StuCoSReC

Proceedings
of the 2018
5th Student
Computer
Science
Research
Conference

StuCoSReC

Proceedings of the 2018 5th Student Computer Science Research Conference

Edited by

Iztok Fister jr. and Andrej Brodnik

Reviewers and Programme Committee

Klemen Berkovič • University of Maribor, Slovenia
Lucija Brezočnik • University of Maribor, Slovenia
Patricio Bulić • University of Ljubljana, Slovenia
Zoran Bosnić • University of Ljubljana, Slovenia
Borko Bošković • University of Maribor, Slovenia
Janez Brest • University of Maribor, Slovenia
Andrej Brodnik • University of Primorska and University
of Ljubljana, Slovenia
Haruna Chiroma • Gombe State University, Nigeria
Mojca Ciglarič • University of Ljubljana, Slovenia
Jani Dugonik • University of Maribor, Slovenia
Iztok Fister • University of Maribor, Slovenia
Iztok Fister Jr. • University of Maribor, Slovenia
Matjaž Gams • Jozef Stefan Institute, Slovenia
Tao Gao • North China Electric Power University, China
Andres Iglesias • Universidad de Cantabria, Spain
Branko Kavšek • University of Primorska, Slovenia
Miklos Kresz • University of Szeged, Hungary
Niko Lukač • University of Maribor, Slovenia
Uroš Mlakar • University of Maribor, Slovenia
Marjan Mernik • University of Maribor, Slovenia
Eneko Osaba • University of Deusto, Spain
Vili Podgorelec • University of Maribor, Slovenia
Peter Rogelj • University of Primorska, Slovenia
Xin-She Yang • Middlesex University, United Kingdom
Damjan Vavpotič • University of Ljubljana, Slovenia
Grega Vrbančič • University of Maribor, Slovenia
Aleš Zamuda • University of Maribor, Slovenia
Nikolaj Zimic • University of Ljubljana, Slovenia
Borut Žalik • University of Maribor, Slovenia

Published by

University of Primorska Press
Titov trg 4, SI-6000 Koper

Editor-in-Chief

Jonatan Vinkler

Managing Editor

Alen Ježovnik

Koper, 2018

ISBN 978-961-7055-26-9 (pdf)

www.hippocampus.si/ISBN/978-961-7055-26-9.pdf

ISBN 978-961-7055-27-6 (html)

www.hippocampus.si/ISBN/978-961-7055-27-6/index.html

DOI: <https://doi.org/10.26493/978-961-7055-26-9>

© University of Primorska Press

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in
univerzitetni knjižnici v Ljubljani

COBISS.SI-ID=296769024

ISBN 978-961-7055-26-9 (pdf)

ISBN 978-961-7055-27-6 (html)

Preface

It is no longer necessary to emphasize that computer science is omnipresent today. Terms as digitalization, Industry 4.0 etc. are frequently heard in the mass media and are becoming a part of our everyday life. The rapid advances in computer science require a large investment in research and development. In particular, it is important to educate young scientists, which is one of the objectives of the StuCoSReC conference.

In front of you is the proceedings of the fifth StuCoSReC conference. The StuCoSReC is intended for masters & PhD students to show their research achievements. Socializing is also important and the conference is an excellent opportunity for students to get to know each other. The conference connects students of computer science and goes beyond the invisible limits of faculties. The uniqueness of StuCoSReC conference is that it is organized each year by different higher education institution. The University of Ljubljana - Faculty of Computer and Information Science is proud to host the conference this year.

Twelve papers addressed this conference, covering several topics of the computer science. All the papers were reviewed by two international reviewers and accepted for the oral presentation.

ref. prof. dr. Nikolaj Zimic

Preface

II

Papers

<i>Regulacija plovca v zračnem tunelu (zračna levitacija) na osnovi mehke logike</i> • Primož Bencak, Dominik Sedonja, Andrej Picej, Alen Kolman, Matjaž Bogša	5–13
<i>Razpoznavanje literature v dokumentih in določanje njihovih sklicev v besedilu</i> • Matej Moravec, David Letonja, Jan Tovornik, Borko Boškovič	15–19
<i>Klasifikacija samomorilskih pisem</i> • Dejan Rupnik, Denis Ekart, Gregor Kovačevič, Dejan Orter, Alen Verk, Borko Boškovič	21–25
<i>Načrtovanje poprocesorja za pretvorbo NC/ISO G-kode v translacije za 3/5 osnega robota ACMA ABB XR701</i> • Gregor Germadnik, Timi Karner, Klemen Berkovič, Iztok Fister	27–33
<i>Using constrained exhaustive search vs. greedy heuristic search for classification rule learning</i> • Jamolbek Mattiev, Branko Kavšek	35–38
<i>Real-time visualization of 3D digital Earth</i> • Aljaž Jeromel	39–42
<i>Towards an enhanced inertial sensor-based step length estimation model</i> • Melanija Vežočnik, Matjaž Branko Jurič	43–46
<i>Breast Histopathology Image Clustering using Cuckoo Search Algorithm</i> • Krishna Gopal Dhal, Iztok Fister Jr., Arunita Das, Swarnajit Ray, Sanjoy Das	47–54
<i>Time series classification with Bag-Of-Words approach post-quantum cryptography</i> • Domen Kavran	55–59
<i>The problem of quantum computers in cryptography and post-quantum cryptography</i> • Dino Vlahek	61–64
<i>Named Entity Recognition and Classification using Artificial Neural Network</i> • Luka Bašek, Borko Boškovič	65–70
<i>Context-dependent chaining of heterogeneous data</i> • Rok Gomišček, Tomaž Curk	71

Klasifikacija samomorilskih pisem

Dejan Rupnik Fakulteta za elektrotehniko, računalništvo in informatiko Koroška cesta 46 Maribor, Slovenija dejan.rupnik@student.um.si	Denis Ekart Fakulteta za elektrotehniko, računalništvo in informatiko Koroška cesta 46 Maribor, Slovenija denis.ekart@student.um.si	Gregor Kovačević Fakulteta za elektrotehniko, računalništvo in informatiko Koroška cesta 46 Maribor, Slovenija gregor.kovacevic@student.um.si
Dejan Orter Fakulteta za elektrotehniko, računalništvo in informatiko Koroška cesta 46 Maribor, Slovenija dejan.orter@student.um.si	Alen Verk Fakulteta za elektrotehniko, računalništvo in informatiko Koroška cesta 46 Maribor, Slovenija alen.verk@student.um.si	Borko Bošković Fakulteta za elektrotehniko, računalništvo in informatiko Koroška cesta 46 Maribor, Slovenija borko.boskovic@um.si

POVZETEK

V našem delu smo se osredotočili na klasifikacijo poslovnih pisem samomorilcev. Pozornost smo posvetili na razpoznavo pristnih pisem te narave, od pisem, ki to niso, oz. so le-ta lažna. S pomočjo procesiranja naravnega jezika in algoritmov strojnega učenja želimo doseči, da se pristna pisma v večini ločijo od lažnih. Implementirali smo program v programskem jeziku Python, pripravili korpus in izvedli nadzorovano strojno učenje. Pisma smo klasificirali z metodami: DecisionTreeClassifier, SVC, GaussianProcessClassifier, AdaBoostClassifier, KNeighborsClassifier, RandomForestClassifier, MLPClassifier, GaussianNB in QuadraticDiscriminantAnalysis. Najboljše rezultate smo dosegli z odločitvenim drevesom, kjer smo dosegli 68% natančnost.

KLJUČNE BESEDE

klasifikacija samomorilskih pisem, procesiranje naravnega jezika, odločitvena drevesa

1. UVOD

Zaradi samomora vsako leto umre več kot 800.000 oseb (v Sloveniji več kot 300), približno 25-krat toliko pa jih samomor poskuša narediti [14][11]. Velikokrat, ko vidimo samomorilen zapis, smo v dilemi ali ta oseba misli resno ali pa je to poizkus iskanja pozornosti [7]. Nekatera pisma so tudi lažna - na primer pri umorih, kjer bi želel storilec prikazati, da je žrtev storila samomor. Posledično je njihova klasifikacija pomembna za preventivo ter za razreševanje morebitnih nejasnosti.

Za samo izvedbo klasifikacije smo najprej potrebovali pristna pisma samomorilcev ter lažna pisma. Korpus pristnih pisem

smo pridobili s spletnih strani [12, 3, 6, 4], lažna pa smo sestavili sami. Vsa pisma so v angleškem jeziku. Uporabili smo 63 pristnih in 19 lažnih pisem. Postopek analize pisem je potekal v več korakih. Prvi izmed njih je bilo predprocesiranje besedila, kjer smo kot rezultat dobili prečiščeno besedilo (tj. samo standardni znaki, brez simbolov in ločil). Nato smo izvedli oblikoslovno označevanje besed in povezavo ključnih besed s posameznimi koncepti iz tega področja. Naredili smo statistiko posameznih pisem (povprečno število besed ipd.). Izvedli smo tudi teste berljivosti, kateri so izračunali dve različni metriki. Tudi te smo uporabili pri strojnem učenju. Dodatno smo opravili tudi analizo čustev, nato pa vse rezultate združili v datoteke CSV. Te so bile osnova za gradnjo odločitvenih dreves pri strojnem učenju. Uporabljeno je bilo nadzorovano strojno učenje. Rezultati eksperimenta so pokazali, da je uporabljen pristop uspešno ločeval med pristnimi in lažnimi pismi.

2. SORODNA DELA

V članku [10] so avtorji uporabili 66 poslovnih pisem, od katerih je bilo 33 pristnih in 33 lažnih. V raziskavi je prisotnovalo 11 strokovnjakov s področja mentalnega zdravja in 31 psihiatrov pripravnikov. Zbrana mnenja so primerjali z 9 algoritmi strojnega učenja:

- LMT,
- LinSMO,
- Descision,
- JRip,
- NB,
- PART,
- J48,
- Logistic,
- IB3 in
- OneR.

Kombinacija določenih algoritmov je v povprečju dala 78% natančnost. Specialisti izbranega področja so dosegli 63% natančnost, pripravniki pa zgolj 49% natančnost pri ločevanju pisem.

V članku [2] so naredili študijo vsebine samomorilskih pisem ljudi, ki so samomor skušali storiti ter druge skupine ljudi, ki je samomor dejansko storila. Študijo so izvedli s programom za tekstovno analizo, kateri uporablja jezikovno povpraševanje in štetje besed (angl. Linguistic Inquiry and Word Count).

Članek [1] opisuje študijo samomorilskih pisem iz Mehike. Preučevali so vzorec 212 samomorov, kjer je 106 oseb napisalo poslovilno pismo, ostalih 106 pa tega ni storilo. Ugotovili so, da se osebnostne lastnosti tistih, ki poslovilna pisma napišejo in tistih oseb, ki jih ne, pretirano ne razlikujejo.

3. PRIPRAVA BESEDILA

3.1 Predprocesiranje

Ko smo pripravili korpus s poslovnimi pismi samomorilcev, je bilo le-te potrebno pripraviti za nadaljnjo uporabo. Iz pisem smo odstranili ločila, številke ter vse ostale nestandardne znake. Pri predprocesiranju besedila smo prav tako pretvorili velike črke v male.

Algoritem 1: Priprava besedila.

Input : neobdelano besedilo (*besedilo*)

Output: obdelano besedilo (*izhod*)

```

for znak in besedilo do
  if znak=JE_CRKA() then
    if znak=JE_VELIKA() then
      DODAJ_V_IZHOD(SPREMENLJ_V_MALO(znak))
    else
      DODAJ_V_IZHOD(znak)
    end
  else if znak=JE_PRESLEDEK() then
    DODAJ_V_IZHOD(znak)
  else if znak=JE_OPUSČAJ() then
    DODAJ_V_IZHOD(znak)
  end
end
end

```

3.2 Oblikoslovno označevanje

Za meritve povprečij in ostalih statističnih podatkov smo naredili oblikoslovno označevanje (angl. parts of speech tagging), kjer smo implementirali tokenizacijo in označili besede glede na pomen v povedi. Vsaki besedi se je po zagonu algoritma dodala oznaka, katera nam je povedala za kakšno vrsto besede gre. Uporabili smo knjižnico NLTK [8]. Bila je dodana podpora za večinitno delovanje, ker je lahko proces na velikih korpusih časovno zelo potraten. Metoda z večinitnim delovanjem rezultatov ne shranjuje v spremenljivke v pomnilniku, temveč jih v datoteke na trdem disku.

Po uspešni implementaciji algoritma smo sešteli vse pojavitve oznak ter naredili slovar povprečnih pojavitev določenih označb. Torej, koliko glagolov, ločil, samostalnikov itd., vsebuje povprečno pismo. Nato smo seznam označb normalizirali na omejen nabor značk. Pridobili smo tudi statistiko za vsako posamezno pismo, kjer ohranimo podatek, kateri slovar pripada kateremu pismu.

3.3 Test berljivosti

Nad korpusom smo izvedli testa berljivosti po Flesch in Flesch-Kicaidovi metodi. Ocena berljivosti besedila (angl. Flesch score) smo izračunali po enačbi (1), kjer RE predstavlja enostavnost branja (angl. readability ease).

$$RE = 206,83 - (1,015 * ASL) - (84,6 * ASW) \quad (1)$$

V enačbi (2) ASL predstavlja povprečno dolžino stavka (angl. average sentence length).

$$ASL = \frac{stBesed}{stStavkov} \quad (2)$$

V enačbi (3) ASW predstavlja povprečno število zlogov na besedo (angl. average number of syllables per word).

$$ASW = \frac{stZlogov}{stBesed} \quad (3)$$

Višji je RE (ocena berljivosti besedila) manjša je kompleksnost besedila. Metoda vrne oceno berljivosti besedila (angl. readability score), katera je realno število med 0 in 100, vendar pa zaradi napak v formatiranju lahko pride do odstopanja. Ta odstopanja smo normalizirali na pričakovane vrednosti. Flesch-Kinicaid metoda nam pove nivo berljivosti besedila in se izračuna po enačbi (4), kjer TW predstavlja število vseh besed (angl. total words), TS je število vseh stavkov (angl. total sentences) in TSYL število vseh zlogov (angl. total syllables). Pričakovan rezultat je decimalno število med 0 in 13, katero sovpadajo s povprečnim nivojem berljivosti v šolskih razredih v Združenih Državah Amerike. Zaradi zahteve po številu besed v stavkih, smo teste bralnosti izvajali na osnovnem - neprocesiranem besedilu [13].

$$RL = 0,39 * \left(\frac{TW}{TS}\right) + 11,8 * \left(\frac{TSYL}{TW}\right) - 15,59 \quad (4)$$

Vrednosti RE in RL smo vključili v datoteko CSV.

3.4 Analiza čustev

Za uspešno delovanje algoritma smo potrebovali tudi analizo čustev. Ta nam iz vsakega pisma izlušči emocije, ki se jih da razpoznati na podlagi zapisanega besedila. Pri analizi smo si pomagali z modulom WNAffect [5]. Procesirano besedilo smo vstavili v funkcijo in kot rezultat dobili slovar čustev, ki se pojavijo v besedilu ter vsoto njihovih magnitud. Na koncu smo iz vseh obdelanih pisem in pridobljenih unikatnih čustev ročno generirali statični slovar vseh emocij.

Algoritem 2: Pridobivanje čustev.

function GetEmotion (*besedilo*);

Input : označeno besedilo (*besedilo*)

wna=WNAffect()

```

for znacka in besedilo do
  if znacka != LOČILO then
    custva=PRIDOBILČUSTVA()
    if custvo != NONE then
      return (znacka, custvo, stopnja)
    end
  end
end
end

```

4. ZDRUŽITEV REZULTATOV FUNKCIJ

Ko smo implementirali vse omenjene funkcije, smo morali rezultate združiti ter jih pripraviti za strojno učenje. Rezultate smo zato združili v datoteke CSV. V prvo izmed datotek

Slika 1: Generirano odločitveno drevo.

smo po vrsticah shranili imena pisem, oznake, ocene ter nivo bralnosti in statistiko (št. besed, št. stavkov, itd.) za vsako posamezno pismo.

V drugo datoteko smo shranili skupna povprečja vseh pristnih in nepristnih pisem za vsakega izmed klasifikacijskih atributov posebej. Omeniti je potrebno, da so posamezne vrednosti, kot na primer: značke oblikoslovnega označevanja in magnitude posameznih čustev, normalizirane s številom besed v besedilu. Tabela 1 prikazuje del datoteke povprecja.csv, ki vsebuje povprečja atributov.

5. NADZOROVANO STROJNO UČENJE

Ko smo pripravili datoteki CSV smo se morali odločiti katero vrsto strojnega učenja bomo uporabili. Uporabili smo nadzorovano strojno učenje in zgradili odločitveno drevo, prikazano na sliki 1.

Za generiranje odločitvenega drevesa smo uporabili knjižnico sklearn [9], katera loči bazo klasifikacijskih podatkov na učno in testno množico ter na dva vektorja klasifikatorjev. S pomočjo teh spremenljivk smo nato generirali napovedni model. Iz napovednega modela pa izračunali preciznost, prikljic in mero F1.

Skupno smo uporabili deset različnih klasifikatorjev, pod-

robnejši rezultati štirih najuspešnejših pa so predstavljeni v poglavju 5.1:

- DecisionTreeClassifier(max_depth=5),
- SVC(kernel="linear", C=0,025),
- GaussianProcessClassifier(1,0 * RBF(1,0)),
- AdaBoostClassifier(),
- KNeighborsClassifier(3),
- SVC(gamma=2, C=1),
- RandomForestClassifier(max_depth=5, n_estimators=10, max_features=1),
- MLPClassifier(alpha=1),
- GaussianNB() in
- QuadraticDiscriminantAnalysis().

5.1 Rezultati

V sledečih tabelah so prikazani rezultati klasifikacije z uporabo različnih klasifikatorjev. Rezultati so bili izračunani iz povprečja 10.000 instanc klasifikacije, kjer se je vsakič naključno izbralo 95% pisem za učno in 5% pisem za testno

Tabela 1: Povprečne vrednosti atributov, ki jih uporabimo pri strojnem učenju.

	Pristna pisma	Nepristna pisma
Ocena berljivosti	85,14	84,35
Nivo berljivosti	4,40	5,38
MD	0,02	0,02
PRP	0,04	0,05
RB	0,06	0,07
NN	0,18	0,19
VBP	0,06	0,07
JJ	0,09	0,09
IN	0,09	0,10
DT	0,08	0,6
Število besed	79,59	173,42
Število zlogov	94,24	209,79
Število znakov	304,98	691,32
Število povedi	7,37	14,05
Upanje	0,02	0,00
Ljubezen	0,05	0,02
Obup	0,01	0,00
Sreča	0,01	0,00
Nejasnost	0,01	0,00
Spokojnost	0,02	0,00
Začudenje	0,02	0,01
Sproščenost	0,01	0,00
Zgroženost	0,01	0,00
Zalost	0,01	0,01
Odpušcanje	0,01	0,00
Dobrohotnost	0,01	0,00
Stiska	0,01	0,00
Gotovost	0,01	0,00
Strah	0,01	0,00
Naklonjenost	0,01	0,00

množico. Instanca v našem primeru predstavlja naključno razdelitev pisem na učno in testno množico (iz nabora vseh pisem - pristnih in lažnih). Prikazani so rezultati preciznosti in priklica za posamezen razred, kot tudi povprečne vrednosti. Za mero F1 so izračunane povprečne vrednosti. Za tovrsten pristop testiranja smo se odločili zaradi majhne učne množice.

Za najboljši klasifikator se je izkazala metoda odločitvenih dreves (angl. decision tree classifier), katera je dosegla 68% natančnost, ostale metrike pa so prikazane v tabeli 2. Za izračun povprečja je bilo v vseh tabelah uporabljeno makro povprečje. Tudi metoda AdaBoost se je izkazala kot zelo natančno, vendar je imela malenkost nižjo vrednost priklica. Pri našem testiranju se je tudi izkazala kot bistveno počasnejša, kot metoda odločitvenih dreves.

Rezultati klasifikatorja SVC z linearnim jedrom so zapisani v tabeli 3, klasifikatorja GaussianProcess v tabeli 4 in rezultati klasifikatorja AdaBoost v tabeli 5.

Po analizi klasifikatorjev smo odločitveno drevo še grafično predstavili v formatu PDF. Po pregledu korenov smo ugotovili, da izločanje določenih atributov vrne boljše rezultate. Najboljše rezultate smo dosegli pri izločitvi atributov za število znakov, besed in črk. Preciznost je bila izboljšana za 4%,

priklic za 6% in mera F1 za 5%.

Tabela 2: Rezultati klasifikatorja DecisionTree.

	preciznost	priklic	mera F1
nepravilna	0,52	0,50	0,49
pravilna	0,84	0,89	0,85
skupno povprečje	0,68	0,70	0,67

Tabela 3: Rezultati klasifikatorja SVC.

	preciznost	priklic	mera F1
nepravilna	0,27	0,27	0,27
pravilna	0,77	1,00	0,86
skupno povprečje	0,52	0,63	0,56

Tabela 4: Rezultati klasifikatorja GaussianProcess.

	preciznost	priklic	mera F1
nepravilna	0,38	0,34	0,35
pravilna	0,80	0,94	0,85
skupno povprečje	0,59	0,64	0,60

Tabela 5: Rezultati klasifikatorja AdaBoost.

	preciznost	priklic	mera F1
nepravilna	0,51	0,50	0,49
pravilna	0,84	0,89	0,84
skupno povprečje	0,68	0,69	0,67

6. ZAKLJUČEK

V tem delu smo se osredotočili na klasifikacijo samomorilnih pisem. Program pisma analizira postopoma. Najprej nad korpusom izvedemo predprocesiranje besedila, da dobimo čisto besedilo (same male črke, brez ostalih znakov). Nato se izvede oblikoslovno označevanje besed in statistika posameznih pisem. Opravili smo tudi test berljivosti besedila, ki ga uporabimo pri strojnem učenju. Analiziramo tudi čustva, nato pa vse skupaj združimo v datoteko CSV in izvedemo nadzorovano strojno učenje.

Kot je razvidno iz rezultatov smo zadani projekt razpoznavne pristnih poslovnih pisem uspešno izvedli. Po podatkih avtorjev članka [10] so strokovnjaki s področja psihologije dosegli 63% natančnost, medtem ko naša metoda vrača rezultate z 68% natančnostjo.

Tukaj je pomembno opozoriti, da v našem članku nismo uporabili istega nabora pisem, kot so ga avtorji prej omenjenega članka. Posledično rezultatov naših eksperimentov ni mogoče neposredno primerjati z rezultati v sorodnih člankih. Pridobitev pristnih in lažnih pisem, ki ustrezajo pogojem, je izredno zapleteno opravilo, pri katerem bi za večjo zanesljivost pri zagotavljanju norm potrebovali ustrezno usposobljene osebe, ki bi obenem imele dostop do tovrstnih arhivov.

V prihodnosti bi lahko metodo razširili tako, da ne bi bila omejena samo na angleški jezik. Izboljšave bi tudi prinesla večja množica pisem in dodatne metode za ocenjevanje strukture besedila.

7. VIRI

- [1] L. A. L. L. Chávez-Hernández AM1, Páramo D. Suicide notes in mexico: what do they tell us? *Suicide Life Threat Behav*, 36:709–15, December 2006.
- [2] L. D. Handelman LD. The content of suicide notes from attempters and completers. *Oxford University Press*, 28:102–104, 2007.
- [3] J. Harper. A collection of real suicide notes | historic mysteries, 2015.
- [4] A. A. Leenaars. Suicide notes in the courtroom. 1999.
- [5] C. Michard. clemtoy/wnaffect a python module to get the emotion of a word., 2017.
- [6] O. P. M. Mukta Rani, Shalini Girdhar. Suicide note: The last words. 2015.
- [7] NIJZ. Svetovni dan preprečevanja samomora: "vzemi si trenutek, reši življenje", 2017.
- [8] NLTK. Natural language toolkit, 2017.
- [9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [10] J. Pestian, H. Nasrallah, P. Matykiewicz, A. Bennett, and A. Leenaars. Suicide note classification using natural language processing: A content analysis. *Biomedical Informatics Insights*, 2010:19–28, August 2010.
- [11] Politikis. Zaradi samomora vsako leto v sloveniji umre

več kot 300 ljudi, a zaznati je upad tega žalostnega dejanja, 2017.

- [12] J. Russel. A collection of suicide notes & letters - russel's cyber journal, 2008.
- [13] stephenhky. Flesch-kincaid readability measure - everything about data analytics big data, data analytics, 2016.
- [14] Wikipedia. Suicide, 2018.

University of Primorska Press
www.hippocampus.si
ISBN 978-961-7055-26-9