Sebastian Littman

PDF-TIEDOSTOJEN PARSIMINEN PYTHONIN AVULLA

Opinnäytetyö

Tradenomi

Data-analytiikka

2024

Kaakkois-Suomen
ammattikorkeakoulu

Kaakkois-Suomen ammattikorkeakoulu

Tutkintonimike	Tradenomi (AMK)
Tekijä/Tekijät	Sebastian Littman
Työn nimi	PDF-tiedostojen parsiminen pythonin avulla
Toimeksiantaja	DataLAB
Vuosi	2024
Sivut	28 sivua,
Työn ohjaaja(t)	Atte Reijonen

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli löytää tehokas pythonin sisäinen vaihtoehto parsimaan tekstiä PDF-tiedostoista. Opinnäytteen tuloksia tullaan hyödyntämään Xamkin ReseptiRobotti-hankkeessa, jossa tarkoituksena on luoda teknologian tasoa kartoittava työkalu. Tutkimuskysymyksenä tässä opinnäytteessä oli ”mikä valituista työkaluista parsii PDF-tiedostoja parhaiten?”.

Tutkimus tehtiin laadullisena tutkimuksena, jossa käytettiin kvalitatiivisia menetelmiä, kuten vertailuanalyysiä. Tähän menetelmään päädyttiin, sillä se sopii asioiden vertailuun, kun tarvitsee löytää vaihtoehto, joka edistyy tietyissä piirteissä. Toimeksiantajan kanssa toteutettiin haastattelu, jossa selvitettiin kriteerit työkalun toimivuuden arvosteluun. Työkaluja testattiin aineistolla, joka kerättiin Lens-hakukonetta käyttäen. Toimeksiantaja antoi hakuehdot, jonka avulla tutkimukseen kerättiin perusjoukko. Otosjoukko työkalujen vertailun suorittamiseen poimittiin 10 eri PDF-tiedostoa eri julkaisijoilta perusjoukosta.

Tuloksista otettiin huomiot erilliseen taulukkoon, jotka jaettiin toimeksiantajan kanssa. Näistä tuloksista luotiin vertailuanalyysi, jossa arvioitiin jokaista kriteeriä erikseen ja päädyttiin antamaan jokaiselle työkalulle arvosana jokaisessa kriteerissä. Näin löydettiin työkalu, joka suoriutuu paremmin esimerkiksi nopeudessa. Tulosten pohjalta saatiin selville, että yksivertaisesti parempaa työkalua ei löytynyt, vaan tarvitsi keskittyä virheisiin, joita molemmat työkalut tekivät, ja tämän avulla löytää paras vaihtoehto. Analyysien tuloksena havaittiin kuvatekstien ja diagrammiakselien tekstien aiheuttavan ongelmia tekstin parsinnassa. Havainnoista löytyi myös pienen fonttikoon aiheuttavan ongelmia tekstin parsinnassa.

Asiasanat: Tekstin parsinta, PDF, Python, Vertailuanalyysi

Degree title	Bachelor of Business Administration
Author	Sebastian Littman
Thesis title	Parsing PDF files with Python
Commissioned by	DataLAB
Time	2024
Pages	28 pages
Supervisor	Atte Reijonen

ABSTRACT

The goal of this thesis was to find out what the best PDF-parsing tool within python is. The findings of this thesis are going to be used in Xamk's ReseptiRobotti project, where the goal is to create a tool that charts technology levels. The research question in this thesis was "what is the best tool that can parse PDF files effectively?".

This thesis was conducted as qualitative research. The methods used were qualitative such as benchmarking. This method was used because it fits well when used to compare alternatives, and when one needs to find one that excels in certain aspects. An interview was conducted with the client to find out what aspects are good for benchmarking the effectiveness of the tool. Contenders were tested with the data that was gathered with a tool called lens. The client gave search conditions for what type of PDF files should be used to gather the fundamental set. From this set, 10 PDF files from different publishers were picked for the sample set.

The findings were put on a separate table which was shared with the client. From these findings a comparative analysis, with 4 different criteria, was made and used to compare contenders in each category. With the use of benchmarking, it was examined which tool excels in which criteria, for example speed. It was found that there was no clearly superior tool to use for PDF parsing. Therefore, attention was focused on what kind of mistakes each tool was making. It was found that the text in figures and diagram axes' texts were causing errors to occur in the parsing process. In addition, it was detected that small font sizes were causing problems.

.

Keywords: Text parsing, PDF, Python, Benchmarking

SISÄLLYS

1 JOHDANTO

Nykyaikana Internetissä välitetään tuhansia PDF-tiedostoja (Portable Document Format - tiedosto), ja niistä on tullut yksi yleisimmistä tiedostomuodoista välittää dokumentteja niiden yleisyyden, turvallisuuden, tiedostokoon ja luomisen takia (Trinh 2022). Tämän takia tekstin parsinta PDF-tiedostoista on hyvä keino kerätä dataa, joka kattaa hyvin laajan valikoiman erilaisia aihepiiriä. Parsinta tässä opinnäytteessä tarkoittaa tekstin poimintaa näistä PDF-tiedostoista. Tässä opinnäytteessä keskitytään PDF-tiedostoihin, jotka käsittelevät ympäristön parantamista.

Opinnäytteessä käytetään kahta eri Python-ohjelmointikielen pakettia työkaluna, jotka sopeutuvat tekstin poimintaan PDF-tiedostoista, ja käydään läpi eri yrityksien PDF-muotoisia julkaisua. Työkalujen on tarkoitus poimia teksti tiedostoista. Lopuksi työkalujen poimimaa tekstiä verrataan alkuperäiseen tiedostoon ja arvioidaan laatua molemmilla työkaluilla. Tämän jälkeen tuloksia vertaillaan ja otetaan selvää kumpi työkaluista sopii parhaiten PDF-tiedostojen parsintaan.

Opinnäytetyön tutkimuskysymys on, mikä valituista työkaluista parsii PDF-tiedostoja parhaiten. Opinnäytetyön tavoite on löytää työkalu, jonka avulla on mahdollista parsia teksti PDF-tiedostoista siten, että teksti pysyy mahdollisimman lähellä alkuperäistä tekstiä. Tavoitteena on myös löytää mahdollisia ongelmakohtia, joissa työkalut tekevät virheitä tekstin parsinnassa tai eivät pysty siihen lainkaan.

Lopputulos opinnäytteessä on löytää työkalu, joka käy helposti läpi erilaisia PDF-julkaisuja ja poimii niistä tekstin jatkokäsittelyä varten. Tähän päästään käyttämällä menetelminä teemahaastattelua ja vertailuanalyysiä. Toimeksiantajan projektissa käydään läpi erilaisia patentteja ja julkaisuja, joiden alkuperäinen tiedostomuoto ei välttämättä ole PDF, mutta ne on julkaistu PDF- muodossa. Tässä opinnäytteessä aihetta on rajattu keskittymällä vain julkaisujen PDF-muotoihin, sillä ne ovat laajasti käytetyimmät tiedostomuodot. Opinnäytetyössä työkaluna käytetään Pythonia sen soveltuvuuden, helppokäyttöisyyden

ja laajan internetissä tarjolla olevan ohjeistuksen vuoksi. Opinnäytteen aineisto on otosjoukko PDF-tiedostoja, jotka käsittelevät samanlaista aihetta ja ovat vuodelta 2023. Toimeksiantaja on kerännyt datan käyttäen Lens.org hakukonetta.

2 TEKSTIN PARSINTA DATAN LÄHTEENÄ

Tässä luvussa esittelen hieman PDF-tiedostojen käyttöä ja tarkoitusta. Sen jälkeen keskityn esittelemään tekstin parsinta -metodia. Lopuksi esittelen natural language processingia ja sen yhteyttä tähän opinnäytetyöhön

PDF-tiedostoista on nykyaikana tullut yksi eniten käytetyimmistä tiedostomuodoista, ja niissä kulkee lukematon määrä tekstiä ja erilaista dataa. Tätä tekstiä on mahdollista hyödyntää erilaisten NLP (Natural Language Process) alojen kehittämiseen. Tämä teksti on mahdollista poimia PDF-tiedostoista ja tekstiä on tämän jälkeen mahdollista, jälki käsitellä ja auttaa kehittämään monenlaisia erilaisia ohjelmia ja työkaluja.

2.1 PDF

Adobe loi PDF-tiedostomuodon 90-luvulla Ne ovat turvallisia. Ne näyttävät samalta monessa eri laitteessa, ja niissä on pieni tiedostokoko. Ne ovat helppokäyttöisiä ja voivat sisältää monipuolisesti kuvia, videoita, linkkejä yms.. (Trinh 2022.)

Turvallisuus on tärkeä osa PDF-tiedostomuodon yleisyyttä. Tiedoston jakamisen jälkeen lukijoilla ei ole mahdollista muokata tiedostoa ja vaihtaa sen tärkeitä kohtia. Tämä on tärkeää dokumenteissa, jotka on tarkoitettu jakamiseen, kuten ohjeissa, sertifikaateissa tai hakemuksissa. PDF-tiedostot voidaan myös lukita salasanalla, joka luo niihin vielä lisää turvallisuutta. (Trinh 2022.)

PDF-tiedostolla on myös muita etuja, joiden vuoksi sitä käytetään paljon. PDF-tiedostojen ulkonäkö on samanlainen kaikissa laitteissa. Tiedostot voidaan avata puhelimella ilman, että PDF-tiedoston asettelu muuttuu. Pieni tiedostoko tekee tiedostojen säilyttämisestä kätevää, ja helposti tarjolla olevat mahdollisuudet avata tiedostot, mahdollistavat sen laajan sopivuuden monessa eri

ympäristössä. (Trinh 2022.)

2.2 Tekstin parsinta

Alkuun tarvitsemme tarkennusta mitä parsiminen on käytännössä. Parsiminen tarkoittaa tekstin muuttamista muotoon, joka on tietokoneelle helppo ymmärtää. Tietokone ei normaalisti ymmärrä tavallisilla kirjaimilla kirjoitettua tekstiä, joten se pitää ensin muuttaa toiseen muotoon, jotta tietokone pystyy lukemaan sitä (Wilson 2024.)

Pääpiirteet tekstin parsinnassa on tokenointi, syntaksianalyysi ja semanttinen jäsentäminen (Text parsing). Tokenointi tarkoittaa tekstin muuttamista erillisiin sanoihin ja symboleihin. Näin tietokoneohjelmat pystyvät erottamaan erilliset sanat parsitusta tekstistä (reintech). Greg Wilson (2024) kuvaa kirjassaan tokenoinnin olevan tekstistä parsittavien kirjaimien jakamista kahteen ryhmään. Ensimmäinen ryhmä on tavalliset kirjaimet, jotka ovat monta tokenia peräkkäin, muodostaen sanoja. Toinen ryhmä on kieliopilliset merkit kuten pilkut pisteet ja sulkeet. Nämä luokitellaan yksittäisinä merkkeinä erottaen sanat toisistaan. Tämä muodostaa tekstin parsinnan ytimen kasaamalla kirjain tokeneita yhteen ja huomattaessa symbolin tämä jaetaan erilliseen jonoon. Näin saadaan kasa tokeneita, jotka muodostavat sana- ja symbolitokenit, joiden avulla teksti jaotellaan erillisiin osiin. (Wilson 2024.)

Nopea ruskea kettu hyppää laiskan koiran yli.							
"Nopea"	"ruskea"	"kettu"	"hyppää"	"laiskan "	"koiran "	"yli"	"."

Kuva 1. Sanojen tokenointia

Syntaksianalyysi on tämän jälkeen sanojen erottelua, verbeihin, substantiiveihin ja adjektiiveihin. Syntaksi tarkoittaa sanojen hierarkiaa ja järjestystä, jotta ymmärrämme mistä puhutaan. Tekstin parsinnassa tämä on tärkeä osa tietokoneen ymmärrystä, mitä tekstissä kerrotaan. Tokenoinnin jälkeen parsintatyökalun pitää erottaa, mitä osaa lauseessa sana edustaa. Kuvassa 2 näkyy, kuinka jokainen sana on luokiteltu sanaluokkiin. (Otten 2024)

Nopea ruskea kettu hyppää laiskan koiran yli.							
"Nopea"	"ruskea"	"kettu"	"hyppää"	"laiskan "	"koiran "	"yli"	"."
Adjektiivi	Adjektiivi	Substanttiivi	Verbi	Adjektiivi	Substanttiivi	Prepositio	Välimerkki

Kuva 2 Syntaksianalyysin kuvaus

Semanttinen jäsentäminen on viimeinen tekstin parsinnan pääpilari. Semanttinen jäsentäminen on tekstin tarkoituksen ymmärtämistä. Siinä, missä syntaksianalyysi keskittyy kielioppiin ja rakenteeseen, semanttinen analyysi yrittää ymmärtää tekstin tarkoituksen keskustelussa (tai kontekstissa). Pääpiirteet luokitellaan neljään eri kategoriaan: sanan merkityksen selitteisyys, nimien tunnistus, semanttinen roolimerkintä ja leksikaalinen analyysi. Sanan merkityksen selitteisyys yrittää saada selville, mitä tietty sana tarkoittaa kontekstissa, sillä sanoilla voi olla monta merkitystä. Nimien tunnistuksessa keskitytään erottamaan, milloin puhutaan nimistä tai päivistä. Semanttinen roolimerkintä erottaa sanat subjektiin, objektiin tai predikaattiin. Leksikaalisessa analyysissa yritetään ymmärtää sanaston, kieliopin ja lauseen rakenteen kautta ymmärtämään tekstiä. (Abdullahi 2023.)

2.3 Natural Language Processing

Chowdhury (2020) kuvailee Natural language processingin (NLP) olevan tutkimuksen alue, jossa yritetään selvittää, kuinka tietokoneita voidaan käyttää ymmärtämään luonnollista kieltä. NLP on ollut yleinen tutkimuskohde jo vuosikymmeniä. Liddy (2001) kuvaili NLP:n tavoitteen olevan, että tietokone voi saavuttaa ihmisten tasoisen kielen ymmärryksen.

NLP on koneoppimisen, kielitieteen ja tietojenkäsittelytieteen risteys. Kaikki nämä piirteet tulevat NLP:ssä esille, kun pyritään kehittämään kielenymmärrystä. Suuri ongelma kielenymmärtämisen kehittymisessä on, kuinka monimutkaista ihmisten käyttämä kieli on. Sanojen järjestykselle lauseessa on lukemattomia vaihtoehtoja. Tämä tekee tekstin ymmärtämisestä tietokoneella todella hankalaa. (Donges 2023.)

Tekstin parsinta on ollut yksi tärkeimmistä osista NLP:n kehityksessä. Parhaasta PDF-parsintavaihtoehdosta on tehty erilaisia tutkimuksia. Parsio esittelee PDF-parsintaa ja pohtii, mikä olisi paras työkalu siihen, sillä PDF-parsintaa varten on olemassa erilaisia keinoja. Esimerkiksi pythonissa on monia eri kirjastoja PDF-parsintaan ja ominaisuuksia (, joilla voi) yhdistää OCR-parsintatyökalut kuten Tesseract. Pythonissa on myös mahdollista luoda omiin tarpeisiin soveltuva parsintatyökalu, mutta se vaatii paljon aikaa. (Jane 2023.)

3 TUTKIMUSMENETELMÄT

Kvalitatiivinen tutkimustapa tarkoittaa laadullista tutkimustapaa. Jyväskylän yliopisto tuo esille, kuinka laadullista tutkimusta voidaan toteuttaa monella erilaisella menetelmällä. Menetelmiä yhdistää esimerkiksi esiintymisympäristöön ja taustaan, kohteen tarkoitukseen ja merkitykseen, sekä ilmaisuun ja kieleen liittyvät näkökulmat. (Laadullinen tutkimus. s.a.) Tärkeä piirre laadullisessa tutkimuksessa on, että sen pitää aina olla aineistoon perustuvaa tutkimusta (Juhila 2021).

Toisena tutkimustyylinä pidetään kvantitatiivista tutkimusta. Kvantitatiivista tutkimustapaa voidaan joskus kutsua myös määrälliseksi tutkimustavaksi. Tämä tutkimustapa pyrkii kuvailemaan ja selittämään ilmiöitä havaintojen avulla (Juhila 2021). Jyväskylän yliopisto kuvailee määrällisen tutkimuksen olevan kohteen kuvaamista tilastojen ja numeroiden avulla. Kvantitatiivisia menetelmiä ovat esimerkiksi tilastojen analysointi ja kyselyjen tuottaminen ja niiden tulosten analysointi. Tähän menetelmään tarvitaan perusjoukko, jota tutkitaan ja otosjoukko mistä havaintoja kerätään. Perusjoukko tarkoittaa kokonaista analysoitavaa aineistoa ja otosjoukko on tästä otettu pieni määrä, jota tutkitaan. (Määrällinen tutkimus. s.a).

Tässä opinnäytteessä tehdään laadullista tutkimusta. Puolistrukturoitu teemahaastattelu on kvalitatiivinen menetelmä, joka sopii tähän tutkimukseen. Päädyin tähän menetelmään, sillä sen avulla on mahdollista kerätä laadullista tietoa toimeksiantajalta, joka auttaa opinnäytteen tavoitteen selventämisessä. Toinen tutkimus menetelmä, jonka valitsin tähän opinnäytteeseen on vertailuanalyysi, joka on myös kvalitatiivinen menetelmä. Valitsin tämän, sillä se sopii hyvin erilaisten vaihtoehtojen vertailemiseen ja auttaa löytämään parhaiten

suoriutuvan vaihtoehdon sopii hyvin vertailemaan erilaisia vaihtoehtoja keskenään ja löytää mikä suoriutuu parhaiten.

Puolistrukturoitu teemahaastattelu

Puolistrukturoitu haastattelua pidetään lomakehaastattelun ja strukturoimattoman haastattelun välimuotona. Tarkkaa määrittelyä tämänlaisesta haastattelusta ei ole saatavilla, mutta Hirsjärvi & Hurme (2022) viittaavat moniin eri lähteisiin, jotka kuvailevat puolistrukturoitua haastattelua monilla eritavoilla. Yhdessä tuodaan esille, että kysymysten muoto on sama kaikille, mutta kysymysten järjestys voi vaihdella osallistujien kesken. Toisessa he tuovat esille, kuinka kysymykset voivat olla samat, mutta vastauksia ei ole sidottu vastausvaihtoehtoihin, sen sijaan niihin voidaan vastata omin sanoin. Lopuksi he tuovat esille, että haastattelussa on jokin näkökulmista lyöty lukkoon mutta ei kaikkia. (Hirsjärvi & Hurme 2022, Luku 4).

Vertailuanalyysi

Vertailuanalyysiä kutsutaan myös benchmarkingiksi. Tuominen & Niva (2011) kertovat benchmarkingin olevan järjestelmällinen prosessi, joka soveltuu tuotteiden, palvelujen ja prosessien suorituskyvyn mittaamiseen ja arviointiin. Vertaamalla parhaisiin ja oppimalla parhailta voi kehittää omaa toimintaa. He tuovat esille, kuinka tekijällä pitää olla nöyryyttä hyväksyä, että muut ovat parempia ja heiltä on mahdollista oppia (Tuominen ym. 2011. Sivu 5.) Tämän avulla on mahdollista katsoa, miten oma tuote vertautuu toisten kilpailijoiden toimintaan. Tässä tuodaan esille monia eri kriteerejä ja katsotaan, miten itse sekä kilpailijat vertautuvat kaikissa näissä piirteissä.

Tämä on hyvä käytäntö löytää missä oma yritys tai tuote on hyvä ja missä kilpailijat ovat parempia. Tämän avulla on mahdollista löytää omia kehittämiskohteita. Vertailuanalyysi on myös hyvä vaihtoehto oikean työkalun tai tuotteen valitsemiseen, jos vaihtoehtoja on useampia. Kaikkia valintoja voidaan verrata keskenään vertaamalla hyviä ja huonoja puolia keskenään. Tässä helposti nähdään, missä piirteissä kukin vertailukohde on hyvä ja huono. Tämän jälkeen käyttäjän pitää ottaa selvää, mikä tai mitkä arvostelukriteereistä ovat

tärkeimpiä ja keskittyä siihen vertailukohteeseen, joka suoriutuu näissä parhaiten. Parhaaseen tulokseen pääseminen on tämän jälkeen helpompaa.

4 TOIMEKSIANTAJA

Opinnäytetyön tilaaja on Kaakkois-Suomen ammattikorkeakoulun (Xamk) DataLAB, joka on 2020 perustettu työpaja, joka tarjoaa harjoittelupaikkoja opiskelijoille sekä erilaista tutkimustoimintaa. DataLAB on yhteydessä erilaisiin Xamkin hankkeisiin ja tämän avulla luo sillan opiskelijoiden ja hankkeiden välille. Tässä opinnäytteessä tehdään myös työtä hankkeeseen, jossa DataLAB on osallisena. Hankkeen nimi on ReseptiRobotti hanke. (Xamk 2024). Hankkeessa on myös yhteys Xamkin BioSampo kiertotalous- ja tutkimuskeskukseen.

Tämä opinnäytetyö on osana reseptirobottihanketta, jossa on tarkoitus luoda teknologian tasoa kartoittava työkalu. Hanke on EU:n rakennerahaston rahoittava 2,5 vuoden pituinen hanke. Siinä on tarkoituksena luoda digitaalisesti skaalattava työkalu eri toimialoille materiaalien kierron ja uudelleenkäytön tehostamiseen. Tämä tehostaisi suuresti PK- yrityksiä, joilla ei välttämättä ole resursseja suurien data-aineistojen käsittelemiseen. (Xamk 2024.)

Toimeksiantajalta on tullut pyyntö tehdä tutkimusta ReseptiRobotti-hankkeelle etsimällä hyvä työkalu PDF-tiedostojen parsimiseen. Tämän opinnäytetyön tutkimuskysymys ”Mikä valituista työkaluista parsii PDF-tiedostoja parhaiten?” keskittyy tähän.

5 TYÖKALUT

Lopputulokseen pääsemiseksi tarvitaan ohjelmia ja erilaisia työkaluja, jotka auttavat meitä luomaan kriteerit oikean työkalun löytämiseksi. Tätä varten täytyy ymmärtää, mitä opinnäytetyössä käytettävät työkalut ovat ja mihin niitä käytetään. Tässä luvussa käydään läpi ohjelmointikieli Pythonia ja sillä käytettäviä paketteja sekä tekoälyä, jonka avulla testataan taulukoiden rekonstruktu-rointia ja mistä alkuperäinen aineisto tulee.

5.1 Python

Python on yleinen ohjelmointikieli, joka on ollut käytössä jo yli 30 vuotta. Tähän aikaan oli muitakin ohjelmointikieliä kehitteillä, mutta Python on näistä vielä nykypäivänä ehdottomasti tunnetuin ja laajasti käytetyin. Alkuperäisen version pythonista loi hollantilainen Guido van Rossum. (Python s.a.). Ohjelma sai tämän jälkeen vuosien aikana päivityksiä, mikä paransi sen toimivuutta. Nykyään python on 3.13 versiossa ja saa vieläkin päivityksiä (Wulian 2024). Maailmanlaajuinen käyttö tekee Pythonista hyvin yleisen ja helposti ymmärrettävän ohjelmointikielen, joka helpottaa monien eri yritysten yhteistyötä keskenään. Yhteensopivuus monien erilaisten ohjelmien kesken vahvistaa myös tätä monipuolisuutta.

Yksi Pythonin hyvistä puolista on sen ilmainen saatavuus. Python on mahdollista ladata heidän omilta sivuiltaan täysin ilmaiseksi. Se on mahdollista ladata Windowsille, Macille, Linuxille ja muille vähemmän käytetyille käyttöjärjestelmille. Pythonin käyttöä edistävät myös monet eri paketit, joita on mahdollista käyttää sen kanssa. Näiden pakettien ominaisuudet ovat hyvin monipuolisia ja muokattavia. (Python s.a).

Pythoniin liittyvät kirjastot ja käyttö

Toimeksiantajan kanssa sovittiin kahdesta kirjastosta, joita päätettiin käyttää tässä opinnäytetyössä. Näihin päädyttiin niiden helpon käyttöönoton ja internetissä saatavilla olevan laajan tietomäärän vuoksi. (Toots 2024). Koko prosessi tullaan suorittamaan Jupyter Notebook -ohjelmiston sisällä, joka tekee pythonilla työskentelemisestä huomattavasti helpompaa.

Ensimmäinen työkalu, jota opinnäytteessä päätettiin käyttää, on PyPDF2. PyPDF2 on avoimen lähdekoodin Python-paketti PDF-tiedostojen käsittelyyn, Pääpiirre paketissa on sen soveltuvuus moniin eri ohjelmointitarpeisiin. Sillä voi muokata PDF-tiedostoja monipuolisesti esimerkiksi muuttaa PDF-tiedostoja kuviksi. (Sachideva 2024.)

Toinen työkalu, joka päätettiin ottaa koekäyttöön, on PDFMiner Python kirjasto. PDFMiner tarjoaa keinon tekstin poimintaan PDF-tiedostoista. Tämä paketti sopii hyvin tekstin poimintaan, sillä se pitää tekstin asetelman mahdollisimman lähellä alkuperäistä. PDFMiner myös tarjoaa paljon mahdollisia vaihtoehtoja tekstin parsinnan tyyliin. (Shinyama ym. 2013.)

Jupyter notebook on project Jupyterin luoma verkkopohjainen ohjelmointiympäristö. Tällä ohjelmalla on mahdollisuus luoda notebook-tiedostoja, jotka sisältävät koodia, dataa, visualisointiesityksiä sekä muita ominaisuuksia. Notebook tiedostoja on myös mahdollista helposti jakaa toisille (Project Jupyter s.a.)

ChatGPT

ChatGPT on tekoäly, jonka on luonut OpenAI. Tämänhetkisessä tutkimusvaiheessa ChatGPT on ilmaiseksi käytettävissä, mikä tarjoaa sille hyvin laajan määrän dataa kehittyä. Tekoälyn kanssa on mahdollista käydä ketustelua, ja se vastaa käyttäjälle keskustelumaiseen tapaan. Tämä tekee siitä helpon käyttää. Se ymmärtää ja pystyy vastaamaan käyttäjän esittämiin tarkentaviin jatkokysymyksiin. (OpenAI 2022.)

Tässä opinnäytteessä ChatGPT tekoälyllä on olennainen osa parhaimman PDF-parsintatyökalun löytämisessä. Tätä tekoälyä tullaan käyttämään parsittujen taulukoiden rekonstruktioon. Tällä arvioidaan, kuinka hyvin parsittu teksti ja tekoäly pystyvät tuottamaan alkuperäisen taulukon uudelleen.

Lens

Lens on Lens.orgin luoma avoimen pääsyn hakukone, joka yhdistää metadataa ja data-artefakteja. Lens käyttää neljää erilaista ominaisuutta, jolla se etsii tuloksia. Ne ovat tieteelliset lähteet, patentit, patseq ja collections eli kokoelmat. (Kock 2023.). Tutkimuksessa käytetään Lens hakukonetta keräämään aineistoa tutkimukseen. Aineisto koostuu julkisista julkaisuista, joita etsitään Lensin avulla. Lens etsii aineiston hakutermeihin liittyen ja tarjoaa linkin julkaisuun, josta PDF-tiedosto on mahdollista ladata.

6 OPINNÄYTTEEN TUTKIMUS

Opinnäytteessä tehdään tutkimusta tekstin parsintaan PDF-tiedostoista. Toimeksiantaja pyysi tutkimusta työkaluista, jotka pystyvät parsimaan PDF-tiedostoja. On olemassa monenlaisia työkaluja, jotka ovat toimivia vaihtoehtoja PDF-parsintaan. Tavoitteena on löytää se oikea työkalu, joka onnistuu tietyssä tehtävässä parhaiten tai kaikissa piirteissä riittävän hyvällä tasolla. Toimeksiantaja on tätä varten toivonut tutkimusta ja testaamista. Aineistona tämän työkalun löytämiseen käytetään toimeksiantajan tarjoamaa aineistoa, joka keskittyy ReseptiRobotti hankkeeseen.

6.1 Tutkimusaineisto

Tutkimusaineistona tässä on käytetty monia erilaisia PDF-tiedostoja. Toimeksiantaja käytti Lens.org-hakukonetta etsimään patentteja ja julkaisuja hakutermeillä ”low carbon concrete”. Hakutermit liittyvät ReseptiRobotti-hankkeeseen, jossa on tarkoitus vähentää hiilijalanjälkeä. Opinnäytetyössä keskityttiin vuoden 2023 julkaisuihin, jotta tiedot ovat ajankohtaisia.

Tällä saatiin eri PDF-tiedostoja, joka toimi meidän perusjoukkonamme. Tämän jälkeen päätin rajata julkaisuja pienemmäksi joukoksi, jota olisi helppo käsitellä mutta silti olisivat erikaltaisia tiedostoja. Päädyin valitsemaan 10 eri julkaisua, jotka olivat kaikki eri julkaisijalta. Nämä julkaisut toimivat opinnäytetyössä otosjoukkona.

Kaikki otosjoukon PDF-tiedostot olivat avoimesti saatavia tiedostoja eri julkaisijoilta. Valitsemalla tiedostoja eri julkaisijoilta pystytään, pystytään analysoimaan erilaisella formaatilla luotuja PDF-tiedostoja. Tällä voidaan verrata erilaisia PDF-tyylejä, esimerkiksi kuinka hyvin työkalu suoriutuu sellaisen PDF-tiedoston, jossa teksti on kirjoitettu kahdelle palstalle, lukemisesta.

6.2 Laadullinen tutkimus

Tässä opinnäytteessä tehdään laadullista tutkimusta hyvän PDF-parsintatyökalun löytämiseksi. Opinnäyte luokitellaan tutkimukselliseksi opinnäytteeksi. Tutkimuksellista opinnäytettä kuvaillaan, että siinä tehdään tutkimusta ongel-

man ratkaisuun ammattialasi rajoissa (Karelia AMK). Toimeksiantaja on toivonut tutkimusta työkalun löytämiseen, jota he voivat käyttää hankkeen edetessä. Tästä syntyy ongelma, pyritään tässä opinnäytteessä selvittämään. Tämän ratkaisemiseksi tehdään laadullista tutkimusta kvalitatiivisia menetelmiä käyttäen. Kvalitatiivisia menetelmiä, kuten haastattelu ja vertailuanalyysi käyttäen saamme tutkimukselle laadullista pohjaa.

Työkalun löytäminen etenee vaiheittain. Ensiksi määritellään tavoite, johon työkalun pitää vastata. Tämän jälkeen valitaan oikeat menetelmät työkalun löytämiseen. Lopuksi testataan, miten työkalut onnistuvat käymään otosjoukkoa läpi ja mitä puutteita ja ongelmia työkaluilla vielä on. Tämän havainnollistamiseen luotiin toimeksiantajan kanssa 4 kriteeriä: nopeus, rakenne, taulukot ja virheiden määrä. Näiden kriteereiden avulla työkaluja verrataan keskenään, jotta voidaan löytää paras vaihtoehto.

7 OIKEAN TYÖKALUN LÖYTÄMINEN

Nykyään on paljon erilaisia tekstinparsintatyökaluja, joita voidaan käyttää tekstin parsintaan PDF-tiedostoista. Tarvitsemme vastauksen nyt kriteereihin mitkä olemme luoneet. Näihin kriteereihin tullaan nyt vastaamaan käyttämällä edellä mainittuja menetelmiä tässä opinnäytetyössä. Vastaamalla näihin kriteereihin löydämme oikean työkalun produktion tekemiseen.

7.1 Haastattelu toimeksiantajan kanssa

Opinnäytteessä on tehty puolistrukturoitu teemahaastattelu Xamkin dataLAB:in työntekijän kanssa, joka on myös TKI-asiantuntija ReseptiRobotti-hankkeessa. Haastattelu sopii menetelmäksi, koska toimeksiantajalta haluttiin tietoja aiheen selventämiseksi. Haastateltava valittiin, koska hän on Xamkilla työssä ja koska hän on toiminut yhteyshenkilönä opinnäytetyön tekijän ja dataLABin välillä. Kysymyksillä kerättiin tietoa aineiston alkuperäisyydestä, dataLABista, reseptirobotti hankkeesta, laadun varmistamiseen liittyvien kriteerien valinta sekä lopputuotteen hyödyntämisestä.

Haastattelu toteutettiin verkkohaastatteluna Microsoft Teamsin avulla 7.8.2024. Tässä haastattelussa kysyttiin kysymyksiä yksi kerrallaan aiheeseen liittyen ja haastateltava oli vapaa vastaamaan omalla tavallaan. Haastattelua

ei nauhoitettu, mutta haastattelija keräsi muistiinpanoja ja otti vastaukset ylös jälkikäyttöä varten.

Puolistrukturoitu teemahaastattelu valittiin lähestymistavaksi, jotta haastattelusta saadaan tietoa juuri tietyistä aiheista. Opinnäytteessä koettiin olennaisena osana kerätä taustatietoa, mihin kehitettävää työkalua tullaan käyttämään. Saatiin selville, että tällä hetkellä on monia erilaisia PDF parsintatyökaluja, mutta ei ole kohdistettua tutkimusta tai kokeilua, mikä niistä olisi paras vaihtoehto mihinkin tilanteeseen. ReseptiRobotti-hankeen edetessä tullaan tarvitsemaan tietoa, minkälaista työkalua on hyvä käyttää PDF-tiedostojen parsintaan. Toimeksiantaja pyysi kokeilemaan erilaisia työkaluja ja löytämään, mikä niistä sopisi heidän hankkeeseensa parhaiten.

Toimeksiantajalta kysyttiin, miten he ovat päätyneet valitsemaan tutkimukseen juuri nämä työkalut. Hankkeen TKI-asiantuntija, jota tässä myös haastateltiin, suosii paljon Pythonin käyttöä ja on tehnyt pintatason testausta jo näille työkaluille. Pythonin käyttöä pohjusti myös se, että sitä on helppo käyttää ja molemmilla on kokemusta sen käytöstä. ReseptiRobotti-hankkeessa tullaan myös luomaan lopputulos täysin Pythonilla, joten tämä tutkimus pyydettiin suorittamaan myös Python-kirjastoja käyttäen. Hankkeessa koitetaan rajata näin mahdollisimman paljon puolia Pythonin sisälle.

Haastattelun toinen teema oli työkalun kehittäminen ja löytäminen. Työkalun löytämiseksi toimeksiantajalta kysyttiin, mitkä olisivat hyvät kriteerit, millä työkalua arvostellaan. Tästä saatiin selville vertailuanalyysiin käytettävät kriteerit, jotka ovat nopeus, parsitun tekstin rakenne, taulukkojen rekonstruktio ja virheiden määrä saadussa tekstissä. Näiden mittaamiseen tuli myös ohjeistusta ja riittävän laadun kriteerit, joiden avulla työkalu luokitellaan onnistuneeksi tässä osassa.

Teemana haastattelussa oli myös aineiston valinta ja aineiston alkuperäisyys. Kysyin haastateltavalta, mistä aineisto saatiin ja miten Lens-työkalua käytettiin aineiston hankkimiseen. Aineiston valintaan johti yhteys Xamkin Biosampo-puolelta, jossa halutaan edistää kiertotaloutta, jolla voidaan vähentää ilmastonpäästöjä. Hakuehdot valittiin sen perusteella, millä on paljon mahdollisuuksia hiilidioksidipäästöjen vähentämiseksi. Betoni on iso hiilidioksidinkuorman

lähde, joten sen hiilijalanjäljen vähentämiseen koitetaan etsiä keinoja. Tekstiilijätteen uudelleenkäyttö on myös yksi kohdennus, jota koitetaan parantaa, jolla voidaan myös laskea hiilidioksidi kuormaa. (Toots 2024.)

7.2 ChatGPT taulukon rekonstruktio

Yksi työkalun kriteereistä on mitata, kuinka hyvin se pystyy parsimaan taulukoissa olevan tekstin. Taulukot yleensä sisältävät hyvin paljon dataa. Koska datan poimiminen on tärkeää tämän tutkimuksen kannalta, sen päätettiin olevan yksi tärkeimmistä kriteereistä oikean työkalun valintaan.

Kriteeriä arvioitiin sen perusteella, miten hyvin tekoäly kykeni rekonstruoimaan alkuperäisen taulukon. Tekoäly, johon tässä päädyttiin, on OpenAI:n luoma ChatGPT. Tähän päädyttiin toimeksiantajan mukaan sen helpon saatavuuden ja helppokäyttöisyyden takia. Tekoälyn kanssa on mahdollista käydä keskustelua, jossa sille syötetään parsittu teksti, joka sisältää taulukon datat. Seuraavaksi kysytään, ”pystytkö tekemään tästä tekstistä taulukon”, ja katsotaan, kuinka hyvin tekoäly onnistuu taulukon uudelleenluomisessa. Tekoälyn tuottama taulukko analysoidaan vertaamalla sitä alkuperäiseen. Parhaimmassa tapauksessa luotu taulukko on identtinen alkuperäisen kanssa.

7.3 Python ja sen paketit

Pythonia ja erilaisia paketteja, jotka sopeutuvat tähän projektiin, tullaan käyttämään PDF-tiedostojen parsimiseen tässä opinnäytteessä. Pythoniin päädyttiin sen avoimen saatavuuden, internetissä saatavilla olevan laajan ohjeistuksen ja sen laadun perusteella. Erilaisia tekstinparsintatyökaluja tullaan vertailemaan niiden nopeudessa, rakenteessa, virheiden määrässä sekä millaisessa muodossa ne parsivat taulukoissa olevan tekstin.

Paketeilla tullaan parsimaan otosjoukko PDF-tiedostoja ja mitataan rakennetta ja virheiden määrää. Rakenne kuvaa minkälaisessa kunnossa teksti parsitaan PDF-tiedostosta. Huono rakenne tarkoittaa, että teksti tulee hyvin erilaisessa muodossa alkuperäiseen verrattuna. Esimerkiksi tekstistä voi puuttua kappalejaot, joka tekee siitä hankalasti luettavaa. Virheiden määrää mitataan sillä, kuinka paljon tekstiin syntyy tekstivirheitä kuten lisäkirjaimia tai merkkejä, joka hankaloittaa tekstin lukemista. Nopeutta arvioidaan mittaamalla aika, kuinka

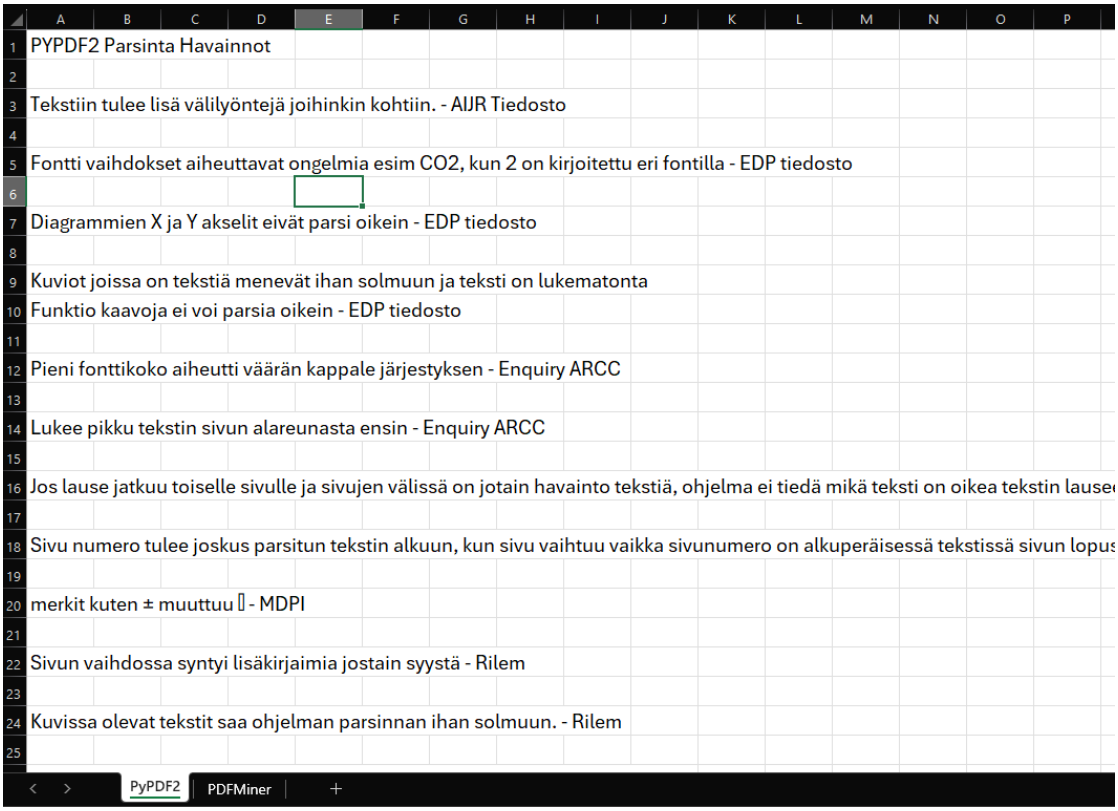
nopeasti työkalu käy läpi 100 PDF-tiedostoa, nämä 100 tiedostoa luodaan kopioimalla otosjoukko työkaluille. Lopullisella työkalulla voidaan mahdollisesti tulla käymään läpi tuhansia PDF-tiedostoja kerrallaan, mikä tekee nopeudesta hyvin tärkeän kriteerin.

8 TOTEUTUS

Lähestyminen tekstin parsintaprosessiin oli, että sen voi suorittaa kokonaan Pythonissa ja parsittu teksti on tämän jälkeen helposti poimittavissa, jotta sitä voidaan jatkokäsitellä. Käytin Jupyter Notebookia, jossa tein ipynb-tiedostomuodon ja käsittelin dataa pythonin avulla. Asensin PyPDF2- ja PDFMiner python kirjastot, joita tarvitsen PDF-tiedostojen parsintaan. Tämän lisäksi käytetään OS (operating system) moduulia, jolla annetaan Pythonille mahdollisuus käyttää järjestelmän ominaisuuksia, kuten tiedostojen luontia ja lukemista.

Alkuperäiset PDF-tiedostot laitetaan kansioon, josta ne voidaan lukea Pythonissa. Pythonissa tämä kansio laitetaan kohteeksi, josta alkuperäiset tiedostot ladataan. Tämän jälkeen osoitetaan toinen kansio, johon tehdään uudet tekstitiedostot parsitulle tekstille. OS-moduulia käyttäen pystymme käyttämään Pythonia luomaan tiedostoja, ja tähän lisäämme utf-8 koodauksen, jotta kaikki teksti tulee järjestelmän ymmärtämällä kielellä. Tämä prosessi tehdään molemmilla työkaluilla erikseen, minkä jälkeen meillä on 2 kansiota joissa toisessa on PyPDF tulokset ja toisessa PDFMiner tulokset.

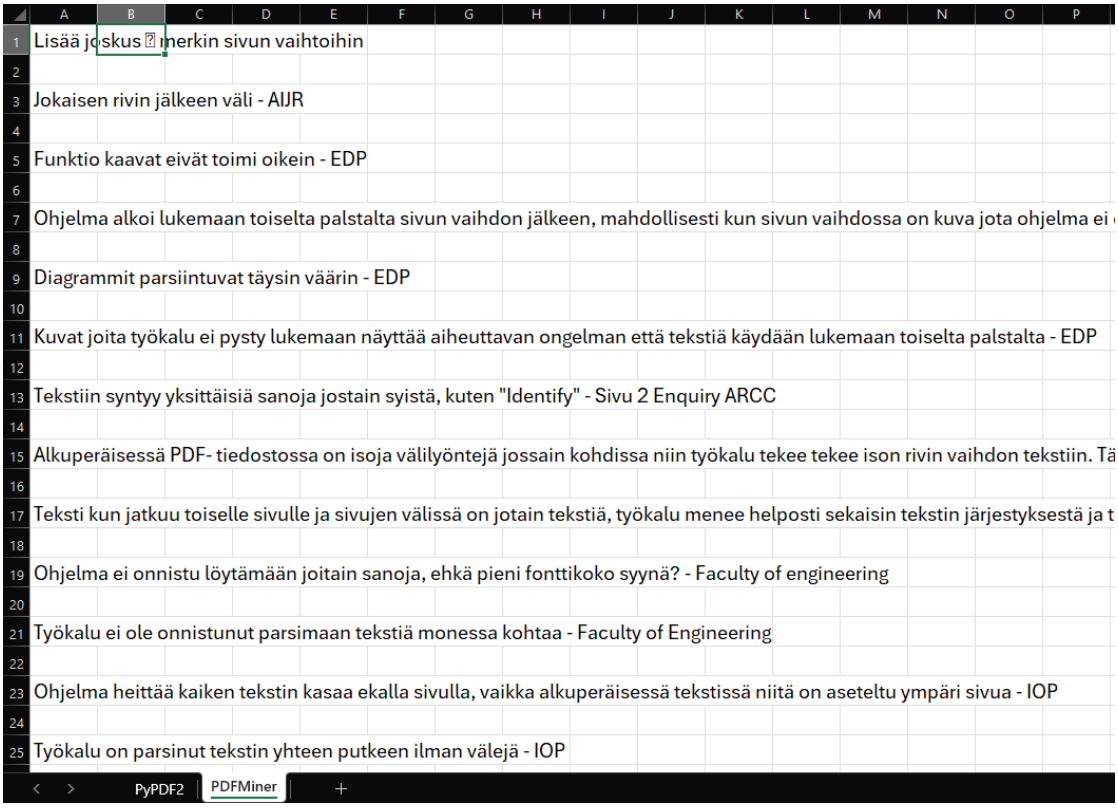
Tämän jälkeen kävin kaikki tulokset läpi ja analysoin, kuinka paljon tiedostot poikkesivat alkuperäisestä. Kirjoitin erillisen tiedoston, jossa oli havaintoja kaikista käsitellyistä PDF-tiedostoista. Kuvat 3 ja 4 tuovat näitä havaintoja esille. Molempien työkalujen havainnot oli lajiteltu erikseen ja tiedosto, johon havainnossa viitataan, on lauseen lopussa. Tämän jälkeen työkaluja verrataan keskenään ja katsotaan, kumpi työkalu tekee vähemmän virheitä ja kumman parsima teksti pysyy lähempänä alkuperäistä tekstiä. Kohdista, joissa molemmat työkalut tekevät virheitä, voidaan tehdä johtopäätöksiä, että tässä piirteessä parsintatyökalut eivät ole sopivia tekstin poimintaan.



	A
1	PYPDF2 Parsinta Havainnot
2	
3	Tekstiin tulee lisä välilyöntejä joihinkin kohtiin. - AIJR Tiedosto
4	
5	Fontti vaihdokset aiheuttavat ongelmia esim CO2, kun 2 on kirjoitettu eri fontilla - EDP tiedosto
6	
7	Diagrammien X ja Y akselit eivät parsi oikein - EDP tiedosto
8	
9	Kuviot joissa on tekstiä menevät ihan solmuun ja teksti on lukematonta
10	Funktio kaavoja ei voi parsia oikein - EDP tiedosto
11	
12	Pieni fonttikoko aiheutti väärän kappale järjestyksen - Enquiry ARCC
13	
14	Lukee pikku tekstin sivun alareunasta ensin - Enquiry ARCC
15	
16	Jos lause jatkuu toiselle sivulle ja sivujen välissä on jotain havainto tekstiä, ohjelma ei tiedä mikä teksti on oikea tekstin lause
17	
18	Sivu numero tulee joskus parsitun tekstin alkuun, kun sivu vaihtuu vaikka sivunumero on alkuperäisessä tekstissä sivun lopus
19	
20	merkit kuten ± muuttuu □ - MDPI
21	
22	Sivun vaihdossa syntyi lisäkirjaimia jostain syystä - Rilem
23	
24	Kuvissa olevat tekstit saa ohjelman parsinnan ihan solmuun. - Rilem
25	

PyPDF2 | PDFMiner

Kuva 3. PyPDF2-havainnot



	A
1	Lisää joskus □ merkin sivun vaihtoihin
2	
3	Jokaisen rivin jälkeen väli - AIJR
4	
5	Funktio kaavat eivät toimi oikein - EDP
6	
7	Ohjelma alkoi lukemaan toiselta palstalta sivun vaihdon jälkeen, mahdollisesti kun sivun vaihdossa on kuva jota ohjelma ei
8	
9	Diagrammit parsiintuvat täysin väärin - EDP
10	
11	Kuvat joita työkalu ei pysty lukemaan näyttää aiheuttavan ongelman että tekstiä käydään lukemaan toiselta palstalta - EDP
12	
13	Tekstiin syntyy yksittäisiä sanoja jostain syistä, kuten "Identify" - Sivu 2 Enquiry ARCC
14	
15	Alkuperäisessä PDF- tiedostossa on isoja välilyöntejä jossain kohdissa niin työkalu tekee tekee ison rivin vaihdon tekstiin. Tä
16	
17	Teksti kun jatkuu toiselle sivulle ja sivujen välissä on jotain tekstiä, työkalu menee helposti sekaisin tekstin järjestyksestä ja t
18	
19	Ohjelma ei onnistu löytämään joitain sanoja, ehkä pieni fonttikoko syynä? - Faculty of engineering
20	
21	Työkalu ei ole onnistunut parsimaan tekstiä monessa kohtaa - Faculty of Engineering
22	
23	Ohjelma heittää kaiken tekstin kasaa ekalla sivulla, vaikka alkuperäisessä tekstissä niitä on aseteltu ympäri sivua - IOP
24	
25	Työkalu on parsinut tekstin yhteen putkeen ilman välejä - IOP

PyPDF2 | PDFMiner

Kuva 4 PDFMiner-havainnot

Yhteenvetona havainnosta voimme tarkastella miten työkalut eroavat toisistaan. Minkälaisia virheitä ne tekevät toisiinsa verrattuna ja missä molemmat tekivät virheitä. PyPDF teki rakennellisia virheitä yhdistämällä tekstiä ja parsimalla tekstin väärästä paikasta. PDFMiner taas piti rakenteen lähempänä alkuperäistä, mutta teki virheitä syöttämällä uusia sanoja keskelle tekstiä. Molemmat työkalut taas kokivat ongelmia diagrammien akselien ja pienen fonttikoon kanssa.

8.1 Vertailuanalyysin työkalun löytämiseen

Tässä tullaan kasaamaan molemmista työkalusta, jotka tähän opinnäytteeseen on valittu, hyviä ja huonoja puolia ja koitetaan etsiä mikä työkaluista sopisi parhaiten. Toimeksiantaja valitsi nämä työkalut, sillä ne olivat nopea ottaa käyttöön ja niiden käyttöönotosta on paljon resursseja internetissä (Toots 2024). Työkaluilla tulee olemaan piirteitä, jotka tekevät siitä paremman joidenkin kriteereiden kohdalla ja huonomman toisien kriteereiden valossa. Vertailuanalyysin avulla koitetaan tuoda esille, mikä näistä vaihtoehdoista olisi paras tähän opinnäytteeseen.

Seuraavat laatukriteerit tulevat olemaan vertailuanalyysin perusteet, joiden avulla työkaluja verrataan keskenään. Tekstin rakenne, nopeus, virheiden määrä ja taulukkorakenne ovat kaikki, mitkä tulevat ratkaisemaan, mikä on sopivin työkalu. Taulukko 1 kuvaa vertailuanalyysia, jossa arvostellaan laatukriteerit arvosanoilla 1–3. 1 on huonoin arvosana ja 3 hyvä. Arvosanat korostetaan värein, joissa punainen tarkoittaa arvosanaa 1, keltainen arvosanaa 2 ja vihreä arvosanaa 3.

Kumpikaan työkalu ei saavuttanut arvosanaa 3. Tämä arvosana määräytyy, kuinka työkalu pystyy täyttämään kriteerin. Tällä menetelmällä jokaiseen kriteeriin löydetään työkalu, joka sopeutuu parhaiten kyseissä piirteessä. Lopulta saadaan selville työkalu, joka suoriutuu kaikista tehtävistä parhaiten.

Taulukko 1. Vertailuanalyysi kaikista työkaluista

	Nopeus	Rakenne	Taulukot	Virhe määrä
PDFMiner	1	2	2	2
PyPDF 2	2	1	2	2

Taulukosta huomaamme, että molemmat suoriutuvat keskivertoisesti monessa kriteerissä. Nopeudessa PyPDF2 on parempi, mutta PDFMiner pitää rakenteen paremmin alkuperäiseen. Taulukoissa ja virheiden määrässä molemmat suoriutuvat saman tasoisesti. Havainnoista kuitenkin huomaamme, että työkalut tekevät välillä erilaisia virheitä, joten tämä arvosana voi vaihdella tapauskohtaisesti.

8.2 Tulokset

Opinnäytteen lopputuloksena on saada selville, mikä työkaluista on paras vaihtoehto tekstin parsintaan. Parsittua tekstiä vertailtiin alkuperäiseen tekstiin ja poimittiin huomioita, missä kumpikin työkalu onnistuu ja epäonnistuu. Toimeksiantajan kanssa käytiin läpi havaintoja ja saatiin vahvistusta laadusta.

Rakennetta arvioidessa kiinnitin huomiota, kuinka hyvin teksti onnistui säilyttämään alkuperäisin tiedoston rakenteen. PyPDF2 onnistui hyvin tekstin poiminnassa, kun alkuperäisin tekstin rakenne oli normaali yhden palstan teksti. Kuviot, jotka sisälsivät tekstiä, olivat suuri ongelma ja aiheuttivat sen, ettei työkalu tiennyt, miten teksti pitäisi poimia. Myös kuvissa oleva teksti aiheutti samalla tavalla ongelmia, sillä ohjelma yrittää lukea tekstiä kuvan sisältä, mutta kuvien tausta tekee tekstin havaitsemisesta hankalaa. Funktiokaavat ja erilaiset merkit olivat kompastuskivi. Esimerkiksi ±-merkki muuttui symboliksi. Parsinnassa tuli isoja virheitä, kun alkuperäisen tekstin fonttikoko oli erikoisen pieni, mikä johti erikoiseen teksti- ja kappalejärjestykseen. Lauseen jatkuminen toiselle sivulle, kun sivun ylä- tai alareunan sisälle, tekee lauserakenteesta epäselvän.

PDFMinerin rakennepuolella toistui jotkin samat ongelmat, mutta joissakin piirteissä se onnistui paremmin. Yhden palstan tekstit onnistuivat hyvin, kuten PyPDF2:llakin. Funktiokaavat ja kuvissa olevat tekstit aiheuttivat samanlaisia ongelmia. Rakenteessa oli kuitenkin huomattavasti enemmän virheitä PyPDF-työkaluun verrattuna: Teksti oli väärässä järjestyksessä, sanoja puuttui lauseista tai tekstiin lisättiin välilyöntejä yms.

Virheiden määrää arvostelin, jos tekstiin tuli lisää sanoja tai ylimääräisiä merkkejä kohtiin, joita ei ole alkuperäisessä. PyPDF2 toimi tässä puolessa paremmin. Työkalu lisäsi tekstiin välilyöntejä ilman syytä, mutta teksti oli helposti luettavissa. Virheitä syntyi myös erilaisten merkkien kanssa, jotka eivät kääntyneet oikein, kuten edellä mainittu ± -merkki. PDFMiner ei onnistunut tässä kovin hyvin. Tekstiin laitettiin nuoliylös symboli kuvaamaan sivunvaihtoa, tämä voi johtaa hankalasti ymmärrettävään tekstiin, jos teksti esimerkiksi jatkuu toiselle sivulle. Virheitä tapahtui myös, jos tekstikoko tai riviväli on pieni, mikä johti PDFMinerin lukemaan tekstiä vääriltä riveiltä. Se johtaa epäselviin lauserakenteisiin.

Taulukkojakriteeriä arvostellessa kiinnitin huomiota, kuinka tarkkoja parsitut taulukot ovat alkuperäiseen verrattuna ja kuinka hyvin ChatGPT pystyy rekonstrukturoimaan taulukon tästä tekstistä. Yleisesti ottaen tulokset taulukkojen kanssa olivat hyvin samanlaisia. Taulukkojen tarvitsi olla helposti luettavassa muodossa, jotta se onnistuttiin rekonstruoimaan täysin oikein. Taulukot, joissa oli useampi ylätunniste eivät rakentuneet oikein. Data taulukoissa vastaa alkuperäistä, mutta järjestely ei ole sama.

Lopuksi nopeutta arvostelin kopioimalla otosjoukkoa 10 kertaa, jotta sain 100 PDF-tiedostoa. Lisäsin Pythonissa time-moduulin, jonka avulla pystytään mittaamaan, kuinka kauan koodien ajaminen kestää. PyPDF2 onnistui paljon paremmin tässä puolessa. 100 tiedoston ajaminen kesti noin 6 minuuttia, mutta PDFMinerilla samaan prosessiin meni noin 15 minuuttia. Nopeuteen vaikuttaa suuresti tietokoneen teho, sekä minkälainen aineisto on käytössä. Tuloksista saamme kuitenkin selville, että PDFMiner on huomattavasti hitaampi käymään tiedostoja läpi.

Johtopäätöksenä voimme kerätä lopullisen arvion molemmista työkaluista. PyPDF2 toimii paremmin nopeaan PDF- tiedostojen parsimiseen, mutta rakenne ei ole niin hyvä kuin PDFMinerillä. PDFMiner tekee enemmän virheitä, joka voi johtaa hankalaan tekstin lukemiseen. Taulukot toimivat molemmissa samalla tasolla. Taulukoiden kanssa tulee ongelmia enemmän, kun ChatGPT koittaa luoda taulukon uudestaan.

9 JOHTOPÄÄTÖKSET

Tämän työn tarkoituksena on ollut löytää hyvä PDF-tiedostojen parsintatyökalu Xamkin ReseptiRobotti-hanketta varten, jossa käydään ympäristönparantamiseen liittyviä PDF-tiedostoja. Työn alkupuolella esiteltiin mitä PDF-tiedostot ovat ja tekstin parsinnan taustaa, jotta lukijalla on parempi käsitys, mitä opinnäytteessä tutkitaan. Toteutuksessa haastateltiin Xamkin DataLABin työntekijää, joka on myös TKI-asiantuntija reseptirobotti hankkeessa. Tässä haastattelussa saatiin selville, mitä työkaluja työssä tullaan käyttämään ja miksi näihin työkaluihin päädyttiin. Lisäksi kysyttiin, mitä puolia työkalujen käytössä arvioidaan. Tästä päädyttiin arvioimaan neljää kriteeriä, joiden avulla parhaan mahdollisen työkalun olisi helpompaa.

Työkalujen ja kriteerien ollessa selvillä päätettiin luoda vertailuanalyysi, jossa työkaluja verrataan keskenään kriteerien perusteella. Havaintoja kerättiin molempien työkalujen käytöstä ja tuloksista. Vertailuanalyysiä tehdessä huomattiin nopeasti, että täydellistä vaihtoehtoa ei tule olemaan työkalujen valinnassa. Molemmat työkalut tekivät virheitä, usein jopa samoissa kohdissa. Täydellisyyden löytäminen oli tässä vaiheessa mahdotonta. Päädyttiin keskittymään, kumpi vaihtoehdoista toimii paremmin yleisellä tasolla ja onko jonkinlaisia PDF-muotoja, joita kannattaa välttää molemmilla työkaluilla. Pieni fonttikoko oli ongelma, jossa molemmat työkalut tekivät virheitä. Tämä aiheutti tekstin sekaisin menemistä kesken lauseen, sillä parsintatyökalu ottaa sanan alemmalta riviltä.

Tulosten perusteella voidaan tehdä johtopäätös, että parempi työkalu vertailtavista vaihtoehdoista on PDFMiner, jos nopeus ei ole tärkeää parsinnassa. PDFMiner tarjoaa paljon enemmän vaihtoehtoja PDF-parsintaan ja säilyttää alkuperäisen rakenteen paremmin PyPDF2 työkaluun verrattuna. Mutta pitää

myös muistaa, että PDFMiner teki myös paljon virheitä. Tätä työkalua käyttäessä pitää kiinnittää huomioita virheisiin, jotka tuotiin esille aikaisemmin.

Heikkilä (2014) kuvaa validiutta eli tutkimuksen pätevyyttä kirjassaan systemaattisen virheen puuttumisella. Hän tuo esille, kuinka mittarien arvot pitää olla oikeat. Mitattavat käsitteet pitää olla määritelty tai koko tutkimuksen tulokset eivät ole valideja. Tutkimuksen validiutta voidaan vahvistaa erilaisilla toimenpiteillä, esimerkiksi kyselyssä kysymysten pitää olla selkeitä ja niiden pitää liittyä tutkimusongelmaan. Perusjoukon tarkka määrittely on myös hyvä keino luoda tutkimukselle validiutta. Luotettavuutta eli reliabiliteettia Heikkilä (2014) kuvaa olevan tuloksien tarkkuutta. Tulokset pitää olla tarkasti kerätty ja perusteltu. Satunnaisia tuloksia ei saa olla. Aineisto on myös tärkeä osa tutkimuksen luotettavuutta. Pieni otosjoukko johtaa satunnaisiin tuloksiin ja tutkimuksen huonoon luotettavuuteen. (Heikkilä 2014, 27–28.)

Tässä tutkimuksessa pätevyyttä ja luotettavuutta tuotiin kiinnittämällä huomiota aineistoon ja vertailuanalyysissä tuodut kriteerit määriteltiin tarkasti jo tutkimuksen alussa. Aineisto saatiin toimeksiantajalta, ja siitä valittiin otosjoukoksi erilaisia PDF-tiedostoja eri julkaisijoilta, satunnaisten tuloksien välttämiseksi. Vertailuanalyysissä käytetyt kriteerit tuovat luotettavuutta, sillä ne antavat selkeät tiedot, miten työkaluja arvioidaan.

10 TUTKIMUKSEN KULKU JA KEHITTÄMINEN

Tutkimusta lähdettiin suorittamaan toimeksiantajan pyynnöstä etsiä hyvää vaihtoehtoa heidän hankkeeseensa, jossa tutkittiin PDF-tiedostaja. Ongelmaa lähdettiin ratkomaan vertailuanalyysin avulla, sillä mahdollisia vaihtoehtoja PDF-parsintaan on paljon. Näin päädyttiin tekemään vertailua työkaluista, jotta saataisiin selville mitä vahvuuksia ja heikkouksia kaikilla on.

Opinnäytteessä oli alkuvaiheilla tarkoitus vertailla useampia työkaluja, mutta lopulta päädyttiin vain edellä mainittuihin kahteen, aikarajoitusten takia. Valitut työkalut olivat toimeksiantajan mielestä tärkeimmät, joita kannattaa tutkia ja vertailla. Tutkimuksessa keskityttiin vain näihin kahteen. Tällä kertaa tutkimuksessa keskityttiin vain englanninkielisiin PDF-tiedostoihin, joten jatkotutkimuksena voisi olla muun kielisten PDF-tiedostojen parsimisen vertailu.

Opinnäytetyön tuloksia on näytetty hankkeen TKI-asiantuntijalle, joka piti tuloksia hyvinä ja piti virheiden havaitsemista tärkeänä osana tutkimusta. Tuloksia ei vielä opinnäytteen julkaisuvaiheessa ole vielä hyödynnetty hankkeessa, mutta toimeksiantaja on tuonut tämän tutkimuksen esille hankkeenjärjestäjien kanssa ja kokee, että tutkimus auttaa hankkeen edistämisessä.

LÄHTEET

Karelia AMK= Karelian ammattikorkeakoulu. 2024. Opinnäytetyön eri muodot. WWW-dokumentti. Päivitetty 30.5.2024. Saatavissa: https://libguides.karelia.fi/c.php?g=679019&p=4901221 [Viitattu 15.8.2024].

Chowdhury, G. 2003. Natural Language Processing. E-Kirja. University of Strathclyde. Saatavilla: https://pure.strath.ac.uk/ws/portalfiles/portal/131112/strathprints002611.pdf [Viitattu 13.8.2024].

Donges, N. 2023. Introduction to Natural Language Processing (NLP). WWW-sivu. Saatavissa: https://builtin.com/data-science/introduction-nlp [Viitattu 2.11.2024].

General Python FAQ s.a. Python Software foundation. WWW-dokumentti. Päivitetty 2.9.2024. Saatavissa: https://docs.python.org/3/faq/general.html#why-was-python-created-in-the-first-place [Viitattu 1.11.2024].

Heikkilä, T. 2014. Tilastollinen tutkimus. 9. painos. Helsinki: Edita. E-kirja. Saatavilla: https://kaakkuri.finna.fi/Record/kaakkuri.223632?sid=4879440060 [Viitattu 27.10.2024].

Hirsjärvi. S, Hurme. H. 2022. Tutkimushaastattelu: Teemahaastattelun teoria ja käytäntö. 2. painos. Gaudeamus oy. E-Kirja. Saatavilla: https://kaakkuri.finna.fi/Record/kaakkuri.229077?sid=4879440870 [Viitattu 20.10.2024].

Jane. 2023. PDF data extraction and OCR: The ultimate guide. Parsio. WWW-dokumentti. Saatavissa: https://parsio.io/blog/pdf-parser/ [Viitattu 24.11.2024].

Juhila, K. 2021. Laadullinen tutkimus ja teoria. Teoksessa Jaana Vuori (toim.) *Laadullisen tutkimuksen verkkokäsikirja*. Tampere: Yhteiskuntatieteellinen tietoarkisto. Saatavilla: https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvali/mita-on-laadullinen-tutkimus/laadullinen-tutkimus-ja-teoria/ [Viitattu 12.8.2024].

Kock, M. 2023. About Lens.org. WWW-sivu. 2024. Saatavissa: https://about.lens.org/ [Viitattu 25.5.2024].

Laadullinen tutkimus. Jyväskylän yliopisto. WWW-dokumentti. Saatavissa: https://sites.app.jyu.fi/mehu/fi/menetelmapolku/tutkimusstrategiat/laadullinen-tutkimus [Viitattu 10.11.2024].

Liddy, E. 2001. Natural Language processing. E-kirja. Syracuse University. Saatavissa: https://surface.syr.edu/cgi/viewcontent.cgi?article=1043&context=istpub [Viitattu 13.8.2024].

Määrällinen tutkimus. Jyväskylän yliopisto. WWW-sivu. Saatavissa: https://sites.app.jyu.fi/mehu/fi/menetelmapolku/tutkimusstrategiat/maarallinen-tutkimus [Viitattu 10.11.2024].

Open AI. 2022. Introducing ChatGPT. WWW-dokumentti. Päivitetty 30.10.2022. Saatavissa: https://openai.com/index/chatgpt/ [Viitattu 28.8.2024].

Otos ja Otantamenetelmät. WWW-sivu. Saatavissa: https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvanti/otos/otantamenetelmat/ [Viitattu 25.8.2024].

Otten, N,V. 2023. WWW-dokumentti. Syntactic Analysis: A Power Tool In NLP Made Easy With Examples, Illustrations & Tutorials Saatavissa: https://spotintelligence.com/2023/10/28/syntactic-analysis-nlp/ [Viitattu 5.11.2024].

Project Jupyter. WWW-Sivu. Project Jupyter Documentation. Saatavissa: https://docs.jupyter.org/en/latest/ [Viitattu 4.8.2024].

Rossum, G. 2009. A Brief Timeline of Python. Blogi. The history of python. 20.1.2009. Saatavissa: https://python-history.blogspot.com/2009/01/brief-timeline-of-python.html [Viitattu 22.8.2024].

Sachideva, S. 2024. PyPDF2 Library for working with PDF files in Python. Blogi. 19.2.2024. Saatavissa: https://www.analyticsvidhya.com/blog/2021/09/pypdf2-library-for-working-with-pdf-files-in-python/ [Viitattu 8.10.2024].

Scarlet, R. 2023. Why python keeps growing, explained. Blogi. 7.3.2023. Saatavilla:https://github.blog/2023-03-02-why-python-keeps-growing-explained/ [Viitattu 15.4.2024].

Shinyama, Y. 2013. PDFMiner. WWW-dokumentti. github. 28.03.2014. Saatavissa:https://euske.github.io/pdfminer/ Päivitetty 24.3.2014. [Viitattu 12.7.2024].

Text parsing s.a. reintechmedia. WWW-dokumentti. Saatavissa:https://reintech.io/terms/category/text-parsing-overview [Viitattu 11.8.2024].

Trinh, P. 2022. 5 Key Advantages & Disadvantages of PDF Files. Mailmergic. Artikkeli. 22.3.2022. Saatavissa: https://mailmergic.com/blog/5-key-advantages-disadvantages-of-pdf-files/. [Viitattu 12.8.2024].

Tuominen, K., Niva, M. & Malmberg, L. 2011. Benchmarking käytännössä. Turku: Turku Benchmarking. Saatavilla: https://kaakkuri.finna.fi/Record/nelli29_mamk.1000000000398443?sid=4879993008 [Viitattu 2.9.2024].

Toots, V. 2024. TKI-asiantuntija. Haastattel. 7.10.2024.Xamk.

Wulian 2024. Status of Python versions. WWW-dokumentti. Päivitetty 31.8.2024. Saatavissa: https://devguide.python.org/versions/ [Viitattu 1.11.2024].

Wilson, G. 2024. Software Design by Example A Tool-Based Introduction with Python. Chapman & Hall. E-Kirja. Päivitetty 5.4.2024. Saatavissa: https://thirdbit.com/sdxpy/intro/ . [Viitattu 4.11.2024].

Xamk. 2024. Reseptirobotti- Tiedosta tehoa tuotekehitykseen. WWW-sivu. Saatavissa: https://www.xamk.fi/hanke/reseptirobotti-tiedosta-tehoa-tuotekehitykseen/ [Viitattu 4.10.2024].