From Table 1 we can see that HYLO offers almost 50 % savings in power, and 10 % savings in chip area, but it has almost 50 % larger critical path delay. Nevertheless, it delivers 17 % savings in energy and almost 33 % in power-area-delay product (PADP), while keeping small MRE.

4 CONCLUSION

In this paper, we propose a design strategy for approximate multipliers, which combines the advantages of truncated and logarithmic multipliers. On the top level, the architecture of HYLO multiplier resembles the design of truncated multiplier. It employs operand decomposition and partial product addition to generate the product. On the other hand,

HYLO approximates partial product $X_0 \cdot Y_0$ using binary logarithms. This makes it similar to logarithmic multiplier. Because of this, HYLO represents a hybrid multiplier, which combines elements from truncated and logarithmic multipliers. This allows it to achieve a good trade-off between accuracy and design efficiency.

REFERENCES

- [1] [Agrawal et al., 2016] Agrawal, A., Choi, J., Gopalakrishnan, K., Gupta, S., Nair, R., Oh, J., Prener, D. A., Shukla, S., Srinivasan, V., and Sura, Z. (2016). Approximate computing: Challenges and opportunities. In *2016 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8.
- [2] [Azizi et al., 2010] Azizi, O., Mahesri, A., Lee, B. C., Patel, S. J., and Horowitz, M. (2010). Energy-performance tradeoffs in processor architecture and circuit design: A marginal cost analysis. *SIGARCH Comput. Archit. News*, 38(3):26–36.
- [3] [Babić et al., 2011] Babić, Z., Avramović, A., and Bulić, P. (2011). An iterative logarithmic multiplier. *Microprocessors and Microsystems*, 35(1):23–33.
- [4] [Eeckhout, 2018] Eeckhout, L. (2018). Approximate computing, intelligent computing. *IEEE Micro*, 38(4):6–7. [Ercegovac and Lang, 2004] Ercegovac, M. D. and Lang, T. (2004). *Digital arithmetic*. Elsevier.
- [5] [Esposito et al., 2018] Esposito, D., Stollo, A. G. M., Napoli, E., Caro, D. D., and Petra, N. (2018). Approximate multipliers based on new approximate compressors. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 1–14.
- [6] [Han and Orshansky, 2013] Han, J. and Orshansky, M. (2013). Approximate computing: An emerging paradigm for energy-efficient design. In *Test Symposium (ETS), 2013 18th IEEE European*, pages 1–6. IEEE.
- [7] [Jerger and Miguel, 2018] Jerger, N. E. and Miguel, J. S. (2018). Approximate computing. *IEEE Micro*, 38(4):8–10.
- [8] [Mitchell, 1962] Mitchell, J. N. (1962). Computer multiplication and division using binary logarithms. *IRE Transactions on Electronic Computers*, (4):512–517.
- [9] [Mittal, 2016] Mittal, S. (2016). A survey of techniques for approximate computing. *ACM Computing Surveys (CSUR)*, 48(4):62.
- [10] [Parhami, 1999] Parhami, B. (1999). *Computer arithmetic*, volume 20. Oxford university press.
- [11] [Wallace, 1964] Wallace, C. S. (1964). A suggestion for a fast multiplier. *IEEE Transactions on electronic Computers*, (1):14–17.
- [12] [Yang et al., 2015] Yang, Z., Han, J., and Lombardi, F. (2015). Approximate compressors for error-resilient multiplier design. In *2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, pages 183–186. IEEE.
- [13] [Zendegani et al., 2017] Zendegani, R., Kamal, M., Bahadori, M., Afzali-Kusha, A., and Pedram, M. (2017). Roba multiplier: A rounding-based approximate multiplier for high-speed yet energy-efficient digital signal processing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(2):393–401.
- [14] [Zervakis et al., 2015] Zervakis, G., Xydis, S., Tsoumanis, K., Soudris, D., and Pekmestzi, K. (2015). Hybrid approximate multiplier architectures for improved power-accuracy trade-offs. In *Low Power Electronics and Design (ISLPED), 2015 IEEE/ACM International Symposium on*, pages 79–84. IEEE.
- [15] *Ratko Pilipović* received his B.Sc. and M.Sc. degrees from the Faculty of Electrical Engineering, University in Banjaluka, Bosnia and Hercegovina in 2015 and 2017, respectively. He is currently working towards the Ph.D. degree at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. His research interests include approximate computing, arithmetic circuit design, FPGA design, embedded processing and machine vision.
- [16] *Patricio Bulić* received his B.Sc. degree in electrical engineering, and M.Sc. and Ph.D. degrees in computer science from the University of Ljubljana, Slovenia, in 1998, 2001 and 2004, respectively. He is an Associate Professor at the Faculty of Computer and Information Science, University of Ljubljana. His main research interests include computer architecture, digital design, approximate computing, computer arithmetics and parallel processing.

Analiza digitalne preobrazbe Slovenije v EU okolju

Cene Bavec
Univerza na Primorskem
cene.bavec@guest.arnes.si

Izveleček

Raziskava je umestila digitalno preobrazbo Slovenije v širši kontekst Evropske unije in izbranih ekonomskih kazalcev, ki opredeljujejo okolje, v katerem poteka digitalizacija. Pokazalo se je, da se ocena posameznih članic EU, ki jo kaže indeks DESI, lahko opazno spremeni, če upoštevamo tudi ekonomsko okolje. Podrobnejša primerjava Slovenije, Finske, Estonije in Češke je pokazala, da se stanje na posameznih področjih zelo razlikuje od države do države. Slovenija ima nekatere izrazite prednosti, kot so visoka vlaganje gospodarstva v raziskave in razvoj ali visoka uvrstitev po inovacijskem indeksu. Po drugi strani pa imamo tudi precej šibkih točk, kot je, na primer, zaostajanje pri digitalizaciji javne uprave in uporabi interneta. Ob vseh statističnih kazalcih pa obstaja še vrsta drugih družbeno in politično obarvanih razlogov, ki lahko bistveno vplivajo na digitalno preobrazbo države, ki so prisotni v uspešnejših državah, kot je Estonija.

Ključne besede: DESI, digitalna preobrazba, ekonomski kazalci, EU okolje, Slovenija 4.0

Abstract

The research focused on the digital transformation of Slovenia in the wider context of the EU and the selected economic indicators that define the environment in which digitization takes place. We have confirmed that assessment of individual EU members, as indicated by the DESI index, can be noticeably changed if we consider the economic environment. We compared Slovenia, Finland, Estonia and the Czech Republic in detail. This part of the analysis revealed that the situation in different areas varies significantly between countries, meaning that the perception of digitization must be multi-dimensional. Slovenia has certain advantages, like high investments of companies in research and development, resulting in a high score on the Innovation index. However, it also has a number of weak points. Slovenia lags behind in the digitization of public administration and the use of Internet. However, there are also many other socially and politically coloured reasons which can significantly affect the digital transformation of the country. We can notice this phenomenon in more digitally advanced countries, such as Estonia.

Keywords: DESI, digital transformation, economic indicators, EU, Slovenia 4.0

1 UVOD

Indeks digitalnega gospodarstva in družbe - DESI je trenutno najpodrobnejša analiza digitalne preobrazbe evropskih držav, ki metodološko omogoča medsebojno primerjavo (DESI 2018). Po eni strani je namenjen evropskim institucijam, ki ga uporabljajo pri načrtovanju in vrednotenju evropske digitalne strategije, po drugi strani pa omogoča posameznim državam, da z medsebojnim primerjanjem ocenjujejo prednosti, slabosti in učinke svojih nacionalnih strategij (MJU, 2017). Vendar indeks DESI prikazuje stanje na ožjem področju digitalizacije, le malo pa pojasnjuje, v kakšnih okoliščinah in zakaj je stanje tako.

Za širše razumevanje digitalne preobrazbe moramo videti tudi širši družbeni in ekonomski kontekst v katerem ta preobrazba poteka. Le tako bomo lahko razumeli, zakaj so nekatera okolja izjemno uspešna ali neuspešna, čeprav na prvi pogled za to ni nobenih izrazitih tehnoloških ali ekonomskih razlogov. Taka, nekoliko nejasna slika je značilna tudi za Slovenijo.

Seveda je vsaka analiza v širokem kontekstu metodološko in podatkovno zahtevna, saj spada v kategorijo kompleksnih primerjalnih analiz (benchmarking), ki slonijo na večjem številu parametrov, ki morajo biti metodološko usklajeni, kar je praviloma zelo težko (Kotarba, 2017). Drug problem s kompleksnimi

analizami pa je, da je kljub velikemu vložku njihova uporabnost časovno omejena, saj se posamezni parametri lahko zelo hitro spreminjajo. Tudi v Sloveniji je bilo narejenih precej podobnih analiz na posameznih področjih in posebej izbranih primerjalnih državah (Jaklič, Zagoršek, 2005), vendar trenutno ni nobene, ki bi bila posebej primerna za širše ocenjevanje naše digitalne preobrazbe.

V prispevku je na nekoliko poenostavljena način ocenjen položaj Slovenije na področju digitalne preobrazbe. V prvi fazi raziskave so članice EU grupirane v pet dimenzionalnem prostoru, ki poleg indeksa DESI upošteva še BDP, inovativnost, konkurenčnost in sposobnost inoviranja gospodarstva v posamezni državi. Tako je bil predstavljen širši vpogled v stanje na področju digitalne preobrazbe, ki presega eno dimenzionalen indeks DESI. V drugi fazi pa je bila na osnovi omenjenih kazalcev narejena še podrobnejša primerjava s tremi izbranimi državami: Finsko, Estonijo in Češko.

2 METODOLOŠKO IZHODIŠČE RAZISKAVE

Indeks DESI je, kljub svoji kompleksni zgradbi (EC, 2017, DESI, 2018) enodimenzionalen pogled na digitalizacijo evropskih držav, ki omogoča njihovo enostavno razvrstitev in vsakoletno ugotavljanje, kako se je njihov vrstni red spremenil. Na tej osnovi se spremlja tudi relativen napredek ali zaostanek držav in s tem uspešnost nacionalnih politik. Za marsikatero namene pa je tak pogled preveč poenostavljen, čeprav je zelo nazoren in lahko razumljiv. V ozadju so vprašanja in dileme, na katere je težko objektivno odgovoriti. Na kaj vse vpliva digitalizacija in kaj vpliva nanjo (Hegyes, Csapó, Farkas, 2017, Karnitis, Virtmanis, Karnitis, 2019)? Korelacije med različnimi tehnološkimi, ekonomskimi in širšimi družbenimi kazalci so več kot očitne, še zdaleč pa ni jasno, kaj so vzroki in kaj posledice.

Kot primer vzemimo samo korelacijo med kazalcema DESI in BDP na prebivalca. Vsi predpostavljamo, da digitalizacija pozitivno vpliva na ekonomsko rast, vendar iz medsebojne korelacije ne moremo ugotoviti, ali je to res in kakšen je ta vpliv. Korelacija namreč lahko nakazuje tudi to, da visoka ekonomska razvitost s spremljajočo družbeno infrastrukturo in sposobnostjo večjih investicij že sama po sebi spodbuja digitalizacijo. V bistvu gre za povratno zanko. Predvidoma bi bil boljši kazalec rast družbenega proizvoda, toda preliminarna analiza v okviru raz-

iskave je pokazala, da med indeksom DESI in rastjo družbenega proizvoda ni nobene korelacije, kar podobne interpretacije in vprašanja samo zaplete.

Večja podjetja, posebej bolj tehnološko usmerjena, lahko razmeroma natančno ocenijo pomen digitalne preobrazbe od vpliva vlaganj v digitalizacijo na učinkovitost podjetja do kadrovske problematike in podobno (Stäbler, 2016). Vendar so njihov pristop in uporabljene metodologije, predvsem izbor kazalcev, neprenosljivi na raven celotne ekonomije ali družbe. Družba ima bistveno bolj zapleteno zgradbo in s tem bolj zapletene vplive in povratne zanke povezane z digitalno preobrazbo (Benner, 2017, Mičić, 2017, Traščā at. al, 2019).

Zato je prva faza raziskave poleg indeksa DESI upoštevala tudi druge kazalce, ki opredeljujejo okoliščine, v katerih poteka digitalna preobrazba. Na osnovi preliminarne analize in dostopnosti sekundarnih virov podatkov ter na osnovi izkušenj so se pri ocenjevanju stanja na področju digitalne preobrazbe upoštevali naslednji kazalci:

1. **DESI** (leto 2018) je še vedno osnovni pokazatelj stanja na področju digitalne preobrazbe (vir podatkov je Evropska komisija).
2. **BDP** na prebivalca v evrih (leto 2018) je izbran ob predpostavki, da ekonomsko močnejša okolja lažje zagotovijo absolutno večja vlaganja v digitalno preobrazbo (vir podatkov je Evrostat).
3. **KONK** je konkurenčnost države, ki je lahko po eni strani posledica digitalizacije, po drugi strani pa predstavlja pritisk za še hitrejšo digitalno preobrazbo (vir podatkov je IMD World Competitiveness Yearbook 2018)
4. **INOV** je inovacijski indeks držav, ki kaže njihovo raziskovalno in inovacijsko sposobnost, kar neposredno vpliva tudi na digitalizacijo (vir podatkov je European innovation scoreboard 2018).
5. **INGOSP** je kazalec sposobnosti inoviranja v gospodarstvu, ki je eden od pogojev za njegovo digitalno preobrazbo in prehod v industrijo 4.0 (vir podatkov je IMD World Competitiveness Yearbook 2018, stran 478).
6. **RDGOSP** kaže na neposredne investicije gospodarstva v raziskave in razvoj (%BDP). Ker so te investicije posredno upoštewane v prej omenjenih kazalcih, se je ta podatek upošteval le pri kvalitativnih ocenah in ne v statističnih obdelavah (vir podatkov je IMD World Competitiveness Yearbook 2018, stran 468).

Tabela 1 kaže, da so vsi omenjeni kazalci v visoki korelaciji z indeksom DESI, kar opravičuje njihov izbor. Po drugi strani pa predstavlja manjšo zadrego s statističnega zornega kota, ker so vsi kazalci tudi v medsebojni korelaciji. Ko jih jemljemo kot samostojne kazalce, moramo biti nekoliko previdni pri interpretaciji rezultatov. Vendar je to značilno za skoraj vse ekonomske in tehnološke kazalce, kar kaže na vsebinsko prepletenost in soodvisnost celotnega področja. Tretja značilnost vseh teh kazalcev pa je, da je skoraj nemogoče objektivno ugotoviti, kaj je vzrok in kaj posledica, zato lahko govorimo samo o medsebojni soodvisnosti.

Tabela 1: **Korelacije med indeksom DESI in izbranimi kazalci za članice EU v letu 2018**

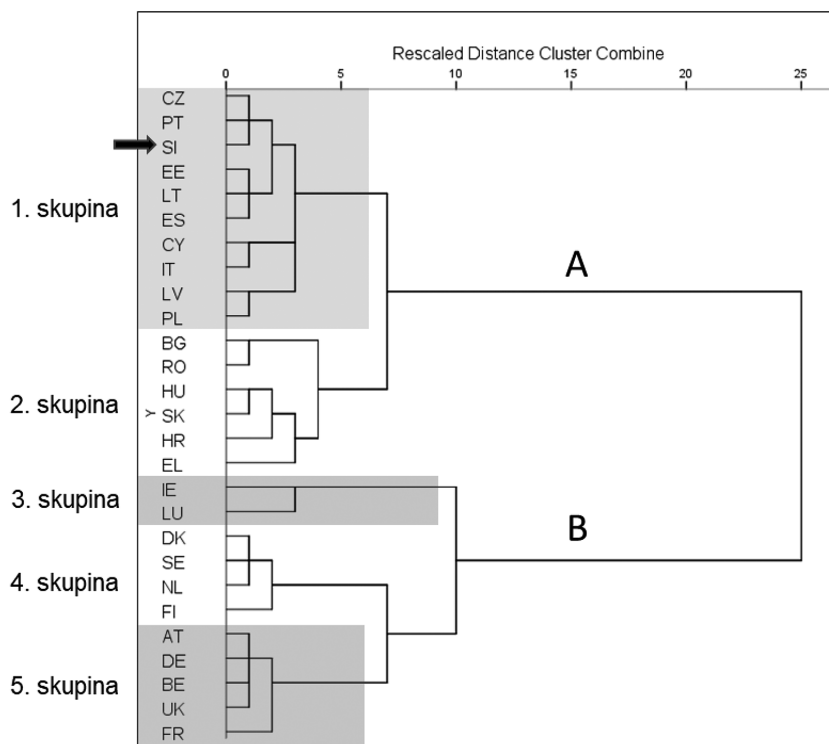
	BDP na prebivalca	INOV Inovativnost države	KONK Konkurenčnost države	INGOSP Inoviranje v gospodarstvu	RDGOSP Vlaganja gosp. v RR
DESI	0,681**	0,872**	0,871**	0,760**	0,623**

** Vse korelacije so statistično značilne $p < 0,01$

V preliminarni fazi raziskave je bila med kazalce uvrščena tudi letna rast BDP, saj naj bi digitalizacija neposredno pospeševala ekonomski razvoj. Vendar se je pokazalo, da rast BDP ni v korelaciji niti z indeksom DESI niti z omenjeni kazalci, zato se kasneje v statističnih obdelavah ni uporabljala.

3 SLOVENIJA V EVROPSKEM OKOLJU

S hierarhično klastersko analizo (razvrščanje v skupine) so bile vse države EU (razen Malte) grupirane v skupine tako, da so pripadniki ene skupine bolj podobni eni drugim, kot pa pripadnikom druge skupine. Vsaka država je bila opisana s petimi kazalci: BDP, DESI, INOV, KONK in INGOSP, kar pomeni, da je grupiranje potekalo v pet dimenzionalnem prostoru. Ker so bili kazalci opredeljeni z različnimi enotami in v različnih intervalih, so bili normalizirani tako, da je bila povprečna vrednost vsakega kazalca 0, njegova standardna deviacija pa 1 (z-score). Na ta način je bila podobnost dveh držav opredeljena kot kvadrat medsebojne razdalje v evklidskem prostoru. Hierarhična klasterska analiza je bila izbrana predvsem zato, ker je njene rezultate lahko interpretirati in jih tudi vizualno predstaviti (Slika 1).



Slika 1: **Dendrogram hierarhične klasterske analize v 5 dimenzionalnem prostoru: BDP, DESI, INOV, KONK in INGOSP**
(Vir: lastna raziskava na osnovi sekundarnih podatkov)

Dendrogram na sliki 1 postavlja digitalno preobrazbo v nekoliko drugačno luč kot jo kaže samo indeks DESI, saj razkriva naslednje značilnosti članic EU:

- Države se na najvišji hierarhični ravni grupirajo v dve izraziti grupi A in B. Grupo A sestavljajo pripadnice 1. in 2. skupine, v katerih so brez izjeme vse nove članice EU, od starih pa vse mediteranske države Italija, Španija, Portugalska in Grčija.
- Grupo B tvorijo države iz 3., 4. in 5. skupine, v katerih pa so samo stare članice EU. Tako grupiranje kaže na temeljne razlike na področju digitalizacije med srednje in severno evropskimi starimi članicami EU na eni strani ter novimi članicami in starimi mediteranskimi članicami na drugi. Delitev je tako izrazita, da odpira vrsto vprašanj o nadaljnji digitalizaciji EU kot celote.
- Ko pogledamo podrobnejšo strukturo prve od teh grup (A), vidimo, da je prva skupina medsebojno zelo podobnih držav največja in vsebuje 10 članic. V njej je tudi Slovenija. Če pogledamo še podrobnejšo razporeditev držav v 1. skupini, pa vidimo, da sta Sloveniji najbolj podobni Češka in Portugalska, kar je enaku rezultatu, kot ga kaže indeks DESI. Zelo podobne pa so ji tudi Estonija, Latvija in Španija. Nekoliko presenetljivo je v tej skupini Estonija, ki je sicer na DESI lestvici bistveno višje.
- Drugo skupino sestavljajo pretežno balkanske države ter Grčija in Slovaška, ki se značilno razlikujejo od držav iz prve skupine. Vidi se tudi to, da sta Romunija in Bolgarija posebna podskupina.
- Zanimiva je 3. skupina, ki jo sestavljata Irska in Luksemburg, saj opazno izstopa od ostalih severnoevropskih držav. Eden od razlogov je njun izjemno visok družbeni proizvod na prebivalca.
- Četrto skupino sestavljajo evropski šampioni na področju digitalizacije Danska, Švedska Finska in Nizozemska. Te štiri članice so tudi prve na DESI listi. Dendrogram kaže, da se močno razlikujejo od vseh drugih članic iz 3. in 5. skupine.
- Peta skupina pa združuje ekonomsko močne srednjeevropske države Avstrijo, Nemčijo, Belgijo, Francijo in Združeno kraljestvo.

Iz dendrograma bi lahko razbrali še nekatere druge podrobnosti, vendar so omenjene značilnosti dovolj za grobo analizo digitalizacije evropskih držav in pogojev v katerih poteka. Ob prej omenjenih predpostavkah imajo države z visokim družbenim

proizdomom že v osnovi prednost pred tistimi z opazno nižjo ekonomsko razvitostjo. Zato sta dve baltski državi, ki sta po indeksu DESI razmeroma visoko, padli v skupino, v kateri je tudi Slovenija. Sprememba je najbolj vidna pri Estoniji, ki je po opisani klasifikaciji uvrščena precej nižje kot po DESI.

Da bi lažje interpretirali omenjene rezultate, se moramo spustiti v podrobnejše primerjave na ravni posameznih kazalcev.

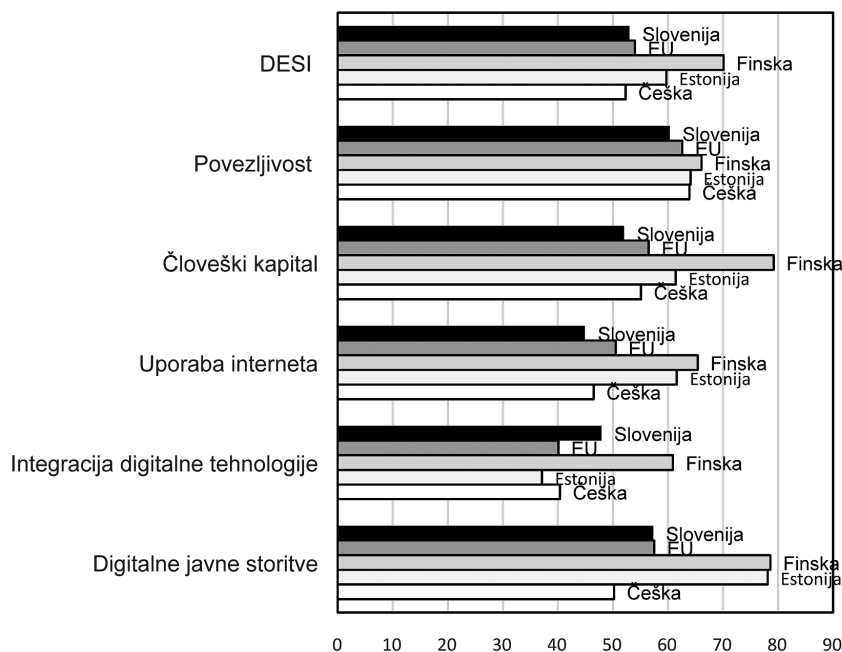
4 PRIMERJAVA SLOVENIJE Z IZBRANIMI DRŽAVAMI

Naslednji korak v raziskav je bila primerjava Slovenije s Finsko, Estonijo in Češko ter iskanje razlik na ravni posameznih kazalcev, ki bi lahko pojasnile prednosti in slabosti posamezne države, predvsem z zornega kota Slovenije. V prvem koraku so bili upoštevani le DESI kazalci, v drugem koraku pa kazalci, ki so bili uporabljeni pri klasterski analizi EU držav. Indeks DESI sestavlja pet osnovnih agregiranih indeksov:

1. Povezljivost - opredeljuje pretežno tehnološke in uporabniške vidike komunikacij,
2. Človeški kapital – opredeljuje digitalna znanja in spretnosti tudi v širši javnosti,
3. Uporaba interneta – opredeljuje predvsem osebno uporabo interneta,
4. Integracija digitalne tehnologije – predstavlja e-poslovanje in e-trgovanje,
5. Digitalne javne storitve – predstavlja storitve e-uprave.

Slika 2 prikazuje vrednost agregiranih indeksov DESI za EU in izbrane države. Po vseh kriterijih izstopa Finska, ki je bila v tej primerjavi uporabljena kot ciljna referenčna država. Razlog je v tem, da je Finska v samem evropskem in tudi svetovnem vrhu na področju digitalizacije, poleg tega pa spada med manjše države, ki so vsaj delno primerljive s Slovenijo. Druga posebej zanimiva država je Estonija, ki je po indeksu DESI precej pred nami, poleg tega pa je po velikosti (okoli 1,3 milijona prebivalcev) primerljiva s Slovenijo. Tretja država je Češka, ki je v bistvu naša najbolj neposredna ekonomska tekmica in nam je na področju digitalizacije tudi najbližja.

Povzemimo samo najpomembnejše rezultate teh primerjav:



Slika 2: Primerjava posameznih komponent DESI indeksa po izbranih državah (Vir: DESI 2018)

- Slovenija je po vseh kazalcih rahlo pod EU povprečjem z izjemo Integracije digitalne tehnologije (v bistvu digitalizacije gospodarstva), kjer je le za Finsko. Na področju povezljivosti in človeškega kapitala pa je celo zadnja med primerjanimi državami.
- Estonija ima izrazito prednost pred Slovenijo le pri uporabi interneta in digitalnih javnih storitvah. Ti dve področji sta tudi paradna konja estonske digitalizacije, ki jo bistveno dvigujeta na DESI lestvici. Zanimivo in na določen način presenetljivo pa je, da je Estonija zadnja po integraciji digitalne tehnologije v gospodarstvu, kar samo potrjuje dileme okoli neposrednega vpliva digitalizacije na ekonomski razvoj in obratno.
- Češka je pred Slovenijo le po človeškem kapitalu in uporabi interneta, toda so razlike minimalne in so v območju statističnih napak.

Končni sklep je, da je v bližnji prihodnosti Finska neulovljiva, Estoniji pa se bomo lahko približali, ko si bo opomogla naša e-uprava, ki je nekoč že bila po stopnji digitalizacije pred njo. Nekoliko bomo morali tudi dvigniti uporabo interneta na osebni ravni. Vse pa kaže, da bomo s Češko še nekaj časa neposredno tekmovali.

V naslednji fazi raziskave je bila opravljena primerjava med temi državami še po kriterijih, ki so bili

uporabljeni pri klasterski analizi. Dodan je bil le še kazalec vlaganja gospodarstva v raziskave in razvoj, ki vsaj na gospodarski ravni delno pojasnjuje, koliko je gospodarstvo pripravljeno in sposobno vlagati tudi v digitalizacijo (Tabela 2).

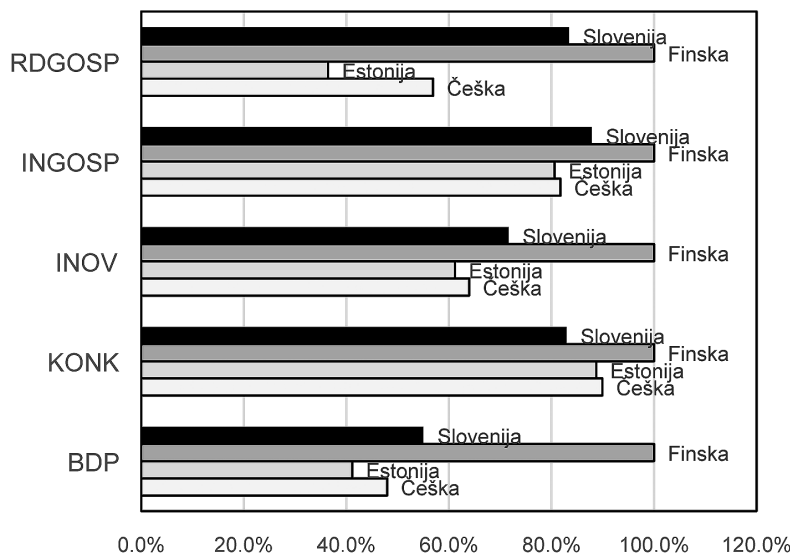
Tabela 2: Primerjalni kazalci v realnih vrednostih

	BDP na preb.	KONK	INOV	INGOSP	RDGOSP v % BDP
Češka	17600	79,507	0,415	5,73	1,03
Estonija	15100	78,475	0,397	5,65	0,66
Finska	36700	88,42	0,649	7,01	1,81
Slovenija	20200	73,346	0,465	6,16	1,51

Ker imajo posamezni kazalci različne merske enote, so bili zaradi grafične primerljivosti in nazornosti normirani tako, da je bila Finska vedno 100%, ker je po vseh kriterijih prva, ostale države pa sorazmerno manj. Realne enote so prikazane v tabeli 2 (opis kazalcev je podan v 2. poglavju).

Primerjava držav po ekonomskih kazalcih (Slika 3), ki opredeljujejo pogoje, v katerih poteka digitalizacija, kaže:

- Finska je po vseh kriterijih najuspešnejša, kar pomeni, da je njena digitalna preobrazba verjetno dolgoročno stabilna in vzdržna. Poleg njene izrazito višje ekonomske razvitosti (BDP) je tudi daleč



Slika 3: Primerjava izbranih držav po petih kazalcih (Vir: Eurostat in IMD World Competitiveness Yearbook 2018)

najvišjo raziskovalno in inovacijsko sposobnost (INOV).

- Slovenija je po teh kazalcih druga najuspešnejša država, razen po kriteriju konkurenčnosti, kjer smo zadnji, vendar te razlike niso izrazite. Naša posebnost pa je vlaganje gospodarstva v raziskave in razvoj, kjer bistveno presegamo Estonijo in tudi Češko. Ta vlaganja so na področju digitalizacije trenutno daleč največja prednost in priložnost Slovenije. Opazno je neravnotežje med inovativnostjo gospodarstva in javnega sektorja, ki v našem primeru ni vezano samo na financiranje (Bavec, 2011).
- Estonija je na področju digitalizacije po indeksu DESI veliko močnejša, kot bi sklepali samo na osnovi prikazanih kazalcev in kaže, da se lahko uspešno digitalizirajo tudi države z nižjo ekonomsko razvitostjo.
- Češka je tudi po teh kriterijih največja konkurentka Sloveniji tako na področju digitalizacije kot ekonomskega razvoja nasploh.

Pri interpretaciji statističnih podatkov kaže posebej omeniti, da se Slovenija po večini tehnoloških in ekonomskih kriterijev v zadnjih letih hitreje razvija kot je bilo EU povprečje in tudi od mnogih neposrednih tekmič s podobnim BDP na prebivalca. Vendar jo je pretekla ekonomska kriza, vsaj statistično gledano, bolj prizadela kot večino drugih, zato je njen trenutni hiter napredek lahko le lovljenje prejšnjih pozicij in ni nujno vzdržan. Vendar je to samo predpostav-

ka, ki pa jo kljub temu moramo imeti v mislih. Tudi na DESI lestvici se Slovenija dviguje in bo verjetno že naslednje leto preseгла EU povprečje, še posebej zato, ker si je delno opomogla tudi javna uprava, ki nam trenutno znižuje ta indeks.

5 SKLEP

V raziskavi je bilo ocenjeno stanje na področju digitalizacije Slovenije s širšega zornega kota, kjer so bile poleg indeksa DESI, ki sloni na pretežno tehnoloških in uporabniških kazalcih, upoštevne tudi določene ekonomske okoliščine, v katerih poteka digitalizacija. Izbor teh kazalcev je bil v veliki meri intuitiven in zagotovo ni zajel vseh ključnih vidikov, zato je potrebno temu primerno interpretirati tudi rezultate. Ne glede na te omejitve pa je kratka raziskava osvetlila nekatere razloge za večjo ali manjšo uspešnost držav. Ugotovljeno je bilo, da se razporeditev posameznih članic EU, ki jo kaže DESI lestvica, nekoliko spremeni, če upoštevamo tudi ekonomsko okolje. Vendar te spremembe razen v primeru Estonije in v manjši meri Litve niso zelo velike. Predvsem pridejo do večjega izraza razlike med ekonomsko razvitejšimi in manj razvitimi članicami EU ter severom in jugom Evrope.

Podrobnejša primerjava Slovenije, Finske, Estonije in Češke je pokazala, da se stanje na posameznih področjih lahko bistveno razlikuje od države do države. Nekatere imajo prednost na enih, druge pa na drugih področjih, zato mora biti pogled na digitalizacijo večdimenzionalen. Tudi Slovenija ima nekate-

re izrazite prednosti, ki bi jih morala bolje izrabiti, kot so, na primer, naša visoka vlaganje gospodarstva v raziskave in razvoj ter raziskovalna in inovacijska sposobnost. Po drugi strani pa imamo tudi precej šibkih točk, kot je, na primer, zaostajanje pri digitalizaciji uprave in uporabi interneta. Tudi to so področja, kjer bi lahko z usmerjenimi akcijami razmeroma hitro nadomestili zaostanek.

Nekoliko nenavadni so slovenski problemi s človeškim kapitalom. Slovenski izobraževalni sistem še zdaleč ni slab, vendar nam je rahlo upadlo število študentov na STEM študijih (naravoslovne znanosti, tehnologija, inženirstvo in matematika) (Cerinšek in ostali, 2013). Verjetno je del problema tudi organizacija naših univerz, ki ni posebej naklonjena multidisciplinarnim študijem, ki pa so vedno pomembnejši za digitalno preobrazbo. Čeprav je nekoliko vprašljiva predpostavka v indeksu DESI, da manjša uporaba interneta med prebivalstvom pomeni tudi probleme z znanjem, je dejstvo, da po tem kriteriju bistveno zaostajamo, še posebej za skandinavskimi državami. Uporaba interneta pa je med drugim odvisna tudi od ponudbe, kar nas zopet pripelje do problemov z digitalizacijo uprave in javnega sektorja, ki je eden od pomembnejših ponudnikov digitalnih storitev za občane. To je eno od področij, kjer se vedno težje primerjamo s skandinavskimi in celo baltskimi državami.

Čeprav so bili pri primerjavah ob indeksu DESI upoštevani tudi ekonomski kazalci, pa obstaja še vrsta drugih družbeno in politično obarvanih razlogov, ki lahko bistveno vplivajo na digitalno preobrazbo države. Kot primer omenimo samo Estonijo. Eden od pomembnih razlogov za njeno uspešnost leži izven tehnologije in ekonomije in je izrazito političen. Estonija, Finska in v veliki meri tudi Švedska namreč medsebojno sodelujejo na državni ravni veliko tesneje, kot je to značilno za druge države in tudi EU kot celoto. Državne agencije si pogosto izmenjujejo ne samo izkušnje ampak tudi operativne rešitve. Nekje prihaja celo do izenačevanja internih standardov in postopkov, kot je primer povezovanja Finske in Estonije na področju zdravstva (Estonia and Finland lead the way, 2019). Predvsem Estonija ima od tega ogromne koristi, ki jih ne zajema noben tehnološki ali ekonomski kazalec. Tako sodelovanje je v slovenski mednarodni okolici praktično nemogoče.

Raziskava je pokazala, da Slovenija nima hujših ovir na poti digitalne preobrazbe in bo v naslednjih letih brez dvoma še napredovala. Posebej spodbudna je digitalizacija v gospodarstvu. Vendar podrobnejše spremljanje trendov v ostalih članicah EU kaže, da naglo napredujejo tudi srednjeevropske in baltske države. Zato bi kazalo zelo hitro opredeliti prednostna področja digitalizacije, kjer bi bila vlaganja najbolj učinkovita. Taki področji sta brez dvoma tudi javna uprava in razne storitve za občane.

LITERATURA

- (1) Bavec, C. (2011). Analysis of the Innovative Climate in the Private and the Public Sector in Slovenia. *Management*, 6(4), 351-364.
- (2) Benner, E. (2017). Cultural Acceptance of Digitalization and Growth of an Economy: A Comparison of East and West Germany.
- (3) Cerinšek, G., Hribar, T., Glodež, N., & Dolinšek, S. (2013). Which are my future career priorities and what influenced my choice of studying science, technology, engineering or mathematics? Some insights on educational choice—case of Slovenia. *International Journal of Science Education*, 35(17), 2999-3025.
- (4) EC. (2017). European Digital Progress Report: review of Member States' progress towards digital priorities
- (5) Estonia and Finland lead the way. (31. januar 2019). Digital Health Society. Pridobljeno s <https://thedigitalhealthsociety.com/estonia-and-finland-lead-the-way/>
- (6) European innovation scoreboard 2018. pridobljeno s <https://ec.europa.eu/docsroom/documents/33147/attachments/1/translations/en/renditions/native>
- (7) Hegyes, É. G., Csapó, I., & Farkas, M. F. (2017). Some aspects of digitalization and sustainability in the European Union. *Journal of Management*, 36(2), 37-46.
- (8) IMD World Competitiveness Yearbook 2018. (2018). IMD: Institute for Management Development, Lausanne
- (9) Jaklič, M., & Zagoršek, H. (2005). Benchmarking countries: comparing competitiveness of Finland, Ireland and Slovenia. *Ekonomska fakulteta*.
- (10) Karnitis, G., Virtmanis, A., & Karnitis, E. (2019). Key Drivers of Digitalization; EU Context and Baltic Case. *Baltic Journal of Modern Computing*, 7(1), 70-85.
- (11) Kotarba, M. (2017). Measuring digitalization—key metrics. *Foundations of Management*, 9(1), 123-138.
- (12) Mičić, L. (2017). Digital Transformation and Its Influence on GDP. *Economics*, 5(2), 135-147.
- (13) MJU. (2017). Digitalna preobrazba Slovenije, Ljubljana.
- (14) Stäbler, A. (2016). The imperative to develop digitalization patterns for enterprises. *Digital Enterprise Computing (DEC 2016)*.
- (15) The Digital Economy and Society Index – DESI 2018. (2019). Pridobljeno s <https://ec.europa.eu/digital-single-market/en/desi>
- (16) Trașcă, D. L., Ștefan, G. M., Sahlian, D. N., Hoinaru, R., & Șerban-Opreșcu, G. L. (2019). Digitalization and Business Activity. The Struggle to Catch Up in CEE Countries. *Sustainability*, 11(8), 2204.

Dr. Cene Bavec je zaslužni profesor Univerze na Primorskem.

Iz Islovarja

Islovar je spletni terminološki slovar informatike, ki ga objavlja jezikovna sekcija Slovenskega društva INFORMATIKA in ga najdete na naslovu <http://www.islovar.org>. Vabimo vas, da tudi vi prispevate svoje pripombe, predloge ali nove izraze. Tokrat objavljamo izbor izrazov iz zbirke »kartica«.

gôvorna kárstica -e -e ž (*angl. voice card*) razširitvena kárstica, ki omogoča prenos govora v internetni telefoniju

gráfična kárstica -e -e ž (*angl. graphic card*) razširitvena kárstica, ki omogoča računalniško grafiko (1); prim. video kárstica, razširitvena kárstica

izmenljíva kárstica -e -e ž (*angl. interchangeable card*) kárstica, ki jo lahko zamenja uporabnik sam

kárstica -e ž (*angl. card*) ploščica s tiskanim vezjem za nadgradnjo funkcionalnosti ali povečanje zmogljivosti naprav, sistema

kárstica SD -e -- ž (*angl. secure digital card, secure digital standard capacity card, SDSC*) izmenljiva pomnilniška kárstica različnih izvedb, namenjena shranjevanju podatkov v prenosnih napravah

krmílnik oddáljenega dostópa -a -- -- (*angl. baseboard management controller, remote access card, remote management card, BMC, RAC*) krmilnik, ki mimo operacijskega sistema omogoča oddaljen dostop do računalniškega sistema in nadzor različnih senzorjev

lastníška kárstica -e -e ž (*angl. proprietary add-on card*) kárstica za preprečevanje nepooblaščne rabe lastniškega programja; prim. zaščitni ključ

pámetna kárstica -e -e ž (*angl. smart card, ICC, integrated circuit card, chip card*) kárstica z vgrajenim integriranim vezjem, ki omogoča avtentikacijo, shranjevanje in dostop do podatkov

pomnilniška kárstica -e -e ž (*angl. memory card, flash card*) izmenljiva kárstica z bliskovnim pomnilnikom, npr. za prenosnik, digitalni fotoaparát

pospeševálna kárstica -e -e ž (*angl. accelerator board, accelerator card*) kárstica, ki pohitri delovanje procesorja

razšírítvena kárstica -e -e ž (*angl. expansion card, expansion board, adapter card, accessory card*) kárstica, ki doda računalniku nove funkcionalnosti ali zmogljivosti; prim. zvočna kárstica

večkárstíčni čítálnik -a -- m (*angl. multiscard reader*) bralnik, ki omogoča uporabo različnih pomnilniških kárstic; sin. večkárstíčni bralnik

vídeokárstica -e ž (*angl. video card*) kárstica, ki omogoča hitrejšo obdelavo videa (2); prim. gráfična kárstica

zvóčna kárstica -e -e ž (*angl. sound card, audio card*) razširitvena kárstica, ki omogoča prenos in predvajanje zvoka

Včlanite se v Slovensko društvo INFORMATIKA

Pristopna izjava

za članstvo v Slovenskem društvu INFORMATIKA

Pravne osebe izpolnijo samo drugi del razpredelnice

Ime in priimek	
Datum rojstva	
Stopnja izobrazbe	srednja, višja, visoka
Naziv	prof., doc., spec., mag., dr.
Domači naslov	
Poštna št. in kraj	
Ulica in hišna številka	
Telefon (stacionarni/mobilni)	

Zaposlitev člana oz. člana - pravna oseba

Podjetje, organizacija	
Kontaktna oseba	
Davčna številka	
Poštna št. in kraj	
Ulica in hišna številka**	
Telefon	
Faks	
E-pošta	

Zanimajo me naslednja področja/sekcije*

- jezik
- informacijski sistemi
- operacijske raziskave
- seniorji
- zgodovina informatike
- poslovna informatika
- poslovne storitve
- informacijske storitve
- komunikacije in omrežja
- softver
- hardver
- upravná informatika
- geoinformatika
- izobraževanje

podpis

kraj, datum

Pošto društva želim prejemati na domači naslov / v službo.

Članarina znaša: 18,00 € - redna

7,20 € - za dodiplomske študente in seniorje (ob predložitvi dokazila o statusu)

120,00 € - za pravne osebe

Članarino, ki vključuje glasilo društva – revijo **Uporabna informatika**, bom poravnal sam / jo bo poravnal delodajalec.

DDV je vključen v članarino.



Naročilnica na revijo UPORABNA INFORMATIKA

Naročnina znaša: 35,00 € za fizične osebe

85,00 € za pravne osebe – prvi izvod

60,00 € za pravne osebe – vsak naslednji izvod

15,00 € za študente in seniorje (ob predložitvi dokazila o statusu)

DDV je vključen v naročnino.

ime in priimek ali naziv pravne osebe in ime kontaktne osebe

davčna številka, transakcijski račun

naslov plačnika

naslov, na katerega želite prejemati revijo (če je drugačen od naslova plačnika)

telefon/telefaks

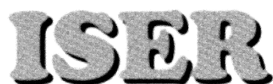
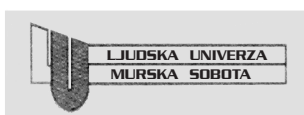
elektronska pošta

Podpis

Datum

Izpitni centri ECDL

ECDL (European Computer Driving License), ki ga v Sloveniji imenujemo evropsko računalniško spričevalo, je standardni program usposabljanja uporabnikov, ki da zaposlenim potrebno znanje za delo s standardnimi računalniškimi programi na informatiziranem delovnem mestu, delodajalcem pa pomeni dokazilo o usposobljenosti. V Evropi je za uvajanje, usposabljanje in nadzor izvajanja ECDL pooblaščen ustanova ECDL Foundation, v Sloveniji pa je kot član CEPIS (Council of European Professional Informatics) to pravico pridobilo Slovensko društvo INFORMATIKA. V državah Evropske unije so pri uvajanju ECDL močno angažirane srednje in visoke šole, aktivni pa so tudi različni vladni resorji. Posebno pomembno je, da velja spričevalo v 148 državah, ki so vključene v program ECDL. Doslej je bilo v svetu izdanih že več kot 11,6 milijona indeksov, v Sloveniji več kot 17.000, in podeljenih več kot 11.000 spričeval. Za izpitne centre v Sloveniji je usposobljenih osem organizacij, katerih logotipe objavljamo.



Znanstveni prispevki

Tamara Čuček, Boštjan Šumak, Maja Pušnik
ANALIZA ORODIJ ZA PODPORO OPTIMIZACIJ PROCESOV

Matej Ulčar, Simon Dobrišek, Marko Robnik-Šikonja
RAZPOZNAVANJE SLOVENSKEGA GOVORA Z METODAMI
GLOBOKIH NEVRONSKIH MREŽ

Kratki znanstveni prispevki

Tadej Škvorc, Marko Robnik-Šikonja
PREPOZNAVANJE IDIOMATSKIH BESEDNIH ZVEZ Z UPORABO
BESEDNIH VLOŽITEV

Borja Bovcon, Matej Kristan
PRIMERJAVA SEGMENTACIJSKIH METOD GLOBOKEGA UČENJA
ZA DETEKCIJO OVIR NA VODI

Ratko Pilipović, Patricio Bulić
HYLO: HIBRIDNI LOGARITMIČNI MNOŽILNIK ZA ENERGIJSKO
UČINKOVITO RAČUNANJE

Razprave

Cene Bavec
ANALIZA DIGITALNE PREOBRAZBE SLOVENIJE V EU OKOLJU

Informacije

IZ ISLOVARJA

ISSN 1318-1882



9 771318 188001

