

TARTU ÜLIKOOL
Loodus- ja täppisteaduste valdkond
Arvutiteaduse instituut
Andmeteaduse õppekava

Ines Anett Nigol

Eesti keeleressurssidel põhinev ristsõnalahendaja

Magistritöö (15 EAP)

Juhendaja: Sven Aller, MSc

Tartu 2024

Eesti keeleressurssidel põhinev ristsõnalahendaja

Lühikokkuvõte:

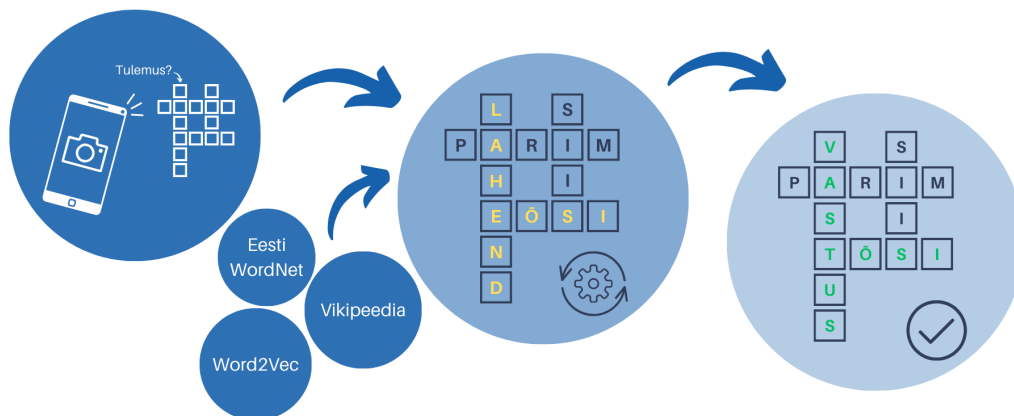
Magistritöö eesmärk oli luua automatiseeritud eestikeelsete ristsõnade lahendaja. Sarnaseid programme on küll ingliskeelsete ristsõnade lahendamiseks loodud, kuid eesti keelel põhinevat lahendust veel teadaolevalt ei leidu. Tulemusena loodi programm, kuhu saab üles laadida pildi ristsõnast ning mis pakub ristsõnas olevatele küsimustele võimalikke lahendusi. Lahendus põhineb Eesti Wordnetist, eeltreenitud Word2Vec mudelist ja Vikipeediast kogutud andmetel. Töös antakse ülevaade pildilt ristsõnaruudustiku ja vihjete tuvastamiseks kasutatud meetoditest, andmete kogumisest, sõnade ruudustikku sobitamisest, loodud veebirakendusest ja töö tulemustest.

Võtmesõnad:

Keeletehnoloogia, keeleteadus, eesti keel, EstNLTK, Wordnet, EstWN, ristsõna

CERCS: P175 - Informaatika, süsteemiteooria

Eesti keeleressurssidel põhinev ristsõnalahendaja



Crossword Solver Based on Estonian Language Resources

Abstract:

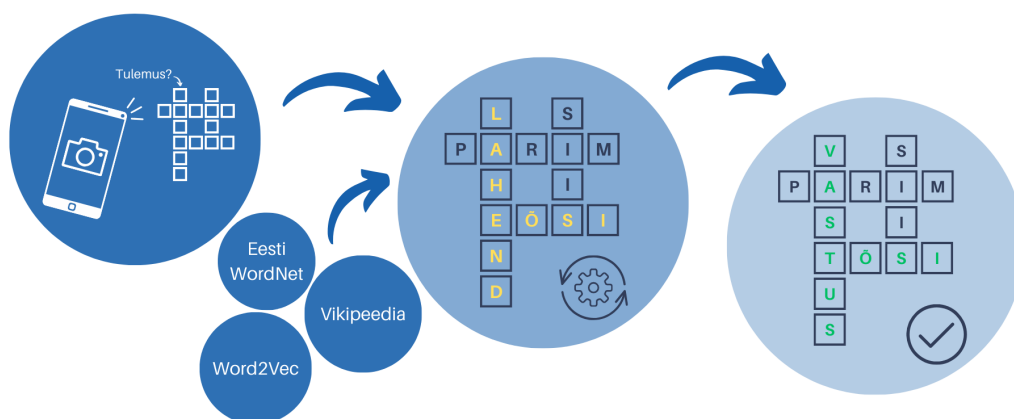
The goal of this Masters's thesis was to create an automated Estonian crossword solver. Similar programs have been created to solve English crosswords, but a solution based on the Estonian language is not yet known. As a result, a program was created where you can upload a picture of the crossword and which offers possible solutions to the questions in the crossword. The solution is based on data collected from Estonian Wordnet, pre-trained Word2Vec model and Wikipedia. The thesis describes the methods used to identify the crossword grid and clues from the image, the data collection methodology, matching the words to the grid, the created web application and results.

Keywords:

Language technology, language science, Estonian language, EstNLTK, Wordnet, EstWN, crossword puzzle

CERCS: P175 – Informatics, system theory

Crossword Solver Based on Estonian Language Resources



UNIVERSITY OF TARTU
Institute of Computer
Science

Author: Ines Anett Nigol
Supervisor: Sven Aller
Data Science (MSc), 2024
#UniTartuCS

Sisukord

1	Sissejuhatus	6
2	Kasutatud mõisted ja terminid	8
3	Kirjanduse ülevaade	9
3.1	Ülevaade olemasolevatest ristsõnalahendajatest	9
3.1.1	Eestikeelsed ristsõnalahendajad	9
3.1.2	Küsimus-vastus paaridel põhinevad lahendajad	9
3.1.3	Veebikoorimisel põhinevad lahendajad	11
3.2	Pildituvastuse rakendamine ristsõnade lahendamisel	11
3.3	Vihjetüübid	12
4	Automaatse ristsõnalahendaja rakendus	14
4.1	Kasutatud tehnoloogilised vahendid	14
4.1.1	EstNLTK	14
4.1.2	Re	15
4.1.3	OpenCV	15
4.1.4	Python-tesseract	15
4.1.5	Flask	16
4.2	Ristsõna tuvastamine pildilt	16
4.2.1	Ristsõnaruudustiku tuvastamine	16
4.2.2	Pilditöötlus	17
4.2.3	Tekstituvastus	20
4.2.4	Lahtrite klassifitseerimine	21
4.3	Vastusekandidaatide genereerimine	23
4.3.1	Eesti Wordnet	23
4.3.2	Eesti Vikipeedia	23

4.3.3	Lühendid	24
4.3.4	Word2Vec	24
4.3.5	Kogutud andmete eeltöötlus	25
4.4	Kandidaatide ruudustikku sobitamine	25
4.5	Veebirakendus	26
5	Programmi testimine ja tulemused	28
5.1	Ruudustiku tuvastamine	28
5.2	Teksti tuvastamine	30
5.3	Lahtrite klassifitseerimine	32
5.4	Pakutud ristsõnalahendused	34
5.5	Tagasiside veebirakendusele	35
5.6	Edasised arenguvõimalused	36
6	Kokkuvõte	38
	Viidatud kirjandus	40
	Lisad	44
	I. Näide lahendatud ristsõnast	44
	II. Litsents	45

1 Sissejuhatus

Ristsõnu kasutatakse nii hariduslikel kui ka meelelahutuslikel eesmärkidel. Esimene eestikeelne ristsõna ilmus 100 aastat tagasi ning tänapäeval ilmub igas kuus nii paberil kui ka internetis mitmeid ristsõnakogumikke [1]. Kõige enam on Eestis tuntud Skandinaavia stiilis ristsõnad, kus küsimused asuvad ruudustiku sees ning vastused kirjutatakse suunaga paremale või alla [2].

Paberil välja antavate ristsõnade hulk on oluliselt suurem kui digiväljaannete oma. Digiristsõnade puhul on sageli olemas automaatne vastusekontroll, kuid paberväljaande täielikult lahendatud versioone üldjuhul samas numbris välja ei anta. Sageli tuleb vastuste kontrollimiseks oodata järgmise numbri ilmumiseni, mõnikord õigeid vastuseid ei avaldatagi.

Ristsõnade lahendamisel leitakse abi mitmetest kohtadest. Välja antakse raamatuid, kus on ristsõnades sageli esinevad sõnad sorteeritud pikkuste ning teemade järgi. Samuti leidub üksikuid eestikeelseid internetilehekülgi, mis vihje ja tähtede arvu põhjal tagastavad lahtritesse sobivate sõnade listi. Võimalik on postitada ka tekkivad küsimused internetifoorumitesse ja oodata teiste lahendajate abi. Viisi, mis aitaks eestikeelset ristsõna kiirelt ja vähese vaevaga kontrollida, ei leidu.

Lõputöö eesmärk on luua programm, mis eestikeelsest ristsõnast pildi üles laadimisel tagastab võimaliku lahenduse. Töö keskendub Skandinaavia stiilis ristsõnade lahendamisele, milles küsimused asuvad ruudustiku sees.

Käesolev töö koosneb sissejuhatusest, kirjanduse ülevaatest, sisupeatükkidest ning kokkuvõttest. Teoreetilise tausta osas tuuakse ülevaade olemasolevatest ristsõnade lahendamise rakendustest ja pildilt ruudustikke tuvastavatest lahendustest. Samuti kirjeldatakse, millised vihjetüübid tavaliselt ristsõnades esinevad. Sisupeatükkides on toodud ülevaade ristsõna loomisel kasutatud tehnoloogilistest vahenditest, ristsõna pildilt tuvastamise etappidest, andmete kogumisest, võimalike lahenduste kokkupanekust ning veebirakendusest. Kirjeldatakse erinevate komponentide funktsionaalsust. Loetletud on antud töö

käigus loodud ristsõnalahendaja piirangud ning võimalikud edasiarendamise võimalused.

2 Kasutatud mõisted ja terminid

API (ingl <i>Application Programming Interface</i>) Reeglite ja protokollide kogum, mis võimaldab tarkvarakomponendid omavahel siduda ning vahetada nende vahel informatsiooni [3].
Eesliides (ingl <i>Frontend</i>) Veebilehe graafiline pool, mille kaudu kasutaja suhtleb rakendusega [4].
Eesti Wordnet ehk EstWN (ingl <i>Estonian Wordnet</i>) Omavahel viidete kaudu ühendatud sünohulkade kogum [5].
Itereerima (ingl <i>Iterate</i>) Korduvate tsüklite abil sooritama, andes igale kordusele kaasa informatsiooni eelmisest tsüklist ¹ .
Optiline märgituvastus (ingl <i>Optical Character Recognition</i>) Optiline märgituvastus on protsess, mis võimaldab pildil leiduva teksti konverteerida masinloetavale tekstikujule [6].
Sünohulk ehk sünonüümihulk (ingl <i>Synset</i> ehk <i>Synonym set</i>) Sünohulk koosneb kõigist ühte ja sama mõistet väljendavatest sõnadest (või sõnaühenditest) [5].
Tagaliides (ingl <i>Backend</i>) Veebirakenduse serveripool, mis tagab eesliidese toimimise andes ligipääsu andmebaasile [4].
Teek (ingl <i>Library</i>) Funktsioonide, moodulite ja muude komponentide kogu, mida on võimalik programmides kasutada ² .
Veebirakendus (ingl <i>Web application</i>) Programm, mis töötab serveris ning millele tagatakse ligipääs veebibrauseri kaudu [7]. Veebirakendus koosneb eesliidest ja tagaliidest.

²Sõnaveebi veebileht: <https://sonaveeb.ee/>

3 Kirjanduse ülevaade

3.1 Ülevaade olemasolevatest ristsõnalahendajatest

3.1.1 Eestikeelsed ristsõnalahendajad

Ristsõnade lahendamisel hätta jäädes otsitakse üldjuhul abi konkreetsele vihjele internetist või raamatutest. Leidub spetsiaalseid teatmikke, kus on mitmetest eluvaldkondadest kokku kogutud sõnad, mida ristsõnades sageli küsitakse. Täielikult automatiseeritud eestikeelset ristsõnalahendajat, mis suudaks pakkuda ristsõnale pildi järgi lahendust, ei ole teadaolevalt loodud.

Internetist leiab kaks veebilehte, mis on loodud abiks eestikeelsete ristsõnade lahendamisele. Nendeks on Ruudustik³ ning Spikker⁴. Kumbki pole terviklik ristsõnalahendaja, kuna korraga saab vastust otsida vaid ühele küsimusele. Sisestades vihje, sõnapikkuse ning olemasolevad tähed otsivad programmid sobiva muustriga sõnu sõnastikest ning Vikipeediast.

Vabalt on kätte saadav ka A. Rootsi poolt loodud poomismängu- ja ristsõnalahendaja kood⁵. Antud lahendus ei otsi sõnu vihjete järgi, vaid valib Eesti Keele Instituudi (EKI) andmebaasist sobiva muustriga sõnu. Kasutaja peab sisestama, millised tähed on tal endal juba leitud, ning tagastatavate sõnade hulgast valima teemasse sobiva vaste.

3.1.2 Küsimus-vastus paaridel põhinevad lahendajad

Paljud ristsõnade lahendamise algoritmid põhinevad suurtel küsimus-vastus paaride hulgal. Eeldatakse, et ligikaudu 98% uutes ristsõnades esinevatest küsimustest on juba

³Veebileht Ruudustik: <https://ruudustik.ee/>

⁴Veebileht Spikker: <https://spikker.planet.ee/>

⁵Poomismängu- ja ristsõnalahendaja kood: <https://gist.github.com/anroots/2656924>

varasemalt esinenud ning on seega andmebaasist leitavad. Näiteks Wallace jt kogusid USA levinumate internetiajakirjade ristsõnadest kokku 6,4 miljonit küsimus-vastus paari [8].

Lahendusalgorithm koosnes mitmest sammust. Kõigepealt otsiti andmebaasist iga le küsimusele sobivad vastused ning pandi paika esialgne vastuse õigsuse tõenäosus. Seejärel kasutati Proverbi ristsõnalahendajast inspireeritud tsüklilist uskumuste edasikandumise (ingl *loopy belief propagation*) meetodit. Meetod otsib lahendust, maksimeerides lahenduse suurimat eeldatud kattuvust õige vastusega [9]. Ruudustik täideti ahne otsinguga (ingl *greedy search*) ning vastusekandidaatide ristsõnasse sobivuse tõenäosusi korregeeriti, arvestades ristuvatest sõnadest tulenevate tähepiirangutega [8].

Põhiliseks probleemiks, mis taolise lähenemise puhul välja toodi, oli see, et ebamääraste küsimusevormingute puhul ei suutnud algoritm andmebaasist sobivaid vasteid leida. Samuti mainiti, et lahendusalgorithm teeb kõige rohkem vigu temaatiliste ristsõnade lahendamisel, kuna iga küsimust käsitletakse kui eraldi probleemi.

Chen jt lähenes ristsõnade lahendamise probleemile sarnaselt [10]. Lisaks internetist leitud täidetud ristsõnadest küsimus-vastus paaride eraldamisele laiendati töös andmestikku mitmete sõnastike ja *Wikidata* abil. Lahendusalgorithmis otsiti välja võimalikud vastusekandidaadid ning kasutati Monte Carlo otsingupuud parima lahenduse leidmiseks. Iga küsimuse juures katsetati iteratiivselt erinevaid sobivaid variante läbi. Kuna võimalikke lahendusi võis olla palju, uuendati parimat lahendust paika pandud ajalimiidi täitumiseni.

Eestikeelseid lahendatud ristsõnade ning küsimus-vastus paaride andmebaase vabalt kättesaadaval ei ole. Seega tuleb eestikeelsete ristsõnade automatiseeritud lahendamise ülesandele läheneda teisiti.

3.1.3 Veebikoorimisel põhinevad lahendajad

Üks tuntuim veebiallikatel põhinev ristsõnalahendaja on Webcrow [11]. Tegu on itaaliakeelsete ristsõnade lahendamise tarkvaraga, mille põhimõtteid on rakendatud ka näiteks saksa- ja prantsusekeelsete ristsõnade automatiseeritud lahendamisel [12, 13]. Iga vihjesõnastatakse ümber selliselt, et otsingumootor leiaks internetist teema kohta käivad veebidokumendid. Neist eraldatakse võimalikud vastusekandidaadid. Ruudustiku täitmise algoritm otsib parimat võimalikku lahendust, maksimeerides ristsõnasse sobivate sõnade arvu.

Otsingumootorite Google ja Bing kasutamiseks on loodud API-d, kuid nende kaudu suure hulga päringute saatmine on tasuline. Seega keskenduti lõputöös vabalt kättesaadavatele allikatele.

3.2 Pildituvastuse rakendamine ristsõnade lahendamisel

Eelnevates peatükkides kirjeldatud ristsõnalahendajatele anti sisendiks ette masintöödeldaval kujul ristsõna. Gammedda jt loodud süsteem keskendub pildi järgi Briti stiilis ristsõnade lahendamisele [14]. See tähendab, et ruudustikus on antud vihjele viitavad numbrid ning vihjed ise asuvad ristsõna kõrval.

Lahtrite tuvastamiseks kasutati OpenCV teeki. OpenCV *findContours()* funktsiooni abil leiti kõik ristsõnas leiduvad kontuurid. Nende hulgast kõige suurem tähistas ristsõna piirdeid. Seejärel otsiti välja kõikide kontuuride lõikumispunktid, mis tähistasid ruudustikus olevate lahtrite nurki. Lahtrid lõigati ruudustikust välja ning iga ruudu kohta salvestati tema number, värv ning lahtris viidatud vihje number. Vihjete tuvastamiseks anti süsteemile ette pildid alla ja paremale suunduvatest vihjetest ning tekstituvastuseks kasutati optiliste märkide tuvastuse tööriista Python-tesseract.

Peale vihjete kategoriseerimist moodustati igale vihjele vastusekandidaadid, põhinedes varasematele ristsõnalahendustele, sõnastikele ning internetist kokku kogutud

andmetele. Varasematel ristsõnalahendustel treenitud närvivõrk valis välja kõige tõenäolisema vastuse. Loodud lahendus saavutas keskmiseks lahendamise täpsuseks üle 90%.

Pildituvastusest on abi ka sudokude automatiseeritud lahendaja loomisel. Simha jt süsteemile oli võimalik ette anda digitaalse kaamera abil tehtud pilt [15]. Esimese sammuna muudeti pilt binaarseks, kasutades filtrit, mis kohanduva läve (ingl *adaptive threshold*) abil muutis heledamad pikslid valgeks ning tumedamad mustaks. See võimaldas eemaldada müra ning pildile jäänud varjude mõju. Seejärel tuvastati sudoku välimine ruut. Kuna sudokus on 9x9 ruudustik, sai vahejoonte asukohad tuletada kõige suuremast ruudust. Ruutude sees olevad numbrid tuvastati etteantud numbrimallide sobitamise meetodil (ingl *template matching*). Ristsõnaruudustikus ei ole alati lahtrite arv sama. Seetõttu tuleb magistritöös ka vahejooned pildilt automaatselt leida, mis omakorda raskendab töö käiku.

3.3 Vihjetüübid

Ristsõnades leidub erinevaid vihjetüüpe. Wallace jt jagasid 200 suvalist ristsõnaküsimust kuude kategooriasse [8]. Kõige enam ehk 37% esines nende seas faktiküsimusi, mis vajasid lahendamiseks entsüklopeediateadmisi. Nendele küsimustele vastates tuleb arvestada ka ristsõna välja andmise ajaga, kuna oodatud vastus võib olla vananenud ning lahendamise ajahetkel võib reaalne olukord olla erinev. Ligikaudu sama palju oli ristsõnades definitsioonile või sünonüümile vastete leidmise küsimusi.

Välja toodi ka loogikaküsimused, mis tuginevad inimeste arusaamal ümbritseva kohta. Näiteks ülem- ja alamhulgad, osa ja tervik või põhjus ja tagajärg seosed. Taolisi küsimusi oli eelmistest kategooriatest poole vähem. Ristsõnades kasutati ka sõnamängudel, lünkade täitmisel põhinevaid vihjeid. Samuti toodi välja, et mõnele ristsõnas leiduvale küsimusele on võimalik vastata alles siis, kui esmalt on lahendatud mõni muu ristsõnas paiknev vihje.

Kulshreshtha jt teadustöös jagati 1000 küsimus-vastuse paari kümnesse erinevasse kategooriasse [16]. Ka nende sõnul oli ristsõnades põhirõhk fakti- ja sünonüümiküsimustel. Lisaks eelnevas teadustöös välja toodud kategooriatele eristati ajaloolisi küsimusi, võõrkeelsete väljendite tõlkimist ning lühendite ja liidete kasutamist.

4 Automaatse ristsõnalahendaja rakendus

Käesolevas peatükis tuuakse ülevaade loodud rakenduses kasutusel olevatest tehnoloogilistest vahenditest, andmete kogumisest ja töötlemisest, ristsõnade pildilt tuvastamise etappidest ning leitud kandidaatide ruudustikku sobitamisest. Kirjeldatud on loodud veebirakendust ning selle testimist.

4.1 Kasutatud tehnoloogilised vahendid

Rakenduse tagaliides baseerub Python⁶ programmeerimiskeelel. Veebirakenduse loomisel on kasutusel hüperteksti märgenduskeel *HTML (HyperText Markup Language)*⁷ koos kaskaadlaadistikuga *CSS (Cascading Style Sheets)*⁸, mis lihtsustab veebilehe kujundamist.

4.1.1 EstNLTK

EstNLTK (*Estonian Natural Language ToolKit*) [17] on põhiliselt Pythoni programmeerimiskeeles loodud vabalt kättesaadav teekide kogumik. Sealt leiab vahendid eestikeelsete tekstide töötlemiseks. Magistritöös kasutati EstNLTK lemmatiseerimise funktsiooni küsimuste ja vastuste korrastamisel, katsetati ka õigekirjakontrolli. Samuti saadi EstNLTK kaudu ligi Eesti Wordneti andmebaasile.

⁶Pythoni kodulehekülg: <https://www.python.org/>

⁷HTML kodulehekülg: <https://html.com/>

⁸CSS kodulehekülg: <https://www.w3.org/Style/CSS/>

4.1.2 Re

Pythoni baasmoodulis leiduv `Re`⁹ moodul tegeleb regulaaravaldistega. Regulaaravaldis koosneb kokkulepitud tähendusega märkidest, mis panevad paika otsingumustri. Moodulit kasutatakse sageli keelega seotud ülesannete lahendamisel, kuna seal asuvad funktsioonid võimaldavad leida sõnedes olevaid mustreid. Antud töös kasutatakse regulaaravaldisi tekstiliste andmete puhastamiseks ning ristsõnas ristuvate sõnade sobivuse kontrolliks.

4.1.3 OpenCV

`OpenCV`¹⁰ (*Open Source Computer Vision Library*) on vabalt kättesaadava lähtekoodiga arvutinägemise ja masinõppe teek. Teegis leiduvad tööriistad ja funktsioonid on suunatud piltandmetega töötamisele. `OpenCV` leiab kasutust pilditöötuse, objektide tuvastuse ja paljude muude piltidega seotud ülesannete lahendamisel. Antud töös kasutatakse teeki pildilt ristsõnaruudustiku tuvastamiseks ning kasutajale tulemuste kuvamiseks.

4.1.4 Python-tesseract

`Python-tesseract`¹¹ on optilise märgituvastuse tööriist. Tööriist põhineb avatud lähtekoodiga Google'i tekstituvastusmootoril `Tesseract-OCR`. Moodul võimaldab piltidelt või skaneeritud dokumentidelt tuvastada teksti rohkem kui 100 erinevas keeles. Käesolevas töös kasutatakse seda ristsõnaruudustikust vihjetekstide tuvastamiseks.

⁹Re dokumentatsioon: <https://docs.python.org/3/library/re.html>

¹⁰OpenCV kodulehekülg: <https://opencv.org/about/>

¹¹Pytesseracti dokumentatsioon: <https://pypi.org/project/pytesseract/>

4.1.5 Flask

Flask¹² on Pythonil põhinev mikroveebiraamistik, mida kasutatakse veebirakenduste loomiseks. Flask võimaldab siduda kasutaja vaate ja tegevused serveriga.

4.2 Ristsõna tuvastamine pildilt

Ristsõnade pildilt tuvastamiseks tuli läbida mitu etappi. Järgnevates alapeatükkides on toodud ülevaade ristsõnaruudustiku tuvastamisest, lahtrite klassifitseerimisest ning lahtrites olevate tekstide tuvastamisest.

4.2.1 Ristsõnaruudustiku tuvastamine

Esimese sammuna teisendati pilt mustvalgele skaalale, mis lihtsustas programmi tööd. Müra vähendamiseks ja piirjoonte suurendamiseks rakendati pildil Gaussi udustamisfunktsiooni (ingl *Gaussian blur*) ning binaarset filtrit. Binaarne filter muutis kohanduva läve abil heledamad pikslid valgeks ning tumedamad mustaks. OpenCV *findContours()* funktsiooni abil leiti pildilt kõik kontuurid, millest suurim loeti ristsõna ümbrisevaks piirjooneks. Selle nurkade põhjal lõigati ristsõnaruudustik pildilt välja.

Ristsõnaruudustikus asuvate lahtrite tuvastamiseks analüüsiti kõiki OpenCV *findContours()* funktsiooni põhjal leitud kontuure. Joonisel 1 olevast ristsõnaväljalõikelt¹³ on näha, et leitud kontuuride seas leidub palju müra. Sageli leitakse kontuure ristsõna illustreerivalt pildilt. Samuti loetakse kontuuri alla vahel ristsõnalahtris asuv tekst. Seetõttu puhastati leitud kontuurid enne lahtrite väljalõikamist.

Ristsõnalahtrite väljafiltreerimisel võeti aluseks, et üldjuhul on ristsõnades esinevad

¹²Flaski dokumentatsioon: <https://flask.palletsprojects.com/en/3.0.x/>

¹³Ristsõna autor on Tõnu Paldra, ristsõna pärineb Nuti ristsõnade leheküljelt <https://ristsonad.ee/r/tasutari-ristsona/> (02.05.2024)



Joonis 1. Ristsõnast tuvastatud puhastamata kontuurid (rohelised). Puhastamata kujul leidub nende hulgas palju müra.

lahtrid riskülikukujulised. OpenCV *boundingRect()* funktsioon võimaldas välja arvutada iga kontuuri piirava riskülikukujulise väljalõike pindala. Juhul kui kontuuri ja teda ümbritseva risküliku pindalad olid ligikaudu võrdsed, võis arvata, et tegu on lahtriga. Antud loogikat järgides oli võimalik riskülikukujulised kontuurid teiste hulgast välja filtreerida. Samuti grupeeriti leitud kontuurid pindalade järgi. Kõige arvukam grupp koosnes ristsõnast leitud lahtritest, eeldusel, et kõik lahtrid on ligikaudu võrdse suurusega. Puhastamise tulemus on näha joonisel 2.



Joonis 2. Ristsõnast väljafiltreeritud kontuurid (rohelised).

Lahtrinurkade asukohtade põhjal konstrueeriti üle kogu ristsõna ruudustik, mis võimaldas leida, kuidas lahtrid teineteise suhtes paiknevad. Ruudustikust lõigati lahtrid väikese ülekattega välja, et lahtrisisu suurema tõenäosusega tervenisti peale jääks. Iga ruudu kohta salvestati ka tema asukoht ristsõnas.

4.2.2 Pilditöötlus

Kuigi sageli on tekst pildil silmaga selgelt eristatav, ei suuda arvutiprogramm seda alati ära tunda. Tähtede tuvastamist häirivad näiteks ebaühtlane taust, müra ning erinevast valgustusest tulenevad heledad laigud ja varjud [18]. Segavate faktorite mõju vähendamiseks on võimalik pilte enne tekstituvastussammu töödelda.

Eeltöötlussammude hulka on võimalik valida palju erinevaid transformatsioone. Transformeerimine on protsess, mille käigus omistatakse pikslitele kindlate reeglite järgi uued väärtused [18]. Sageli muudetakse näiteks pildi heledust, kontrasti ja teravust, viiakse pilt hallskaalale või omistatakse pikslitele kindlate reeglite põhjal uued väärtused.

Enne teiste transformatsioonide rakendamist viidi pilt hallskaalale ning eemaldati tumedad äärejooned, mis tekkisid lahtrite ristsõnast väljalõikamisel.

Antud ülesande jaoks teiste sobivate transformatsioonide väljavalimiseks pandi kirja 30 erinevat transformatsioonivalemit. Nende hulgast koostati kõik kolmest transformatsioonist koosnevad kombinatsioonid. Transformatsioonide kasulikkust hinnati järgneval meetodil.

Kõigepealt valiti kolmest erinevast ristsõnast välja kolm keerulisemat vihjet sisaldavat lahtrit. Igale lahtrile lisati käsitsi juurde silt, millel oli kirjas pildilt silmaga loetav tekst. Silt tähistas vastust, mida tekstituvastusmodelilt oodati.

Pilt igast väljavalitud lahtrist läbis kombinatsiooni transformatsioonidest. Seejärel tuvastati pildilt Python-tesseract tekstituvastusmodeli abil tekst. Tuvastusmodeli headust hinnati, võrreldes mudeli tagastatud teksti ja käsitsi kirja pandud õiget vastust Levenshteini kauguse [19] põhjal. Mõõdeti, mitu muudatust on vaja teha, et tekstid teineteisega vastavusse viia. Nõnda käituti iga pildi ja igast kolmest transformatsioonist koosneva kombinatsiooni korral. Kokku leiti tulemused seega 30x30x30 kombinatsiooni jaoks.

Eelnevalt kirjeldatud etapist valiti välja 10 kõige parema tulemuse saavutanud kombinatsiooni. Tulemuste vaatlusel oli näha, et tekstituvastus teeb jätkuvalt palju vigu. Seetõttu suurendati korraga rakendatavate kombinatsioonide hulka järgneval viisil. 30-st transformatsioonist loodi kõikvõimalikud transformatsioonipaarid. Paarid lisati eelmisest etapist leitud 10 parimale kombinatsioonile. See tähendas, et iga pilt läbis enne tekstituvastust 5-st transformatsioonist koosneva eeltöötlusetapi. Tekstituvastuse headust hinnati samamoodi nagu eelnevas etapis. Selles etapis katsetati läbi 10x30x30 kombinatsiooni.

Parimaks viiest transformatsioonist koosnevaks kombinatsiooniks loeti valimil kõi-

ge kõrgema tulemuse andnud kombinatsioon. Järgnevas listis on toodud parimad sel meetodil leitud transformatsioonid:

1. pildimõõtmete 5-kordne suurendamine
2. piksliväärtuste korrutamine 1,5-ga
3. piksliväärtuste vähendamine 50 võrra
4. morfoloogiline avamine
5. piksliväärtuste binaarseks muutmine

Esimest transformatsiooni ehk pildi suurendamist kasutatakse sageli tekstituvas-
tismudeli täpsuse parandamiseks. Kõige paremini tuvastab tekstituvas-
tismudel teksti piltidelt, mille resolutsioon on umbes 300 punkti tolli kohta (ingl *DPI* ehk *dots per inch*) [20]. Ristsõnadest väljalõigatud lahtrid olid sellest väiksema resolutsiooniga. Pildi 5-kordne suurendamine koos puuduvate piksliväärtuste interpoleerimisega viis resolutsiooni soovitatud vahemiku lähedusse.

Teise sammuna korrutati iga piksli väärtus 1,5-ga. Transformatsioon muutis pildi heledamaks. Kolmandaks transformatsiooniks oli piksliväärtuste vähendamine 50 ühiku võrra. See muutis pildi tumedamad osad veelgi tumedamaks, mis tõi piirjooned paremini esile.

Neljas transformatsioon ehk morfoloogiline avamine (ingl *morphological opening*) on meetod, mis aitab eemaldada pildilt müra ning tuua servi paremini esile [21]. Antud teisendus muudab piksliväärtused naaberpikslitele sarnasemaks. Seeläbi eemaldatakse isoleeritud teist värvi pikslid ühtlaselt taustalt. Morfoloogilist avamist kasutatakse sageli koos morfoloogilise sulgemisega (ingl *morphological close*), mis eemaldab pildi esiplaanil asuvatest objektidest augud. Seetõttu lisati eeltöötlusetappide hulka mõlemad meetodid.

Viienda sammuna muudeti piksliväärtused binaarseks. See tähendab, et tumedamad pikslid muudetakse mustaks ja heledamad valgeks. Üks võimalus tumeda ja heleda piirkonna eraldamiseks on paika panna globaalne lävi, kuid see eksib piltidel, mis on erinevalt valgustatud. Kohanduva läve (ingl *adaptive thresholding*) meetodit kasutades arvutatakse lävi jooksvalt pikslit ümbritsevate naaberpikslite järgi [22]. Nõnda kasutatakse pildi eri piirkondades pikslite mustaks või valgeks määramisel erinevat läve. Binaarselt pildilt otsib tekstituvastusmudel mustade pikslite seast teksti ning valged pikslid loetakse tausta hulka.

4.2.3 Tekstituvastus

Pärast eelnevas peatükis kirjeldatud pilditöötlussammude läbimist tuvastati igalt ristsõnast välja lõigatud pildilt Python-tesseract teegi abil tekst. Teegile tuli ette anda, millises keeles tulemust oodatakse. Tehti eeldus, et eestikeelsetes ristsõnades domineerib eestikeelne tekst.

Python-tesseract eeldab vaikimisi, et ette antakse lehekülg täis teksti. Vastavalt sisendpildile on võimalik valida ka teiste lehekülje jaotamise režiimide (ingl *page segmentation modes* ehk *PSM*) vahel [23]. Käesolevaks tööks sobivaima režiimi valimiseks võeti ette valim teksti sisaldavatest lahtritest ning vaadeldi, milliseid tekste erinevad režiimid tagastavad. Tulemusi vaadeldes ilmnes, et erinevat PSM-i kasutades tagastatakse vahel erinev tekst. Mõni režiim suutis paremini eristada üherealist teksti, teine mitmerealist. Samuti tehti erinev hulk kirjavigu. Seega rakendati igal pildil kolme sobivat lehekülje jaotamise režiimi. Valituks osutusid PSM-3, PSM-4 ja PSM-6:

- **PSM-3** on Tesseract teegis vaikimisi valitud säte. Eeldatakse, et pilt sisaldab lehekülge täis teksti, millel on mitu sõna, rida ja lõiku. Sageli on vihjed mitmesõnalised ja lahtrites leidub mitu rida. Seetõttu suutis antud režiim enamasti etteantud pildilt üpris edukalt teksti tuvastada.

- **PSM-4** eeldab, et pildil olev tekst on ühes tulpas. Antud režiim suudab tuvastada teksti ka juhul, kui pilt sisaldab erinevas suuruses tähemärke. Vihjelahtrites on tekst ühes tulpas, mistõttu on see režiim antud ülesande jaoks sobilik.
- **PSM-6** eeldab, et ette on antud ühtlase kirjastiili ja -suurusega tekst, mis võib sisaldada mitut rida. Sageli kehtib see ühte lahtrisse kirjutatud teksti kohta ning annab seega õigeid tulemusi.

Lahtris tuvastatud tekstiks määrati eri viisidel tuvastatud tekstide mediaanväärtus Levenshteini kauguse põhjal. Vahel leidis tekstituvastus teksti piltidelt, kus seda tegelikult ei olnud. Kui vaid üks kolmest tuvastamisviisist tagastas teksti, loeti see veaks ning lahtrile teksti ei määratud.

Teksti tuvastamisele järgnes teksti puhastamine. Sageli poolitatakse vihjetekste, et need lahtritesse paremini ära mahuksid. Seega ühendati omavahel kõik sidekriipsuga eraldatud sõnad. Vihjetest eemaldati kõik kirjavahemärgid, välja arvatud punktid, mis üldjuhul tähistasid puuduvat sõnaosa.

Vahel sisaldas tuvastatud tekst kirjavigu. Nende eemaldamiseks prooviti tekstidel EstNLTK teegist *SpellCheckRetagger* tööriista rakendamist. Tööriist hindas kas sõna on õigesti kirjutatud. Juhul kui sõna loeti vigaseks pakuti võimalikke parandusi. Antud töös asendati iga vigaseks loetud sõna esimese tööriista pakutud vastega. Vihjetes esineb tihti sõnaosi ja nimesid. Sellised kohad parandas õigekirjakontroll sageli valeks. Seega otsustati lõppversioonis teksti puhastamisel õigekirjakontrollivahendit mitte kasutada, et vähendada juhtumeid, kus tekstituvastusel leitud õige tekst valeks muudetakse.

4.2.4 Lahtrite klassifitseerimine

Ristsõnad koosnevad vihjetest, vastuselahtritest ning illustratsioonidest. Töö raames jagati pildilt välja lõigatud ruudud 4 erinevasse klassi. Klasside iseloomustus on toodud järgnevas loetelus:

1. **Vihje:** sisaldab teksti, külgneb paremalt **või** alt vastuselahtriga
2. **Topeltvihje:** sisaldab teksti mitmes osas, külgneb paremalt **ja** alt vastuselahtriga
3. **Vastuselahter:** täitmata lahter
4. **Ebarelevantne:** ei klassifitseeru teiste klasside alla, näiteks osa pildist või üleni must ruut

Ruutude klassifitseerimine toimus mitmes etapis. Esimeses etapis arvutati lahtri heledus järgneval meetodil. Lahter viidi hallskaalale, arvutati keskmine piksli väärtus ja jagati see 255-ga. Selle tulemusena saadi heleduse väärtuseks arv vahemikus 0-1, kus 0 tähistas üleni musta ning 1 üleni valget pilti. Katsete tulemusena selgus, et lävendi 0,65 juures on tõenäoline, et sellest heledamad pildid sisaldavad vihjeid või on vastuselahtrid. Seega loeti klassi Ebarelevantne kõik ruudud, mille heledus jäi alla antud läve. Vihjet sisaldavad ning vastuselahtrid on üldjuhul heleda taustaga, mistõttu on nende heledus kõrgem.

Ülejäänud lahtrid jaotati järgnevalt. Kui pildilt teksti ei tuvastatud, klassifitseeriti see Vastuselahtriiks. Kui pildilt tuvastati tekst, loeti lahter Vihjeks. Juhul kui tekst sisaldas topeltreavahet, oli tegu Topeltvihjega.

Vahel tekkisid esimeses etapis vead. Näiteks kui ristsõna illustreerival pildil oli väga hele laik, võidi seda lugeda Vihje või Vastuselahtri klassi alla. Mõnikord ei leidnud tekstituvastus topeltreavahet ning luges mitut vihjet sisaldava ruudu tavalise Vihje klassi alla.

Vigade parandamiseks puhastati teises etapis klassid järgnevalt. Iga Vihje ja Topeltvihje ruudu jaoks loeti üle neist paremal ja all asuvate järjestikuste Vastuselahtrite arv. Juhul kui paremal ega all ei leidunud ühtegi Vastuselahtrit, seati ruut Ebarelevantssesse klassi. Kui ruudust paremal ja all asus Vastuselahtreid, muudeti Vihje Topeltvihjeks.

4.3 Vastusekandidaatide genereerimine

Suur osa ristsõnaküsimustest põhineb sünonüümidel ja faktiteadmistel. Seetõttu valiti sobivate sõnade otsimise allikateks Eesti Wordnet, Vikipeedia ning eeltreenitud Word2Vec mudel. Järgnevates alapeatükkides tuuakse täpsem ülevaade antud allikatest ning sellest, kuidas neid vastusekandidaatide genereerimisel kasutatakse.

4.3.1 Eesti Wordnet

Eesti Wordneti (EstWN) andmebaas on üles ehitatud sünohulkadele, mis hoiavad endas kõiki ühe mõiste kohta käivaid sõnu [5]. Sünohulgad on omavahel ühendatud viidete kaudu, mis kirjeldavad mõistete semantilisi või leksikaalseid suhteid.

Ristsõnades esitatakse sageli küsimusi, mis ootavad vastuseks samatähenduslikke või lähedasi sõnu. EstWN on kasulik vahend sellistele küsimustele vastuste leidmiseks. Esmalt lemmatiseeriti vihje EstNLTK lemmatiseeriija abil. Seejärel käidi läbi kõik andmebaasi poolt tagastatud sünohulgad. Võimalike vastusekandidaatide hulka loeti iga sünohulka kuuluv sõna, lisaks sellele nendega lähedalt seotud sõnad ning hüponüümid ehk alammõisted.

Leidub vihjeid, kus vastuseks oodatakse puuduvat sõnaosa. Enamasti tähistatakse puuduvat osa kolme punktiga. Et sellistele küsimustele vastata, koguti kokku EstWN-s leiduv sõnavara. Tsükliliselt käidi üle kõikide sünohulkade ning salvestati leitud mõistete algvormid. Kandidaadiotsingul valiti sõnastikust välja kõik etteantud mustriiga ühtivad sõnad. Näiteks vihje „...gramm“ puhul leiab teiste hulgas sõnad „anagramm“ ja „kilogramm“.

4.3.2 Eesti Vikipeedia

Sageli on ristsõnades vaja faktiteadmisi, mida sõnastikust ei leia. Nendele küsimustele vastuste leidmiseks kasutati allikana Eesti Vikipeediat, kuhu sai päringuid esitada

Vikipeedia API¹⁴ kaudu. Vikipeediapäringu esitamiseks lemmatiseeriti vihjetekstid. Päriti maksimaalselt 10 vihjetekstiga seotud artiklit, mille pealkirjade hulgast genereeriti vastusekandidaadid.

Tagastatud pealkirjades esines tähemärke, mis ei kuulu eesti tähestikku. Sellistele tähemärkidele leiti lähim eesti tähestikku kuuluv vaste Unidecode¹⁵ mooduli abil. Lisaks võõrastele tähemärkidele eemaldati kandidaadikirjete hulgast kirjavahemärgid, numbrid ning üleliigsed tühikud. Mõnikord oli Vikipeediast pärinevatele vastetele sulgudesse lisatud täpsustav märg. Puhastamisel sulgude sisu eemaldati. Vastusekandidaatide hulka loeti mitmesõnaliste Vikipeedia-vastete korral sõnad kokkukirjutatult, kuna ristsõnades vastustesse tühikuid ei oodata. Samuti loeti kandidaatide sekka ka eraldiseisvad sõnaosad, kuna näiteks nimede puhul oodatakse vahel vastuseks vaid ees- või perenime. Seeläbi suurendati tõenäosust, et mõni leitud kandidaatidest on oodatud vastus.

4.3.3 Lühendid

Üldtuntud lühendid on EstWN andmebaasis toodud ning on leitavad otsingu kaudu. Sageli küsitakse ristsõnades ebatavalisi lühendeid. Näiteks on vihjena antud inimese täisnimi ning vastuseks oodatakse tema nimetähti. Selliste juhtumite katmiseks genereeriti vihje iga sõna esitähtedest lühend. Juhul kui vihjes olevate sõnade arv kattus oodatud vastuse pikkusega, lisati genereeritud lühend vastusekandidaatide sekka.

4.3.4 Word2Vec

Word2Vec on 2013. aastal Mikolov jt poolt välja pakutud viis sõnade vektorruumis kujutamiseks [24]. Vektorruumis asuvad teineteise lähedal sõnad, mida sageli koos kasutatakse. Sõnavektorid õpitakse suure tekstikorpuse põhjal.

¹⁴Vikipeedia teek: <https://pypi.org/project/wikipedia/>

¹⁵Unidecode teek: <https://pypi.org/project/Unidecode/>

Eesti keele koondkorpusel on treenitud mitmeid Word2Vec mudeleid [25]. Koondkorpus sisaldab tekste ajalehtede, ilukirjanduse, teaduse ning seadusandluse valdkondadest. Avaldatud on nii sõna originaalvormidel kui ka lemmatiseeritud vormidel treenitud mudelid.

Sageli on ristsõnades kirjeldavas laadis küsimused, mida teistest allikatest otse ei leia. Samuti küsitakse vahel mingi sõnaga sageli koos esinevat sõna. Antud küsimustele on võimalik leida vastuseid, otsides Word2Vec'ist algvormis vihjetekstile vektorruumis lähedal asuvaid sõnu. Funktsioon *mostSimilar()* võimaldab vektoreid kombineerides leida vasteid mitmesõnalistele vihjetele. Näiteks tagastab Word2Vec vihjele „kahe küüruga loom“ muuhulgas sõna „kaamel“.

4.3.5 Kogutud andmete eeltöötlus

Kõikidest allikatest leitud kandidaadid puhastati. Puhastamisel eemaldati kirjavahemärgid, numbrid ja tühikud. Igale kandidaadile anti kaasa ka info allika kohta, mille kaudu kandidaat leiti.

Vastusekandidaatide seast eemaldati kõik kandidaadid, mille pikkus ei ühtinud oodatud vastuse pikkusega. Igale kandidaadile määrati kaaluks 1, mille järgi oli hiljem võimalik hinnata lahenduse headust. Mida suurem oli lahenduse kogukaal, seda suurem hulk küsimusi oli leidnud lõpplahenduses vastuse.

Juhul kui sama kandidaat leiti mitme allika järgi, on suurem tõenäosus, et tegu võiks olla õige vastusega. Sellise kandidaadi kaal tõsteti 1,1-le.

4.4 Kandidaatide ruudustikku sobitamine

Kui kõikide vihjete jaoks olid pikkuselt sobivad vastusekandidaadid väljafiltreeritud, hakati kandidaate ruudustikku sobitama.

Esmalt pandi paika lahendamisel vihjete läbimise järjekord. Vihjenimekirjast eemaldati vihjed, millele ei leitud ühtegi vastusekandidaati. Esimeseks määrati kõige pikema

vastusega vihje. Järgmiseks valiti sellega ristuv vihje. Edasi valiti vihje, mis eelnevalt valitud vihjetega kõige rohkematest kohtadest ristus. Sarnast mõttekäiku jätkati, kuni kõik vihjed olid järjekorda lisatud. Juhul, kui ristuvaid vihjeid oli mitu, valiti see, millele oodati kõige pikemat vastust.

Vihjete läbimise järjekorra määramine aitas vähendada lahendamiseks kuluvat aega. Ristuvad kohad tekitavad tähepiiranguid, mis aitavad välistada võimalikke vastusekandidaate.

Ristsõna täitmisel põhineti sügavuti otsingule (ingl *depth first search*) [26]. Kõigepealt valiti vihjenimekirjast esimene vihje ning seati tema vastuseks esimene kandidaat. Järgmise vihje vastuseks valiti kandidaat, mis sobitus esimesest vastusest tekkinud tähepiiranguga. Olukorras, kus ükski kandidaat ei sobinud või lubatud vastuse tühjaks jätmise võimaluste arv oli täitunud, tagastati sinnamaani täidetud ristsõna. Iteratiivselt liiguti eelneva vihje juurde tagasi ning muudeti sealset vastust. Otsiti lahendust, kuhu sobituks võimalikult paljud kandidaadid. Mõnede tühjade vastuste lubamine oli vajalik, kuna leitud vastusekandidaatide hulgas ei pruukinud olla õiget vastust.

Leitud lahendusi hinnati mitmel viisil. Igale lahendatud ristsõnale arvutati skoor, liites kokku vastuste kaalud. Ristsõna pole võimalik hinnata vaid vastatud küsimuste koguarvu järgi. Sellisel juhul pakub algoritm lahenduseks variandi, kus vastused on antud vaid ühes sihis küsimustele, kuna nii ei takista tähepiirangud vastuste valimist. Seega loeti parimaks lahenduseks ristsõna, milles leidus kõige rohkem omavahel ristuvaid vastuseid. Kui neid oli mitu, vaadeldi ristsõna skoori.

4.5 Veebirakendus

Käesolevas peatükis selgitatakse, kuidas töö käigus loodud veebirakendust kasutada. Veebirakenduse lähtekood koos ülesseadmisjuhendiga on kättesaadaval GitHubis¹⁶.

¹⁶Link lähtekoodile: <https://github.com/inesanett/ristsonalahendaja>

Rakenduse avavaates on võimalik üles laadida pilt uuritavast ristsõnast. Pildi üleslaadimisel tuvastab programm ristsõnaruudustiku ning ruudustikus leiduvad vihjetekstid. Kui programm on edukalt ristsõnaruudustiku tuvastanud, kuvatakse kasutajale üles laetud ristsõnapilt. Pildile lisatakse info tuvastatud ruutude kategoriseerimise kohta. Selle järgi on võimalik hinnata, kui hästi programm etteantud pildilt ruudustiku tuvastab.

Pildi juurde kuvatakse nupp "Lahenda ristsõna", millele vajutades otsib programm ristsõnasse sobivaid kandidaate ning genereerib nende põhjal võimalikke ristsõnalahendusi. Lubatud on ka osalised lahendused, kuna sageli ei leia programm igale küsimusele õiget vastust ning ei suuda tervet ristsõnaruudustikku täita. Kasutajale kuvatakse ristsõna koos parima programmi leitud lahendusega. Pildi kõrval asuvast rippmenüüst on võimalik valida kümne parima lahenduse vahel.

Lahendamise protsessi kuuluvad pilditöötlus, tekstituvastus, kandidaatide genereerimine ja nende ruudustikku sobitamine on kõik arvutuslikult keerulised. Seetõttu võivad eelnevalt kirjeldatud sammud võtta mitmeid minuteid.

5 Programmi testimine ja tulemused

Käesolevas peatükis hinnatakse, kuidas rakenduse erinevad komponendid toimisid. Järgnevates alapeatükkides toodud näited põhinevad töö autori poolt läbiviidud testimise tulemustel.

Testimiseks koguti kokku 10 erinevat ristsõnapilti. 5 ristsõna saadi Nuti ristsõnade veebilehelt¹⁷, kus olevad ristsõnad on samas formaadis. See tähendas, et ruudud olid sama suured, vihjetekstides oli kasutusel ühtlane font, ristsõnaruudustiku jooned olid paralleelsed pildi servaga. Sellistel ristsõnadel toimib rakendus kõige paremini. Internetist leiti veel üks veidi teistsuguse formaadiga Nuti portaalist pärinev ristsõna ja Postimehe veebiväljaande ristsõna. Mõlemad olid ühtlase ruudustikuga ja olid programmile etteandmiseks sobilikud. Internetiotsingul leiti ka erineva suuruse ja kujuga lahtritest koosnev ristsõna. Lisaks tehti telefoniga pildid kahest trükitud ristsõnakogumikku kuuluvast ristsõnast. Iga pilt läbis rakendusse kuuluvad etapid. Analüüsi saadud vahe- ja lõpptulemusi.

Rakendust testisid töö lõppjärgus lisaks autorile veel 3 inimest. Alapeatükis 5.5 kirjeldatakse testijatele antud ülesandeid. Samuti on välja toodud testijatelt saadud hinnangud ja soovitusel.

5.1 Ruudustiku tuvastamine

Ruudustiku tuvastamisel tehti eeldus, et kõik ristsõnalahtrid on sama suured ja sama kujuga. Seetõttu tuvastati ruudustik õigesti juhtudel, kui ruudustik oli ühtlane, selgelt nähtav ja pildi serva suhtes joondatud.

Mõnikord esineb ristsõnades erineva suuruse ja kujuga lahtreid. Üks näide sellisest

¹⁷Nuti ristsõnade veebileht: <https://ristsonad.ee/>

ristsõnast¹⁸ on toodud joonisel 3. On näha, et ruudukujulisi lahtreid, mida oli ristsõnas kõige rohkem, tuvastas programm õigesti. Ebastandardset laiad lahtrid on tuvastamisel tükeldatud.

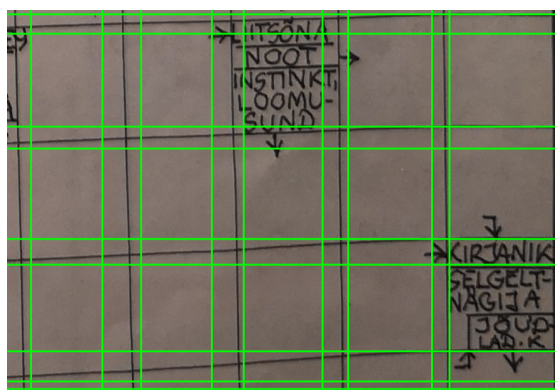
Hispaania neti- domeen	→		Inlila püha ahv 1945 natsiväe kapitulatsioon a kirjastanud natsikindral	→
Kesksugu	→	3 Poola kuninga nimi Veelind	→ ↓	
Holmium	→	↓		Vana- Kreeka
Kaarlimäng				roivas Bändiplaat
Eesti helilooja (1889-1920)	→	↓		↓

Joonis 3. Ruudustiku tuvastamine ristsõnast, milles esineb ebaühtlase kuju ja suurusega lahtreid. Tuvastatud lahtrid on märgitud roheliselt.

Sageli jääb trükitud ristsõnast pildi tegemisel ristsõna kaamera suhtes kaldu. Samuti on jooned erineva paksusega ja trükkimisest võib tekkida katkendlikke ning erineva heledusega jooni. Antud aspektid raskendavad ruudustiku tuvastamist. Joonisel 4 on toodud näide ristsõnast, mille korral leiab programm palju üleliigseid omavahel veidi nihkes olevaid lahtreid. Olukord tekib, kuna rakendus eeldab, et ristsõnaruudustik on pildi servade suhtes joondatud.

Ristsõnaruudustiku valesti määramine põhjustab ka järgnevate komponentide ebaõnnestumise. Etteantud kümnest pildist ei tuvastatud ruudustikku õigesti kolmel korral. Seetõttu neid ristsõnu järgnevates punktides edasi ei analüüsitud.

¹⁸Eesti meistrivõistluste ristsõna, autor Rauno Pärnits, ristsõna pärineb Ringvaate Facebooki leheküljelt 25.01.2012 postitusest <https://www.facebook.com/ringvaade/> (14.05.2024)



Joonis 4. Ruudustiku ebaõnnestunud tuvastamine ebaühtlase ruudustikuga pildilt. Tuvas-
tatud lahtrid on märgitud roheliselt.

5.2 Teksti tuvastamine

Tekstituvastuse tulemuste hindamiseks valiti kolm erinevat ristsõna. Kaks neist võeti Nuti ristsõna leheküljelt. Üks neist oli halvema kvaliteediga, natuke udune. Kolmandaks valiti Postimehe veebiväljaandest pärinev ristsõna. Analüüsides vaadeldi kõiki vihjeid sisaldavaid lahtreid ning silmaga näha olevat teksti võrreldi programmi tagastatud vihjetekstiga.

Üks Nuti portaali ristsõna ja Postimehe ristsõna olid hea kvaliteediga. Tekst tuvastati neilt üpris edukalt. Kokku leidis neis 71 vihjeteksti sisaldavat lahtrit, millest 58 tuvastati täpselt õigesti ning 13 sisaldasid vigu. Üksikud vead tulenesid sarnaste tähtede tuvastamisel tekkinud kirjavigadest. Põhiliseks veaks oli see, et programm poolitas mitut vihjet sisaldava lahtri teksti valest kohast.

Kehvema kvaliteediga Nuti portaali ristsõna analüüsides ilmnas, et tekstituvastuse võimekus väheneb oluliselt pildi kvaliteedi halvenedes. Programm suutis 34 vihjet sisaldava lahtri seast teksti õigesti tuvastada vaid 3 korral. Ühel korral ei leidnud tuvastaja ruudust üldse teksti, ülejäänud sisaldasid kas kirjavigu või suvalisi täheühendeid.

Järgnevalt on näha, mida tagastab tekstituvastus erinevate sisendpiltide korral. Joonisel 5 on toodud näited, kus väljalõikel on silmaga selgelt loetav tekst ning kus ka mudel

leiab pildilt täpselt õige tulemuse. Mudel tagastas teksti koos reavahetusmärkidega, juhul kui kahe teksti vahel on suurem vahe, tagastati kaks järjestikust reavahetusmärki. Joonisel 5a toodud näidete korral tuvastas programm õigesti, et pildil olev tekst on kahes osas.

NÄPUEHE

HENDRIK ...-
SALLER

(a) NÄPUEHE

HENDRIK ...-
SALLER

18
KORDA
VÄIKSEM

(b) 18

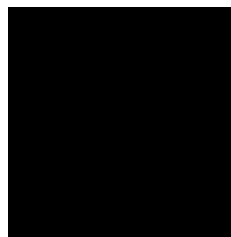
KORDA
VÄIKSEM

Joonis 5. Näited, kus tekstituvastusprogramm tuvastas piltidelt õige teksti. Piltide all on toodud leitud tekst.

Mudel sai üldjuhul õigesti aru ka sellest, kui pildil teksti ei leidunud. Joonisel 6 asuvate piltide analüüsimisel tagastas mudel tühja sõne, mis ühtis oodatud tulemusega.



(a)



(b)

Joonis 6. Näited, kus programm tuvastas õigesti, et pildil ei ole teksti.

Teksti tuvastamisel tuli ette üksikuid juhtumeid, kus mudel ei tuvastanud teksti, kuigi pildil oli tekst selgelt näha. Paar sellist näidet on toodud joonisel 7. Seevastu leidis mudel

vahel teksti piltidelt, kus midagi kirjas ei olnud. Joonisel 8 on näha näited, mille korral mudel tagastas seosetu teksti.

SIMSI
RODO-
DEND-
RON

(a)

LUULE-
TAJA
DORIS ...

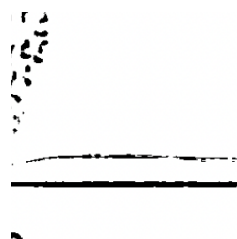
(b)

Joonis 7. Näited, kus programm ei tuvastanud piltidelt teksti, kuigi see oli seal olemas



(a) Mamadfi

ka



(b) Mh)

a" Kali

Joonis 8. Näited, kus programm tuvastas pildilt teksti, mida seal tegelikult ei olnud. Piltide all on toodud leitud tekst.

5.3 Lahtrite klassifitseerimine

Lahtrite klassifitseerimise tulemusi hinnati 7 ristsõna põhjal, mis ruudustiku tuvastamise etapi edukalt läbisid. Tulemuste põhjal moodustati segadusmaatriks, et näha, milliste ruutude klassifitseerimisega eksitakse. Tulemused on toodud tabelis 2.

Tulemused on üldiselt head, kuid programm teeb ka mõningaid vigu. Vead tekkisid põhiliselt siis, kui ristsõna illustreerivate osade sees oli heledam ala või kui vihjelahtrist

Tabel 2. Ristsõnalahtrite klassifitseerimise tulemustel põhinev segadusmaatriks.

		Ennustatud lahtritüüp			
		Vihje	Topeltvihje	Vastuselahter	Ebarelevantne
Tegelik lahtritüüp	Vihje	159	4	5	0
	Topeltvihje	0	103	1	0
	Vastuselahter	1	0	802	0
	Ebarelevantne	1	0	4	185

teksti ei tuvastata. Heledad alad määrati Vastuselahtri või Vihje klassi. Juhul kui ala asus illustratsiooni sees, ei mõjutanud see lahendamise tulemust. Vastasel juhul võis viga mõjutada vihjele oodatud vastuse pikkust.

Joonisel 9 on toodud väljalõige ristsõnast, millele on kuvatud lahtrite tüübid. Rakendus loeb algselt Vihjete alla ka „Vastus algab“ ja „Vastus jätkub“ lahtrid, kuid need eemaldatakse vihjenimekirjast enne vastusekandidaatide genereerimise etappi.

	X	X	X	X	X
	SUVEÕISES MUUSIKAS ...				
X	LÜHIKE FILMIV KIRJUTIS	SULGRIIK	UVE AENE	ROVIT-TAM	
VASTUS ALGAB	0	0	0	0	
SIMSI RIGI-O-DEMO-RION	0	0	0	0	
VASTUS JÄTKUB	0	0	0	0	
FLVDR	0	UUTA BÜST	0	0	
		JÄRVA MAAL			

Joonis 9. Rakenduse poolt klassifitseeritud lahtrid. Kategooriad on märgitud järgmiselt: V-Vihje, W-Topeltvihje, O-Vastuselahter, X-Ebarelevantne. Mustad täpid tähistavad lahtrite nurki.

5.4 Pakutud ristsõnalahendused

Ristsõna pakutud lahenduste hindamiseks uuriti kõigepealt, mitu vastusekandidaati igale vihjele leiti. Analüüs põhineb eelnevas peatükis kirjeldatud 7 ristsõnal. Ristsõnades oli kokku 361 vihjet, nendest 278-le leiti vähemalt üks vastusekandidaat. Keskmine leitud kandidaatide arv vähemalt ühe kandidaadiga vihjete kohta oli 3,6. Mediaanväärtus oli 2.

Automaatselt genereeritud vastusekandidaadid olid väga erineva kvaliteediga. Selle hindamiseks täideti käsitsi ära kolm ristsõna (2 Nuti ja 1 Postimehe ristsõna), et leida igale vihjele õige vastus. Vaadeldi igale vihjele väljapakutud kandidaate ning loeti kokku, mitmel korral esines vastusekandidaatide hulgas õige vastus. Leitud õigete vastuste arv varieerus ristsõnade vahel. Tulemusena leiti ristsõna kaupa õiged vastused 20-le 42-st vihjest, 12-le 38-st vihjest ning 7-le 54-st vihjest. Oli näha, et õiged vastused pärinesid erinevatest allikatest. See näitab, et iga valitud andmeallikas oli lahendamisel kasulik.

Vastusekandidaatide leidmist raskendab sageli see, kui ühes lahtris on mitu vihjet, kuid vihjetekst poolitatakse vihjete vahel vales kohast. Samuti raskendavad vastuste leidmist tuvastamisel tehtud kirjavead.

Näide Nuti ristsõnale¹⁹ programmi poolt pakutud lahendusest on toodud lisas 1. On näha, et rakendus suudab osaliselt loogilisi vastuseid pakkuda. Pakutud on ka vastuseid, mis ei sobi vihjetekstiga üldse kokku.

Sageli jääb suur osa ristsõnast täitmata. Seda põhjustab nii sobivate kandidaatide puudumine kui lahendamise algoritm. Algoritmile on seatud piirang, mitu vastust võib tühjaks jätta enne lahenduse tagastamist. Juhul kui lubatud tühjade vastuste arv tõstetakse liiga kõrgeks, tekib vastusekandidaatide ruudustikku sobitamise võimalusi rohkelt juurde. Kõikide võimaluste läbi käimine võtaks väga kaua aega.

¹⁹Ristsõna autor on Tõnu Paldra, ristsõna pärineb Nuti ristsõnade leheküljelt <https://ristsonad.ee/r/tasuta-ristsona/> (02.05.2024)

5.5 Tagasiside veebirakendusele

Programmi proovisid arendamisprotsessi lõpufaasis kolm testijat. Testijatele tutvustati programmi ülesehitust ning paluti katsetada programmi funktsionaalsusi vabalt valitud ristsõnapiltide peal. Ette anti järgnevad juhised:

1. Hinda programmi kasutajamugavust.
2. Kirjelda, millistel pildidel rakendust katsetasid.
3. Pildi üleslaadimisel kuvatakse info ruudustiku tuvastamise ja lahtrite klassifitseermise kohta. Hinda, kas programm suutis pildilt ristsõnaruudustiku tuvastada ning kas lahtrite klassifitseerimine toimib ootuspäraselt.
4. Kuidas hindad tagastatud lahenduste kvaliteeti?
5. Üldised kommentaarid rakendusele.

Testijate kommentaaridest ilmnes, et veebirakenduse ülesehitus on loogiline, kuid kasutajamugavuse parandamiseks on arenguruumi. Näiteks tuli välja, et väga suurte mõõtmetega ristsõna puhul on seda lehelt ebamugav vaadata. Ristsõna ei mahu korraga ekraanile ning kasutada tuleb kerimisribasid. Samuti mainiti, et kuna protsessid võtavad kaua aega, võiks lehele kuvada hinnangu kui kaua tulemise ekraanile ilmumiseks läheb. Testijad leidsid olukordi, kus vigade käsitlemine oli puudulik ning veebilehele kuvati automaatne veateade. Lõppversioonis parandati veateadete haldamist.

Testijad mainisid, et rakendust katsetati internetist leitud ristsõnadest tehtud kuvatõmmistel. Leiti, et kehva kvaliteediga pildile lahendusi ei leita.

Öeldi, et programm tuvastab enamasti vihjed ja täitmist vajavad ruudult pildilt edukalt. Leiti, et juhul kui ristsõnast tehtud pildil ei ole näha ristsõna välimist äärt, ei tuvastata äärmisi lahtreid.

Üldjuhul suutis programm ristsõnale loogilisi lahendusi pakkuda. Mainiti, et tagastatud lahendusse jääb palju vastamata küsimusi. Paljud ühe ristsõna kohta leitud

lahendused erinesid vaid paari sõna võrra ning öeldi, et võiks olla võimalik vaadelda rohkem erinevaid lahenduskäike.

Lisakommentaaride hulgas toodi välja, et veebileht võiks olla interaktiivsem. Näiteks võiks jooksvalt kuvada vastuste ruudustikku paigutamise protsessi. Mainiti ka seda, et kuigi rakendus ei suuda ristsõna täielikult täita, on tulemus muljetavaldav.

5.6 Edasised arenguvõimalused

Töö põhieesmärk, milleks oli luua automatiseeritud eestikeelsete ristsõnade lahendaja, sai täidetud. Leidub mitmeid viise, kuidas loodud veebirakendust edasiarendada.

Antud töö käigus loodud ristsõnalahendaja eeldab, et kõik ristsõnalahtrid on sama suured ning riskülikukujulised. Eeldatakse, et igasse vastuselahtrisse käib üks täht. Osades ristsõnades on lahtrid näiteks ringi- või kärjemustrikujulised. Samuti leidub vahel vihjelahtreid, mis on teistest lahtritest suuremad, et pikem vihjetekst ära mahuks. Selliseid lahtreid saab tuvastada näiteks OpenCV *findContours()* funktsiooni abil. Et ebastandardse kuju ja suurusega lahtritega ristsõna tuvastada, peaks leidma uue viisi lahtrite omavahelise asukoha määramiseks.

Sageli on telefoniga tehtud pildid moonutatud. Näiteks avatud raamatulehekülje raamatuselja poolne osa on kõrgendatud ja teine äär madalamal. Seetõttu on sageli ristsõnakogumikust telefoniga pilte tehes ruudustik kaldus. Antud töös rakendatud ruudustikutuvastus suudab eraldada lahtreid vaid perfektselt joondatud ruudustikust. Moonutustega piltidelt ruudustiku leidmiseks on võimalik pilte eelnevalt töödelda näiteks *page-dewarp*²⁰ teegi abil.

Tekstituvastusmudel ei suuda hetkel ära tunda vastuselahtritesse lahendaja poolt käsitsi kirjutatud tähti. Nende tuvastamiseks on võimalik kasutada teistsuguseid eeltöötlussamme. Python-tesseract teegi kasutamise alternatiivina on käsikirjas tähti võimalik

²⁰Dewarp teek: <https://pypi.org/project/page-dewarp/>

tuvastada ka mallisobitamise meetodil.

Vastusekandidaatide otsinguks on antud töös kasutatud Vikipeediat, EstWN andmebaasi ning eeltreenitud Word2Vec mudelit. Sageli ei leita neist allikatest õigeid vastuseid. Samuti on vihjed sõnastatud kaudselt, mis raskendab vastuste leidmist. Kandidaatide otsingut on võimalik parandada, lisades uusi allikaid. Näiteks tasub proovida vastuseid pärida veebiotsingust ning tehisintellektilt.

Loodud on mitmeid ingliskeelsete ristsõnade lahendajaid, mis suudavad ristsõnu üpris edukalt lahendada. Tulemused on head, kuna enamik uutes ristsõnades küsitud küsimustest on juba varasemalt ristsõnades esinenud ning on leitavad andmebaasidest. Antud töös oleks vastuste genereerimise etappi võimalik parandada, kogudes kokku varasemalt väljaantud ristsõnades esinevad küsimuste ja vastuste paarid.

6 Kokkuvõte

Magistritöö eesmärk oli luua rakendus, mis pakuks eestikeelsetele ristsõnadele võimalikke lahendusi. Kasutajal on võimalik ette anda ristsõnast tehtud pilt ning talle tagastatakse täidetud ristsõna. Töös uuriti varasemalt loodud ristsõnade lahendamise rakendusi ning nende puudujääke. Samuti kirjeldati ristsõnades leiduvaid vastusetüüpe.

Valminud ristsõnalahendajale pildi etteandmisel tuvastatakse ristsõnaruudustik koos ruudus leiduvate vihjetekstidega. Vihje ja oodatud vastuse pikkuse järgi genereeritakse andmeallikate põhjal võimalikud vastusekandidaadid. Andmete kogumiseks kasutati Eesti Wordneti, eestikeelset Vikipeediat ja eeltreenitud Word2Vec mudelit. Vastusekandidaadid paigutatakse ruudustikku ning kasutajale kuvatakse parimate lahenduste põhjal täidetud ristsõnad.

Teadaolevalt ei ole varasemalt automatiseeritud eestikeelset ristsõnalahendajat loodud, mis tuvastaks ristsõna pildilt ning tagastaks kasutajale võimaliku lahenduse.

Rakenduse funktsionaalsust testiti 10-st erinevast ristsõnast tehtud pildidel. Rakendus suudab ristsõna edukalt tuvastada kõrge kvaliteediga pildilt. Ristsõnas peavad olema ühtlase suuruse ja kujuga lahtrid, mis on paralleelsed pildi servaga. Tekstituvastuse hindamiseks valiti 71 lahtrit, mis sisaldasid vihjetekste. Neist 58 puhul tuvastati tekst õigesti ning 13 tagastatud tekstidest sisaldasid vigu. Lahtrite klassifitseerimine oli üldjuhul edukas. Enamasti ei takistanud klassifitseerimisel tehtud vead edasisi samme. Vihjetele vastusekandidaatide leidmise edukus sõltus konkreetsest ristsõnast. Parimal juhul leidis kandidaatide seas õige vastus umbes pooltel vihjetel. Tagastatud ristsõnalahendused sisaldavad nii õigeid kui ka valesid vastuseid. Sageli jääb osa ristsõnast täitmata. Seda põhjustab lahendamise algoritm ja see, et paljudele vihjetele ei leita ühtegi kandidaati.

Veebirakendust testis lisaks autorile 3 inimest. Rakendust katsetati internetist leitud ristsõnadel. Ristsõnaruudustik tuvastati enamasti edukalt. Probleeme tekkis kehvema kvaliteediga piltide kasutamisel. Tagastatud lahendused sisaldasid loogilisi vastuseid. Küll aga jääb enamasti osa ristsõnast täitmata.

Rakenduse komponendid toimivad, kuid igal sammul esineb võimalusi edasiarendusteks. Töö eesmärk täideti, kuna magistritöö raames valmis automatiseeritud eestikeelsete ristsõnade lahendamise programm.

Viidatud kirjandus

- [1] „Kas suudad lahendada esimest eestikeelset ristsõna?“ *Eesti Rahvusraamatukogu blogi* (veebruar 2024). URL: <https://www.rara.ee/uudised/kas-suudad-lahendada-esimest-eestikeelset-ristsona/>. Kasutatud 14.04.2024.
- [2] A. Villmann. *Tarmo Tuule: Ristsõnakoostaja amet on maailma parim*. Oktoober 2014. URL: <https://dea.digar.ee/article/parnupostimees/2014/10/04/6.1>. Kasutatud 22.04.2024.
- [3] M. Goodwin. *What is an API (application programming interface)?* Aprill 2024. URL: <https://www.ibm.com/topics/api>. Kasutatud 23.04.2024.
- [4] *Front End (In a Website)*. URL: <https://airfocus.com/glossary/what-is-a-front-end/>. Kasutatud 23.04.2024.
- [5] *Eesti Wordnet*. URL: <https://www.cl.ut.ee/ressursid/teksaurus>. Kasutatud 18.04.2024.
- [6] *What is OCR (Optical Character Recognition)?* URL: <https://aws.amazon.com/what-is/ocr/>. Kasutatud 28.04.2024.
- [7] *What is a Web Application?* URL: <https://aws.amazon.com/what-is/web-application/>. Kasutatud 23.04.2024.
- [8] E. Wallace, N. Tomlin, A. Xu, K. Yang, E. Pathak, M. Ginsberg ja D. Klein. „Automated Crossword Solving“. *Association for Computational Linguistics* (2022). arXiv: 2205.09665.
- [9] M. L. Littman, G. A. Keim ja N. Shazeer. „A probabilistic approach to solving crossword puzzles“. *Artificial Intelligence* 134.1-2 (2002), lk. 23–55.



- [10] L. Chen, J. Liu, S. Jiang, C. Wang, J. Liang, Y. Xiao, S. Zhang ja R. Song. „Crossword Puzzle Resolution via Monte Carlo Tree Search“. Teoses: *Proceedings of the International Conference on Automated Planning and Scheduling*. Köide 32. 2022, lk. 35–43.
- [11] M. Ernandes, G. Angelini ja M. Gori. „Webcrow: A web-based system for crossword solving“. Teoses: *AAAI*. 2005, lk. 1412–1417.
- [12] A. Zugarini, T. Röthenbacher, K. Klede, M. Ernandes, B. M. Eskofier ja D. Zanca. „Die Rätselrevolution: Automated German Crossword Solving“ (2023).
- [13] G. Angelini, M. Ernandes, T. Iaquina, C. Stehlé, F. Simões, K. Zeinalipour, A. Zugarini ja M. Gori. „The WebCrow French Crossword Solver“. Teoses: *International Conference on Intelligent Technologies for Interactive Entertainment*. Springer. 2023, lk. 193–209.
- [14] T. L. H. Gammedda ja M. G. N. A. S. Fernando. „End-to-End Automated Crossword Solver using Image Processing, Natural Language Processing and Neural Network“. Teoses: *2022 2nd International Conference on Advanced Research in Computing (ICARC)*. 2022, lk. 43–48. DOI: 10.1109/ICARC54489.2022.9753742.
- [15] P. J. Simha, K. Suraj ja T. Ahobala. „Recognition of numbers and position using image processing techniques for solving sudoku puzzles“. Teoses: *IEEE-international conference on advances in engineering, science and management (ICAESM-2012)*. IEEE. 2012, lk. 1–5.
- [16] S. Kulshreshtha, O. Kovaleva, N. Shivagunde ja A. Rumshisky. „Down and Across: Introducing Crossword-Solving as a New NLP Benchmark“ (2022). arXiv: 2205.09665.

- [17] S. Laur, S. Orasmaa, D. Särg ja P. Tamm. „EstNLTK 1.6: Remastered Estonian NLP Pipeline“. Teoses: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, mai 2020, lk. 7154–7162. URL: <https://www.aclweb.org/anthology/2020.lrec-1.884>.
- [18] A. Rosebrock. *Improving OCR results with basic image processing*. November 2021. URL: <https://pyimagesearch.com/2021/11/22/improving-ocr-results-with-basic-image-processing/>.
- [19] F. P. Miller, A. F. Vandome ja J. McBrewhster. *Levenshtein Distance: Information theory, Computer science, String (computer science), String metric, damerau? Levenshtein distance, Spell checker, Hamming distance*. Alpha Press, 2009.
- [20] *Enlarge low DPI images and PDF scans for better OCR results*. URL: <https://ocr.space/autoscale>. Kasutatud 29.04.2024.
- [21] M. Elmore ja M. Martonosi. „A morphological image preprocessing suite for ocr on natural scene images“. *Georgia Institute of Technology, Princeton University* (2008).
- [22] *Image Thresholding*. URL: https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html. Kasutatud 29.04.2024.
- [23] A. Rosebrock. *Tesseract Page Segmentation Modes (PSMS) explained: How to improve your OCR accuracy*. September 2021. URL: <https://pyimagesearch.com/2021/11/15/tesseract-page-segmentation-modes-psms-explained-how-to-improve-your-ocr-accuracy/>. Kasutatud 13.05.2024.
- [24] T. Mikolov, K. Chen, G. Corrado ja J. Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

- [25] S. Orasmaa, T. Petmanson, A. Tkachenko, S. Laur ja H.-J. Kaalep. „EstNLTK - NLP toolkit for Estonian“. Teoses: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2016, lk. 2460–2466.
- [26] D. C. Kozen. „Depth-First and Breadth-First Search“. Teoses: *The Design and Analysis of Algorithms*. New York, NY: Springer New York, 1992, lk. 19–24. ISBN: 978-1-4612-4400-4. DOI: 10.1007/978-1-4612-4400-4_4.

Lisad

I. Näide lahendatud ristsõnast

						PÕHJA- MAINE PEALINN	KAUP- LUSTE KETT	AMETI- ÜHING	AMPER	LÕPE- TAMA
					VÄÄRIS- KIVI	O	P	A	A	L
					SÕLTU- MATUTE RIIKIDE ÜHEN- DUS	S	R	Ü	ROOMA ÜKS	
					LIITIUM	L	I	VIRUMAA KOLLEDŽ	V	K
					VALDAV		S	K	A	
SUVEÕISES MUUSIKAS ...										
	LÜHIKE PILKAV KIRJUTIS	SUURRIIK	ÜLE- AEDNE	ROHT- TAIM	NELIK + KOLMIK	LINNU- MAGUS TIK...	M	E	S	I
VASTUS ALGAB					U		A	V	A	
SIMSI RODO- DEND- RON					N		ALEVIK PÄRNUMAAL PÕLETATUD MAA	A	R	E
VASTUS JÄTKUB					C			D	ESMAS- PÄEV	E
FLUOR		ÜÜTA BÜST			H	SÕJAVÄE- LAAGER	L	E	E	R
NORD- VEST		ALEVIK JÄRVAMAAL		ANSAM- BEL IN ... ENERGIA- SALVESTI	A			NIKKEL	N	I
IGNO- REERIMA	O	T	H	E	R	S	KALA KÄSI	U		
TSENT- NER			VARSTI ABI- ELLUMA		T		LAURI NEBEL	U		
VÄIKE SÄDE					E		N	S		

Joonis 10. Rakenduse pakutud lahendus Nuti ristsõnale. Ristsõna autor on Tõnu Paldra.

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Ines Anett Nigol**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose
Eesti keeleressurssidel põhinev ristsõnalahendaja,
mille juhendaja on Sven Aller,
reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi
DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks
Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative
Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost
reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja
kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi
ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Ines Anett Nigol

15.05.2024