Tampereen yliopisto

Jari Rauhala

HAKUROBOTTIEN SUUNNITTELUPERIAATTEIDEN SOVELTAMINEN SCRAPYLLA

Kandidaatintyö
Informaatioteknologian ja viestinnän tiedekunta
Kesäkuu 2020

TIIVISTELMÄ

Jari Rauhala: Hakurobottien suunnitteluperiaatteiden soveltaminen Scrapylla
Kandidaatintyö
Tampereen yliopisto
Tieto- ja sähkötekniikka, TkK
Kesäkuu 2020

Hakurobotti on yksi internetin valtavan tietomäärän hallintaan kehitetyistä työkaluista. Se on ohjelma, joka lähettää palvelimille pyyntöjä ja jäsentää vastauksena saamansa datan. Poimimalla datasta linkkejä hakurobotti voi kulkea sivulta toiselle ja saada näin kerättyä suuriakin määriä tietoa. Yksi keskeisimmistä käyttökohteista hakuroboteille ovat hakukoneet, jotka käyttävät niitä tietovarastojensa rakentamiseen. Hakurobotteja on käytetty myös muussa kaupallisessa toiminnassa sekä tieteellisen tutkimuksen apuvälineinä. Vaikka eri hakurobottien käyttökonteksti yksityiskohdiltaan ainutkertainen ja siten suunnitteluratkaisut ovat erilaisia, on niissä paljon yhtäläisyyksiä. Tässä työssä selvitetään, mitkä ovat keskeisimmät hakurobottien suunnittelua koskevat kysymykset ja kuinka käyttökonteksti vaikuttaa niiden soveltamiseen.

Tärkeimmät suunnittelukysymykset koskevat sitä, millä tavoin ja kuinka usein hakurobotti liikkuu sivulta toiselle, samalla huomioon ottaen myös sivustojen palveluntarjoajan tarpeet. Erityisesti tapauksissa, kuten hakukoneiden tekemissä hauissa, ladattavan tiedon määrä voi kasvaa hyvin suureksi, jolloin myös resurssien kulutus on merkittävää. Strategisen suunnittelun tavoitteena onkin optimoida hakurobotin toimintaa mahdollisimman resurssitehokkaaksi, samalla varmistaen laadukkaiden tulosten saamisen pitkällä aikavälillä. Strategisen suunnittelun lisäksi sekä hakurobotin suunnittelijan että sivustojen ylläpitäjän tulee ottaa huomioon eettisyyteen sekä turvallisuuteen liittyviä kysymyksiä. Hakurobotit ovat tehokkaita haalimaan tietoa – joskus sellaistakin, jota tiedon haltija ei tahtoisi luovuttaa. Molempien osapuolten etujen varmistamiseksi on kehitetty Robotin rajausstandardi, johon ylläpitäjä voi määrittää sivustoaan koskevia ehtoja.

Työn konstruktiivisessa osassa suunnittelukysymysten pohjalta määritetään turvallisen ja eettisen toiminnan mahdollistava konteksti ja toteutetaan siinä toimiva hakurobotti. Ohjelma kirjoitetaan avoimen lähdekoodin viitekehyksellä Scrapylla ja se hakee ja käsittelee Yleisradion uutissisältöä. Sillä suoritetaan eri algoritmeja hyödyntäen kaksi eri hakua, joiden tuloksista voidaan päätellä, että syvyyteen ensin -algoritmilla päästään käsiksi vanhempaan uutissisältöön nopeammin, kuin leveyteen ensin -algoritmilla. Suoritetut haut antavat viitteitä sovellettujen suunnitteluratkaisuiden toimivuudesta, vaikkei yleispäteviä johtopäätöksiä voidakaan vetää. Kuitenkin, koska internet määrittää kaikille sen käyttäjille tietyiltä osin rajatun kontekstin, ovat tämän työn suunnitteluratkaisut ja tulokset siten muihin tapauksiin vertailukelpoisia ja oikeellisia.

Avainsanat: Hakurobotti, Scrapy, internet, hakukone, asiakas-palvelinmalli

SISÄLLYSLUETTELO

1. JOHDANTO

Tieteelliset saavutukset ovat muokanneet ihmisten maailmankuvaa kautta historian, ja ne ovat saaneet aikaan vallankumouksellisia kehitysaskeleita yhteiskunnassa. Charles Darwinin evoluutioteoria muokkasi käsitystämme luonnosta ja Sigmund Freud avasi käsityksen ihmismielestä ja tiedostamattomasta. Vastaava, maailmaa mullistava vallankumous on käynnissä: informaation vallankumous. [14]

Perusta internetille syntyi 1960-luvulla Yhdysvalloissa kylmän sodan vauhdittamana. Teknologisen kehityksen kilpajuoksu Neuvostoliittoa vastaan loi tarpeen tiedon tehokkaammalle jakamiselle ja saatavuudelle. Pian se kuitenkin levisi myös siviilien käyttöön. [10] Viime vuosikymmenten aikainen kehitys on ollut räjähdysmäistä. Yhä useammalla on mahdollisuus olla yhteydessä toisiinsa internetin välityksellä ja päästä käsiksi tietoon – kuten uutisiin ja tieteellisiin tutkimuksiin. Tietoa liikkuu valtavia määriä ja kasvun on arvioitu yhä kiihtyvän tulevina vuosina [9].

Jo 1990-luvun alkupuolella internet oli kasvanut niin suureksi, että tiedon ja sivustojen hallintaan kaivattiin apuvälineitä. Tarpeeseen vastasivat hakukoneet, kuten vuonna 1998 perustettu Google. [34] Sen toiminnan ytimessä on hakurobotti, Googlebot, joka kerää ja tallentaa automaattisesti tietoa internet-sivuista ja niiden välisistä suhteista [38]. Sen tavoitteena on kerätä tietoa mahdollisimman laajalta skaalalta relevantteja sivustoja, jotta hakukone pystyy vastaamaan mahdollisimman moneen hakutarpeeseen [8].

Hakuroboteille on myös muita käyttötarkoituksia, kuin mahdollisimman laajan tietokannan rakentaminen hakukonetta varten. Erilaisilla suunnitteluratkaisuilla hakuroboteilla voidaan hakea tarkemmin rajattua tietoa. Niitä onkin käytetty esimerkiksi sähköpostiosoitteiden keräämiseen. [8]

Tämän työn tarkoituksena on selvittää, mitkä ovat hakurobotin suunnittelun kannalta keskeisimmät kysymykset ja kuinka niitä voidaan soveltaa tietyssä kontekstissa. Hakurobotteihin ja niiden käyttöä ja suunnittelua koskeviin kysymyksiin ja käytäntöihin tutustutaan taustoittamalla olemassa olevaa kirjallisuutta. Kysymysten ja käytäntöjen soveltamisen tutkimisessa käytetään avoimen lähdekoodin hakurobottiviitekehystä Scrapya. Sen avulla toteutetaan hakurobotti, joka hakee uutissisältöä Yleisradion verkkosivuilta ja rakentaa siitä tietokannan. Hakua varten arvioidaan ja määritetään tilanteeseen parhaiten sopivat suunnitteluperiaatteet ja algoritmit.

Hakurobotteihin tutustutaan aluksi hakukoneiden kautta. Hakukoneet ovat monelle erittäin tuttuja työkaluja, mikä auttaa hahmottamaan hakurobottien merkitystä. Luvussa 2 käsitellään myös, millaisiin muihin käyttötarkoituksiin robotteja voidaan käyttää, sekä tutustutaan toimintaperiaatteisiin ja keskeisimpiin komponentteihin tarkemmin. Luvussa 3 selvitetään, miten eri suunnitteluperiaatteet määrittävät hakurobotin toimintaa. Strategiset kysymykset ohjaavat robotin toimintaa suoraan, kun taas turvallisuus ja eettisyys ovat seikkoja, jotka eivät aina vaikuta suunnitteluratkaisuihin suoraan, vaan liittyvät laajempaan kokonaisuuteen. Luvussa 4 paneudutaan Scrapyn toimintaan ja kutakin suunnitteluperiaatetta peilaten toteutetaan hakurobotti ja määritetään sille käyttökonteksti. Sillä tehdyn haun tuloksia arvioidaan luvussa 5. Lopuksi johtopäätöksissä arvioidaan kokonaisuutta, sovellettuja käytänteitä sekä saatujen tulosten merkityksellisyyttä.

2. HAKUROBOTIT OSANA INTERNETIÄ

2.1 Hakurobotit hakukoneiden toiminnassa

Jotta internet-selaimen käyttäjä voi avata tahtomansa sivun, on hänen lähetettävä HTTP-pyyntö oikeaan URL-osoitteeseen. Selain vastaanottaa palvelimelta saamansa tiedot ja muuntaa ne käyttäjälle esitettävään muotoon. Kokonaisuudessaan olemassa olevia sivustoja on lähes kaksi miljardia ja tiedon määrä kasvaa jatkuvasti [37], joten oikean tiedon löytäminen ilman apuvälineitä on usein mahdotonta.

Hakukone ylläpitää palvelimellaan tietokantaa, jonne on tallennettu suuri määrä internetistä löytyvää tietoa, mistä tieto on löydetty, sekä tietoa sivujen välisistä suhteista. Hakukone päivittää tietokantaa usein, jotta tietokanta vastaisi mahdollisimman ajantasaisesti sivuja, joilta tieto on tallennettu. Tietokanta on järjestetty niin, että sieltä voi hakea tietoa mahdollisimman helposti avainsanoja tai niiden yhdistelmiä käyttäen. Hakukone tekee käyttäjän haun perusteella haun tietokantaansa, ja esittää löytämänsä URL-osoitteet järjestettynä hakuun vastaavuuden perusteella. [6]

Hakukoneiden tietokannat rakennetaan hakuroboteilla indeksoinniksi kutsutussa prosessissa. Hakurobotit ovat ohjelmia, jotka etsivät ja lataavat dataa internetistä. Ne tutkivat itsenäisesti lataamaansa tietoa, poimivat sieltä muun muassa hyperlinkkejä ja siirtyvät niitä pitkin seuraaville sivustoille. Hakurobottien toimintaa räätälöimällä voidaan määrittää, millaista dataa ja millaisilta sivustoilta sillä halutaan tavoittaa. Vaikka hakukoneet joutuvat rajallisten resurssiensa takia myös ohjaamaan hakurobottejaan indeksoimaan esimerkiksi suosittuja sivustoja useammin kuin muita, on niiden pyrkimys lähtökohtaisesti saada aikaan mahdollisimman kattava otos internetistä. [8]

2.2 Muita käyttökohteita hakuroboteille

Hakurobotteja voidaan käyttää hakukoneiden lisäksi moniin muihin tarpeisiin. Hakurobottien toimintaa rajaamalla esimerkiksi tiettyyn aihepiiriin tai sivustoon voidaan saada kerättyä rajatumpaan käyttökohteeseen sopivaa dataa. Tiedonkeruusta ja datan hyödyntämisestä on tehty useita tutkimuksia. Myös monen, suurimpia hakukoneita huomattavasti pienempien, yrityksen liiketoiminta perustuu datan keräämiseen ja tiedon myymiseen.

Nisar ja Yeung (2018) [23] tutkivat, voiko Twitteristä kerättyjen twiittien perusteella ennustaa pörssikurssien liikettä. He keräsivät vuoden 2016 Iso-Britanniassa pidettyihin useisiin eri vaaleihin liittyviä twiittejä. Niiden perusteella he analysoivat yleistä poliittista

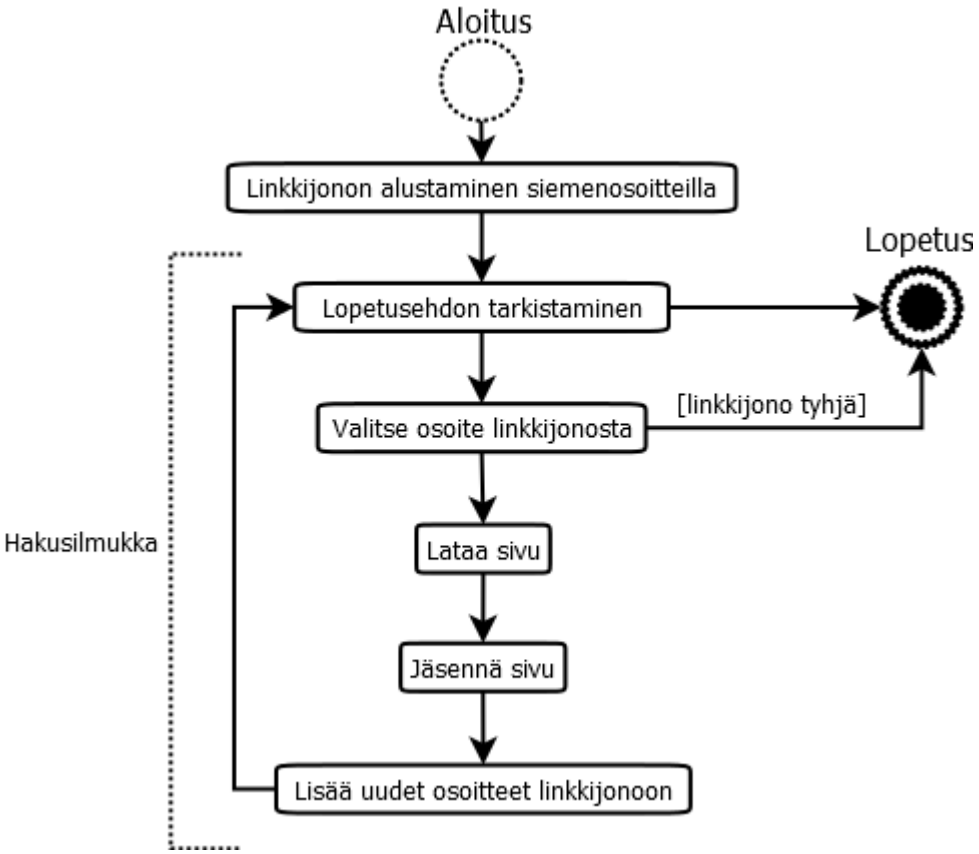
ilmapiiriä ja tutkivat, onko sillä yhteyttä Lontoon pörssin muutoksiin. He pitivät keräämäänsä datamäärää jokseenkin pienenä, mutta osoittivat, että twiittien perusteella on mahdollista tehdä päätelmiä yhteiskunnassa vallitsevista mielipiteistä ja siten ennustaa poliittisia ja taloudellisia kehityssuuntia. [23]

Nemeslaki ja Pocarovszky [22] tutkivat hakurobottien käyttöä yhteiskunta- ja taloustieteen tutkimuksessa verkkokaupankäynnin mallien kehityksessä. He loivat malleja, joilla kartoitettiin unkarilaisten verkkoviranomaisten toimintaa, luokittelivat unkarin suurimpia verkkosivujen tarjoajia sekä seurasivat lentoliikenteen hinnoittelun kehitystä. Tuloksilla oli huomattavaa arvoa eri teollisuudenalojen yrityksille, mutta he törmäsivät myös ongelmiin. Heidän hakurobottinsa käyttö estettiin eräällä tutkitulla sivustolla, kun sitä oli käytetty ilman lupaa ja luokiteltu siksi haittaohjelmaksi. [22].

Hir InfoTech on intialainen yritys, joka tarjoaa muille yrityksille palvelua, jossa suoritetaan hakurobotilla räätälöity haku asiakkaan tarpeisiin. He voivat esimerkiksi analysoida kilpailevien verkkokauppojen tarjontaa [1]. Georanker myy niin ikään tarpeisiin räätälöityjä ratkaisuja, kuten seurantaa yritysasiakkaan kuvien väärinkäytölle tai potentiaalisten asiakkaiden yhteystietojen etsimiseen [12].

2.3 Hakurobotin toiminta

Hakurobotin toiminnan yksi keskeisimmistä komponenteista on linkkijono. Se on lista URL-osoitteita, joita ohjelma käsittelee ennalta määrätyssä järjestyksessä. Robotin toiminta alkaa määrittelemällä linkkijonoon siemenosoitteet, eli linkit sivustoille, joista haun tahdotaan aloitettavan. Kuvan 1 mukaisesti linkkijonon alustamisen jälkeen siirrytään hakusilmukkaan. [25]



Kuva 1. *Hakurobotin toimintaperiaate [25].*

Hakusilmukka on sarja toimintoja, joita toistetaan niin kauan, kun hakurobotti on toiminnassa. Sen päättymiselle voidaan määrittää lopetusehto, jonka täyttyessä ohjelma pysähtyy. Voidaan määrätä esimerkiksi tietty määrä vierailtavia sivuja tai rajallinen aika, jonka se toimintaan saa käyttää. Suoritus voi päättyä myös, jos robotti on käsitellyt kaikki linkkijonon osoitteet. [25]

Mikäli lopetusehto ei ole täyttynyt ja linkkijonossa on osoitteita, ottaa hakurobotti sieltä seuraavan käsittelyyn. Se lähettää osoitteeseen HTTP-pyynnön ja jää odottamaan vastausta. Sellaisen saadessaan se käsittelee lataamansa sivun ja poimii sieltä tarvitsemansa tiedot, jotka se voi tallentaa esimerkiksi paikalliseen tietokantaan. Löytäessään tarpeisiinsa sopivia hyperlinkkejä se lisää ne linkkijonoon. Löydetty linkki voi olla kuitenkin käyttökelvoton, jos se esimerkiksi vie sivustolle, jonne ei haluta mennä, tai jos robotti on sen jo haussaan aiemmin löytänyt. Robotin on siksi pidettävä kirjaa löytämistään osoitteista. Ladatun datan käsiteltyään se palaa hakusilmukan alkuun. [25]

3. HAKUROBOTTIEN SUUNNITTELUPERIAATTEET

3.1 Strateginen suunnittelu ja käytännöt

Sivustojen määrä internetissä on valtava [37] ja vapaasti liikkuvan hakurobotin linkkijonoon voi nopeasti kerääntyä kymmeniä tuhansia osoitteita [25]. Vaikka tietokoneiden laskentateho on kehittynyt erittäin tehokkaaksi ja ne suorittavat ohjelmia erittäin nopeasti, tiedonsiirto ei ole pysynyt kehityksen vauhdissa. Siksi hakurobottikaan ei pysty lataamaan annetussa ajassa kuin vain murto-osan koko internetin sivuista. Sen on siis priorisoitava mistä ja mitä tietoa se lataa. Hakurobotin toimintaa voidaan määrittää neljän eri toimintaperiaatteen kautta: valinnat, uudelleenvierailu, kohteliaisuus ja rinnasteisuus. [7]

3.1.1 Valinnat

Haun etenemiselle keskeistä on järjestys, jossa hakurobotti valitsee linkkijonosta uuden osoitteen tarkasteltavaksi. Valinnan vaikutus kertautuu, koska valitun osoitteen kautta saatavat uudet linkit määrittävät taas niistä löytyvät linkit. Leveyteen ensin -algoritmissa linkkijonosta valitaan käsittelyyn linkki FIFO-periaatteella, eli ensimmäisenä lisätty linkki otetaan ensimmäiseksi käsittelyyn. Tällöin hakurobotti tulee käsitelleeksi kaikki yhdeltä sivulta löydetyt ennen seuraavaan siirtymistä. Sen vastakohtana on syvyyteen ensin -algoritmi, jossa linkeistä valitaan viimeiseksi jonoon liitetty, eli kyseessä on LIFO-periaate. Kun valitaan aina uusin linkki, siirtyy robotti aina heti sivulta seuraavaan. [7] Hakukoneiden kontekstissa Najork ja Wiener [21] sekä Boldi et al. [5] pitivät leveyteen ensin -algoritmia parhaana. Cho et al. [8] saivat parhaat tulokset käyttämällä mittaria, joka arvioi sivuja niille johtavien linkkien määrän perusteella.

Hakurobotin toimintaan voidaan vaikuttaa paljon arvioimalla kerättyjä linkkejä ennen niiden avaamista. Toiminta tehostuu huomattavasti, koska tällöin säästetään resursseja, jotka kuluisivat sivun lataamiseen. Ensisijaista on karsia niistä pois ne, joilla on jo vierailtu tai jotka ovat jo jonossa. Myös tarkastelemalla linkkejä ja niiden ankkuritekstiä voidaan ennustaa, mitä aihepiiriä sivu käsittelee. [7] Näin voidaan tarkentaa haun etenemisen suuntaa karsimalla esimerkiksi tiettyjä aihepiirejä pois.

3.1.2 Uudelleenvierailu

Internetiin lisätään ja sieltä poistetaan tietoa päivittäin suuria määriä. Hakuroboteilla tehtävä haku saattaa kestää jopa viikkoja tai kuukausia, joten jo sen päätyttyä on osa

haun alkupään tiedosta todennäköisesti vanhentunutta. Hakukoneet pyrkivät tarjoamaan mahdollisimman ajantasaista tietoa käyttäjilleen, joten vanhentuneen tiedon tarjoamista voidaan mitata kustannuksena. Sitä voidaan arvioida kahtena eri funktiona: ikänä ja tuoreutena. Ikä kertoo, kuinka vanhentunut hakukoneen tallentama kopio on, tuoreus taas ei kerro muuta kuin onko kopio ajantasainen vai ei. [7]

Uudelleenvierailulle voidaan määrittää käytäntöjä, päivitetäänkö jokaista sivua yhtä usein, vai arvotetaanko eri sivuja korkeammalle kuin toisia [7]. Keskimääräistä tuoreutta mittaamalla paras käytäntö on tasa-arvoinen päivittäminen [8]. Hakukoneiden tapauksessa optimaalinen uudelleenvierailutoimintaperiaate on jossain yksilökohtaisen priorisoinnin ja tasa-arvon väliltä. Tasa-arvoisuus on lähtökohtaisesti hyödyllistä, mutta on myös sivustoja, joita on kannattavaa päivittää useammin kuin toisia. [7]

3.1.3 Kohteliaisuus

Palvelun ylläpitäjälle on toivottavaa, että kyseistä sivua käytetään, eikä yksittäisen HTTP-pyynnön lähettäminen ei kuluta palvelimelta huomattavasti resursseja, kuten aikaa, energiaa tai rahaa. Myös, koska on sekä ylläpitäjän että käyttäjän etu, että sivusto löytyy hakukoneen tuloksista, on hakukoneiden ja siten hakurobottien tuoma hyöty selvä. Kuitenkin, koska hakurobotit pystyvät lähettämään lukuisia pyyntöjä lyhyessä ajassa, ja koska hakurobotteja, niin kuin internetin käyttäjiäkin, on yhä enemmän ja enemmän, ovat palvelimet ja muut verkon toiminnallisuuksien tarjoajat vaarassa tulla ylikuormitetuiksi. Palvelimen kannalta myös hakurobotin tyyppi ja tarkoitus voi olla merkitsevää arvioitaessa, halutaanko robotin vierailevan sivulla vai ei. [36]

Yhtenä ratkaisuna on vuonna 1994 sovittu Robotin rajausstandardi. Se on robots.txt-niminen tiedosto, joka sijoitetaan palvelimen hakemiston juureen. Eettisesti toimiva hakurobotti pyytää tiedostoa luettavaksi ennen muuta sivuston tietoa. Sivuston ylläpitäjä voi määrittää tiedostoon, mitkä ja millaiset robotit sivustolle pääsevät. Protokolla on kuitenkin vai sopimus, jonka noudattaminen on pelkästään eettinen kysymys, eivätkä kaikki sitä siksi noudata. [36]

Yksi merkittävä tekijä palvelimen resurssienkulutuksen kannalta on, kuinka usein robotti lähettää sinne pyyntöjä. Tähän ei Robotin rajausstandardissa ole kuitenkaan mahdollista asettaa rajoitusta. Yhteydenottojen väliselle ajalle on annettu silti useita ehdotuksia, joiden kelpoisuutta on arvioitava tilannekohtaisesti. Lokitietoja tutkimalla on havaittu, että tunnettujen hakurobottien vierailuväli vaihtelee 20 sekunnista 3–4 minuuttiin. [7]

3.1.4 Rinnasteisuus

Kun tavoitteena on ladata suuri määrä sivuja lyhyessä ajassa, on rinnakkain ajettavien hakurobottien käyttö hyödyllistä ja usein myös välttämätöntä. Rinnakkain toimivien hakurobottien täytyy kuitenkin toimia koordinoidusti, jotta ne eivät hae turhaan tietoa samoilta sivuilta. Koordinointi kuluttaa kuitenkin resursseja, mikä on kokonaisuutta tarkastellessa otettava huomioon. [8]

Koordinointia voidaan tehdä joko dynaamisesti tai staattisesti. Dynaamisessa tavassa hakuroboteille yhteinen palvelin pitää kirjaa URL-osoitteista ja jakaa niitä roboteille tarpeen mukaan. Palvelin huolehtii, että vain yksi robotti käy annetussa osoitteessa. Sen sijaan staattisessa tavassa ennen robottien ajoa niille on määrätty sääntöjä, joiden mukaan osoitteiden jako tapahtuu. Jokainen robotti tietää toistensa vastuualueet ja osaavat jakaa tehtäviä niiden mukaisesti. [8]

3.2 Turvallisuus

Hakurobotteihin liittyy useita turvallisuuskysymyksiä niin niiden suunnittelijoiden, käyttäjien kuin sivustojen tarjoajienkin osalta. Kuten kaikki muutkin tietokoneohjelmat, hakurobotit ovat haavoittuvaisia – niissä voi olla virheitä ja niille voidaan tehdä ansoja. [20] Hakurobotit ovat myös tehokkaita löytämään sellaistakin tietoa, jota ihmisen olisi hankala löytää, mikä voi edesauttaa arkaluonteisen tiedon joutumista vääriin käsiin [4].

Haun suoritus voi keskeytyä, jos hakurobotti jää jumiin ikuiseen silmukkaan. Jumiutuminen voi tapahtua joko vahingossa tai sen voi aiheuttaa suunniteltu ansa. Robotin suunnittelijan tulee ottaa huomioon tilanteet, jotka voivat aiheuttaa silmukkaan jäämisen ja suunnitella keinoja tilanteiden torjumiseksi. Sivustot voivat määrittää ehtoja hakuroboteille Robotin rajausstandardiin, mutta kaikki eivät sitä välttämättä noudata. Roskapostin lähettämistä varten sähköpostiosoitteita keräävälle hakuroboteille on tehty tarkoituksellisia ansoja niiden pysäyttämiseksi. Jotkut sivustot ovat tehneet ansoja suosittujen hakukoneiden hakuroboteille manipuloidakseen asemaansa hakutuloksissa. [17]

Mahdollisuutta löytää tehokkaasti tietoa voidaan käyttää kuitenkin myös turvallisuuden edistämiseen. Moshchuk et al. [20] tutkivat vakoiluohjelmien uhkaa internetissä. He käyttivät hakurobottia, jolle annettiin kategorisoidusti siemenosoitteiksi eri aihepiirien sivustoja, joiden kautta ne kulkivat enintään kolmen linkin päähän ladaten sivujen tiedot. Sivuilta etsittiin erityisesti tunnisteita mahdollisesti suoritettavasta ohjelmasta esimerkiksi .exe-päätteen perusteella. Hakurobotit tutkivat yhteensä noin 20 miljoonaa osoitetta ja he löysivät, että 4,4 %:lla verkkotunnuksista löytyi haittaohjelma. Hakurobotin käyttö haittaohjelmien löytämiseksi osoittautui varsin tehokkaaksi. [20]

3.3 Etiikka

Viime vuosikymmenten aikana, etenkin älypuhelinten yleistyttyä, internetistä ja sosiaalisesta mediasta on tullut kiinteä osa suomalaista yhteiskuntaa ja suomalaisten arkipäivää. Niin kaupankäynti kuin sosiaalinen kanssakäyminenkin on siirtynyt suurelta osin virtuaaliseksi, mikä on johtanut sekä tahallisiin että tahattomiin väärinkäytöksiin. Harva tietää tarkkaan, mikä on oikein ja mikä väärin – normistoa ja lakeja joka tilanteeseen ei vielä edes ole muodostunut. [26] Hakurobotteihin pätevät monet tavallisiakin internetin käyttäjiä koskevat kysymykset, mutta luonteestaan johtuen on niitä tarkasteltava tietyissä tilanteissa eri tavoin. Tärkeitä kysymyksiä ovat ainakin palvelunesto, kulut, yksityisyys sekä tekijänoikeudet [36].

Koska hakurobotit ovat huomattavasti tavallista käyttäjää tehokkaampia, voi palvelimeen kohdistuva kuormitus kertyä suureksi. Etenkin, jos robotin pyyntötiheyttä ei ole rajoitettu, voi palvelimen kyky vastata sille annettuihin pyyntöihin estyä tai huomattavasti rajoittua. Jokainen palvelimelle annettu pyyntö kuluttaa myös resursseja, kuten sähköä ja muistia ja siten rahaa. [36]

Yksityisyydensuojan parantamiseksi on tehty isoja lainsäädännöllisiä toimia, kuten EU:n tietosuoja-asetus GDPR [16]. Yksityisyyttä saatetaan loukata erityisesti, jos isoa määrää henkilökohtaisia tietoja käytetään väärin, kuten jos sähköpostiosoitteita kerätään roskapostin lähettämistä varten. Myös tekijänoikeuksiin tulee kiinnittää huomiota riippuen siitä, mitä tietoa hakurobotilla hankitaan ja miten sitä käytetään ja säilytetään. [36]

Robotin rajausstandardilla voidaan määritellä pelisääntöjä hakurobottien käyttöön palveluissa. Palveluntarjoaja voi esimerkiksi rajata, mitkä robotit voivat milläkin sivuston osalla vierailla ja mitkä eivät. Siten palvelimeen kohdistuvia kustannuksia pystytään tarvittaessa pienentämään. On kuitenkin usein myös sivuston ylläpitäjälle eduksi, että esimerkiksi suurten hakukoneiden, kuten Googlen, hakorobotin pääsevät sivustolle, sillä ajantasaisin tiedoin hakutuloksisissa esiintyminen tuo sivustolle lisää kävijöitä. Suurten hakukoneiden tietokannat kasvavatkin näin muita paremmiksi, mikä tekee niistä yhä suositumpia. Hakukonemarkkinoita dominoi kapea joukko yrityksiä, joiden suosio vain yhä rajautuu, mikä luo epäoikeudenmukaisen vinouman markkinoille. [36]

Sivustot saattavat myös määrittää hakurobotteja koskevia rajoituksia niiden käyttöehdoissa. Rapala kieltää hakurobottien käytön verkkosivustonsa yhteydessä [18]. Sanoma Media rajoittaa automaattisten järjestelmien käyttöä palveluissaan, kuten Helsingin Sanomien verkkosivuilla, ”kappaleiden valmistamiseksi Palvelusta, Sisällöstä tai niiden osista, tai niiden saattamiseksi yleisön saataviin” [29]. Rapalan selkeästä ja

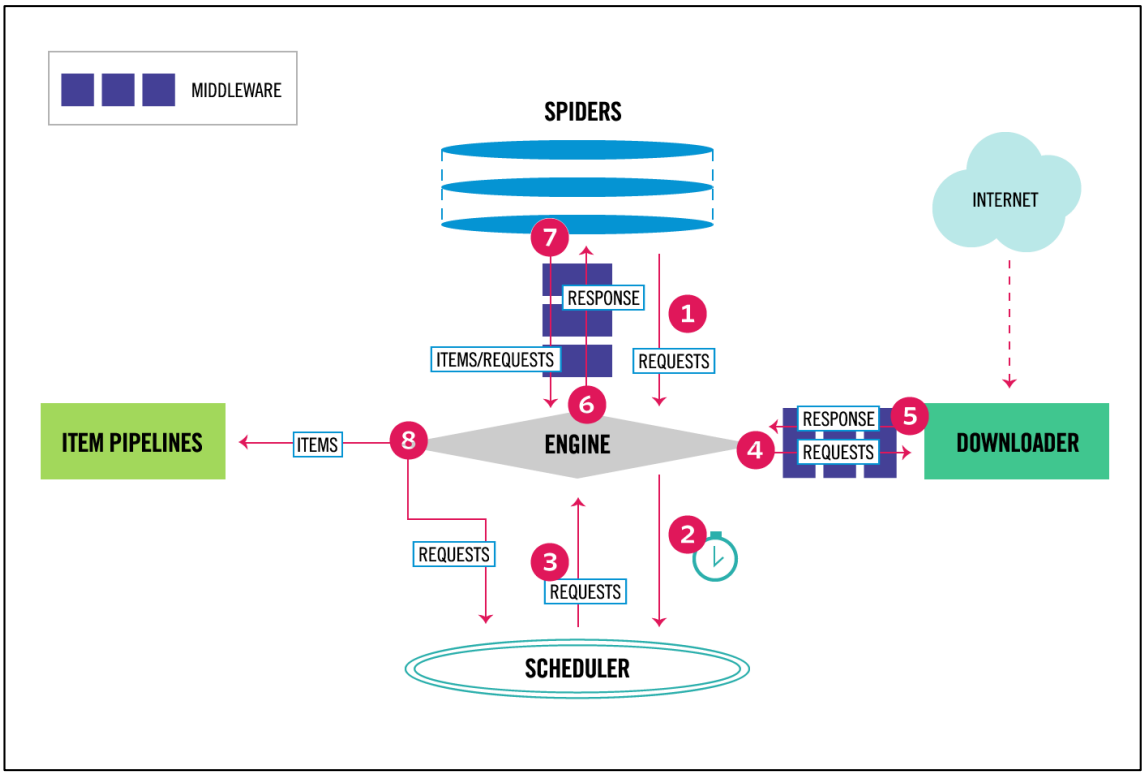
kaiken kieltävästä linjauksesta huolimatta sen Robotin rajausstandardiin ei ole kirjattu estoa kaikille sivuston osoitteille [27]. Hakurobottien voi myös todeta käyneen sivustolla hakemalla sitä hakukoneesta.

4. HAKUROBOTIN TOTEUTUS

4.1 Scrapy

Scrapy on Pythonilla toteutettu avoimen lähdekoodin hakurobotin viitekehys. Sillä toteutettuja hakurobotteja voidaan käyttää moniin tarkoituksiin, kuten tiedonlouhintaan, tiedonkäsittelyyn ja arkistointiin. Siinä on valmiina paljon laajennuksia ja väliohjelmistoja esimerkiksi evästeiden ja Robotin rajausstandardin hallitsemiseen. [24] Scrapya on hyödynnetty tämän kandidaatintyön lisäksi useissa eri konteksteissa – niin tutkimuksessa kuin kaupallisestikin. Farooq et al. [13] keräsivät tietoa japanilaisilta kiinteistömarkkinoiden verkkosivuilta, Dallmeier at al. [11] kehittivät tavan verkkosovellusten automaattitestaamiseen ja Shi et al. [33] latasivat uutissivuja. Tämän kandidaatintyön konstruktiivinen osa mukailee perusidealtaan Shi et al:n työtä. Liiketoiminnassaan Scrapya käyttäviä yrityksiä ovat esimerkiksi Parse.ly, Lyst sekä DirectEmployers, jotka keräävät ja käsittelevät tietoa uutisia, muotia sekä työpaikkailmoituksia sisältäviltä sivustoilta [19].

Kuva 2 esittää yleiskatsauksen Scrapyn ja sen komponenttien toiminnasta. Punaiset nuolet kuvaavat datan liikettä niiden välillä ja tummansiniset suorakaiteet väliohjelmistoja (middleware). [2]



Kuva 2. *Scrapyn komponentit ja datan liike niiden välillä [2].*

Scrapyn viitekehyksen (Scrapy framework) ytimessä on Moottori (Engine). Se vastaa tiedon liikkumisesta eri komponenttien välillä. Hämähäkki on luokka, jonka toiminnan Scrapyn käyttäjä itse määrittää. Se saa käsiteltäväkseen HTML-dataa Lataajalta (Downloader), joka kommunikoi verkkosivujen kanssa HTTP-ptotokollalla. Hämähäkki voi jäsentää saamastaan datasta esimerkiksi tietynlaisia linkkejä, joista halutaan hakea tietoa myöhemmin. Jos muuta tietoa halutaan tallentaa, se voi luoda niistä Esineitä (Items), jotka voidaan tallentaa paikalliseksi tietokannaksi. [2, 33]

Tiedon liikkuminen komponenttien välillä alkaa Hämähäkkiin määritellyistä aloitus-URL-osoitteista. Hämähäkki välittää osoitteet Moottorille, joka taas välittää ne Aikatauluttajalle tallennettavaksi. Moottori pyytää Aikatauluttajalta yhden osoitteen kerrallaan siirrettäväksi Lataajalle. Se lähettää saamaansa osoitteeseen GET-pyynnön. Sivun latauduttua Lataaja välittää saamansa vastauksen Lataajan väliohjelmistojen kautta Moottorille, josta se siirtyy Hämähäkille Hämähäkin väliohjelmistojen kautta. Mikäli Hämähäkki kerää uusia osoitteita uutta hakua varten, se välittää ne taas Moottorille ja sitä kautta Aikatauluttajalle. Luomansa Esineet Hämähäkki välittää niin ikää Moottorin kautta Esineputkistoon (Item Pipelines). Siellä Esineitä käsitellään ja voidaan tarkastaa, onko tieto sellaista, jota halutaan tallentaa. Esineputkisto voi siirtää oikeelliseksi toteamansa tai muokkaamansa tiedon tietokantaan. Kun tieto on tallennettu, Moottori

pyytää Aikatauluttajalta uuden osoitteen ja prosessi alkaa alusta. Hakurobotin toiminta jatkuu niin kauan, kunnes Aikatauluttajalla ei ole enää uusia osoitteita annettavaksi. [2, 33]

4.2 Suoritetun haun konteksti

Työ suoritettiin koronavirus SARS-CoV-2 aiheuttaman pandemian aikana. Aihe hallitsi uutissisältöä lähteestä riippumatta; muista aiheista uutisointia oli huomattavan vähän. Tilanne tarjosi mahdollisuuden hakurobotin soveltamiseen ajankohtaisessa ja merkityksellisessä kontekstissa.

Työssä toteutettua hakurobottia ajettiin aluksi Windows 10 -käyttöjärjestelmän PC:ltä ja se toimi tunteja moitteettomasti. Jossain kohtaa kävi kuitenkin ilmi, että tietokone oli käynnistynyt uudelleen, ohjelman suoritus keskeytynyt ja hakurobotin keräämät tiedot menetetty. Uudelleenkäynnistymisen syystä ei ollut merkkejä, eikä ohjelman suorituksesta ollut tallentunut lokitietoja. Tietokoneen huono toimintavarmuus johti päätökseen pilvipalvelun käyttöönotosta.

Käytettäväksi pilvipalveluksi valikoitui Scrapinghub. Se on yksi suurimmista Scrapyn yhteistyökumppaneista [19] ja tarjoaa Scrapyn käytölle rajatuilta ominaisuuksilta ilmaisen ja yhteensopivan alustan [31]. Palvelu tarjoaa rajatuista ominaisuuksista huolimatta myös ominaisuuksia, joita ei omalta tietokoneelta ajettaessa ole. Esimerkiksi haun päätteeksi kerätyt Esineet voi ladata haluamassaan tiedostomuodossa, kun taas komentoriviltä ajettaessa on haluttu tiedostomuoto tiedettävä etukäteen.

Hakurobotti suunniteltiin siten, että se lataa ja käsittelee dataa Yleisradion uutissisällöstä. Se jäsentää löytämänsä linkit, suodattaa niistä pois muuta kuin uutissisältöä sisältävät ja laittaa loput linkkijonoon. Linkin takaa löytyvästä artikkelista se etsii koronavirukseen viittaavia avainsanoja ja sellaisen löytäessään tallentaa tekstin, siihen liittyviä oheistietoja sekä hakuun liittyviä tietoja tietokantaan.

Jotta työssä pystyttiin keskittymään olennaiseen asiaan, eli kysymyksiin käytäntöjen soveltamisesta, pyrittiin tekninen toteutus pitämään mahdollisimman rajattuna. Sivuston eri sivujen rakenne on yleensä samanlainen, jolloin niiltä tiedon jäsentäminen on yksinkertaista. Haun rajaaminen yhteen sivustoon mahdollisti myös luvallisen toimimisen varmistamisen. Myös tietoturvakysymykset olivat helpompia rajatulla kohteella. Vaikka aina on mahdollisuus ohjelmointivirheen tai odottamattoman tilanteen takia päätyä tekemään jotain, mihin ei ole varautunut, on ison, julkisen toimijan julkisesti tarjoaman sisällön käsitteleminen suhteellisen turvallista.

4.3 Sovellettavat käytännöt ja algoritmit

4.3.1 Valinnat

Tapauksessa, jossa tutkittava sivusto on etukäteen tarkasti rajattu, on linkkien valintojen rooli hakurobotin suunnittelussa vähemmän merkittävä, kuin esimerkiksi hakukoneen tapauksessa. Hakukoneen indeksointia suorittava hakurobotti pyrkii lataamaan mahdollisimman monta eri sivua saadakseen laajan otannan internetistä. Tähän tarkoitukseen sopivin algoritmi on leveyteen ensin -haku [5].

Yhdellä uutissivulla olevat linkit ovat ehdotuksia muista uutisista. Sivustot pyrkivät kohdentamaan sisältöä käyttäjälle tarjoamalla kyseistä uutista vastaavia artikkeleita. Tällöin hakurobotin keräämät linkit johtavat aihepiiriltään ja julkaisuajaltaan samankaltaisiin uutisiin. Kuitenkin, mitä pidemmälle linkkejä pitkin edetään, sitä enemmän variaatiota aihepiirissä esiintyy. Leveyteen ensin -haulla hakurobotti lataa kaikki samalta sivulta keräämänsä linkit ennen seuraavaan siirtymistä. Tällöin aihepiirin variaatio on hyvin vähäistä. Syvyyteen ensin -haussa sen sijaan linkkejä pitkin kuljetaan sananmukaisesti nopeammin syvemmälle, eli yhden linkin takaa löytyviä tutkitaan ennen kuin saman sivun muita linkkejä. Tällöin aihepiirin variaatiota pystytään maksimoimaan.

Scrapyn Aikatauluttaja toimii oletuksena LIFO-periaatteella, eli tällöin tehdään syvyyteen ensin -haku. Tässä työssä käytettiin sekä syvyyteen että leveyteen ensin hakuja. Hakujen vertailu on luvussa 6. Leveyteen ensin -algoritmin saa käyttöön kirjoittamalla asetustiedostoon [15]:

```
DEPTH_PRIORITY = 1
SCHEDULER_DISK_QUEUE = 'scrapy.squeues.PickleFifoDiskQueue'
SCHEDULER_MEMORY_QUEUE = 'scrapy.squeues.FifoMemoryQueue'
```

4.3.2 Kohteliaisuus

Hakua suoritettaessa yhdelle sivustolle on hakurobotin aiheuttamien vaikutusten palvelimille perustavanlaatuisesti erilainen, kuin laajalle eri sivustoja kattavan haun tapauksessa. Kun kaikki hakurobotin pyynnöt kohdistuvat samaan sivustoon, kohdistuu kaikki rasite samalle palvelimelle. Yleisradio julkaisee kymmeniä uutisia päivittäin, minkä vuoksi uutissisältöä on kerääntynyt saataville vuosien saatossa sadoille tuhansille sivuille. Mikäli hakurobotin pyyntötiheyttä ei rajoiteta, lähettää se suuren määrän pyyntöjä huomattavan pitkän aikaa. Lähettämällä satakin pyyntöä sekunnissa vuorokauden ajan tulee se lähettäneeksi silti alle sata tuhatta pyyntöä. Sivuston ylläpitäjä voi estää käyttäjän tai robotin pääsyn sivustolle, mikäli havaitsee sivustolle haitallista toimintaa. Sen sijaan pitkä viive ei ole haun suorittajan kannalta toivottavaa, sillä mitä

pidempi viive on, sitä vähemmän tuloksia saadaan. Koska Robotin rajausstandardiin ei ole mahdollista määrittää pyynnöille vähimmäisviivettä pyyntövälille, jää sen määrittäminen hakurobotin suunnittelijalle.

Yleisradion Robotin rajausstandardi kieltää kaikilta hakuroboteilta tiettyjen polkujen käytön. Kuitenkin suuri osa sisällöstä, kuten uutiset, jäävät tämän kiellon ulkopuolelle. [28] Sivustolle ei ole määritetty käyttöehtoja, kuten monien kaupallisten uutissivustojen tapauksessa. Luvan varmistamiseksi työtä varten lupa uutissivuston käyttämiseen pyydettiin ja saatiin sähköpostitse. Lupa kattoi ainoastaan hakurobotin käytön tutkimuskäytössä.

Scrapy tarjoaa automatisoidun toiminnallisuuden Robotin rajausstandardin pyytämiseen ja noudattamiseen. Käyttäjän on huolehdittava, että toiminnallisuus on otettu käyttöön asetuksissa:

```
ROBOTSTXT_OBEY = True
```

Scrapyn asetuksissa on myös viiveen hallintaan valmiita ominaisuuksia. Hakuun Yleisradion uutissisältöön vähimmäisviiveeksi asetettiin viisi sekuntia:

```
DOWNLOAD_DELAY = 5
```

Viive on pienempi kuin tunnettujen hakurobottien usein käyttämät viiveet [7]. On huomioitava, että edellä määrätty asetus on nimenomaan viiveen vähimmäisajalle. Scrapyssa on ominaisuus, joka arvioi itse tarvetta viiveelle ja säätää sitä tarpeen mukaan. Sen saa käyttöön asetuksista:

```
AUTOTHROTTLE_ENABLED = True
```

Tällöin Hakurobotti mittaa aikaa, joka sivustolla kestää vastauksen lähettämiseen. Jos aika on pitkä, se arvioi, että sivusto on ruuhkautunut, jolloin se pidentää viivettä pyyntöjen välillä. [3] Tässä työssä luotettiin automaattisen viiveenkorjaimen pitävän viiveen sellaisena, ettei se kuormita palvelinta liikaa.

4.3.3 Rinnasteisuus ja uudelleenvierailu

Työssä pyritään maksimoimaan sivustollavierailutiheys laajemman tuloskannan saamiseksi, mutta samalla toimimaan palveluntarjoajaa kohtaan kohteliaana. Siten palvelinta kohtaan asetettu kuorma on enimmäismäärässään. Rinnakkain ajettavat hakurobotit nostaisivat tehokkuutta, mutta samalla kuormittaisivat suoraan verrannollisesti palvelinta. Rinnakkain usean hakurobotin käyttö ei siis tässä tapauksessa toisi etua.

Myöskään uudelleenvierailu ei ole tässä tapauksessa keskeisin kysymys. Tavoitteena ei ole ylläpitää reaaliaikaista kuvaa uutissisällöstä, vaan pikemminkin saada otanta siitä, mihin hakurobotti pystyy senhetkistä materiaalia hyödyntäen. Toisaalta, koska hakurobotin käyttämät hakusanat ovat nimenomaan työn suoritusajalle ajankohtaisia, ei kysymys uudelleenvierailusta ole myöskään mitätön. Tapauksessa, jolloin pandemia on jo väistynyt ja uutisia hallitsee muut yhteiskunnalliset teemat, tulevat haun tulokset samaa robottia käyttäen olemaan varsin erilaiset. Aikajänne muutoksen merkityksellisyyteen on kuukausista vuosiin. Työn suorittamisen aikajänne on viikoista kuukausiin, eikä siten tulosten kannalta merkittävä.

4.4 Hakurobotin keräämä tieto

Työssä toteutettu Hämähäkki jäsentää Lataajan sille välittämää tietoa uutissivuista. Se luo tiedosta Esineitä, jotka siirtyvät Esineputkistoon käsiteltäväksi. Siellä yksittäisestä Esineestä tarkastetaan, täyttääkö se tahdotut ehdot. Sen mukaan Esine joko tallennetaan tai hylätään.

Hakurobotilla on mahdollista luoda samanaikaisesti useita eri Esineitä, mutta tässä työssä luodaan vain yhdenlaisia. Esineen määrittely on items.py-tiedostossa. Luotava esine on taulu, jonka sarakkeet määritellään MycrawlerItem-luokassa. Nimen Mycrawler luokka on saanut projektin nimestä Mycrawler. Taulun sarakkeet ovat osoite (url), artikkelin julkaisupäivä (publish_date), kirjoittaja (author), otsikko (header), uutisen tekstisisältö (text), latausaika (crawl_time) ja kieli (language). Sarake saadaan määriteltyä komennolla scrapy.Field().

```
class MycrawlerItem(scrapy.Item):
    url = scrapy.Field()
    publish_date = scrapy.Field()
    author = scrapy.Field()
    header = scrapy.Field()
    text = scrapy.Field()
    crawl_time = scrapy.Field()
    language = scrapy.Field()
```

Kun Hämähäkki on luonut Esineen ja tallentanut sen kentille arvot, se välittää sen Esineputkistoon. Scrapy luo Esineputkistoon luokan, jonka nimi tässä tapauksessa on `MycrawlerPipeline`, ja sille esineenkäsittelyfunktion `process_item`. `MycrawlerPipeline` tallentaa muuttujiin `text` ja `language` Esineen tiedot uutistekstistä sekä sivulla käytetystä kielestä. Muuttujista ensimmäiseksi tarkistetaan,

onko kieli suomi. Sen jälkeen siitä uutistekstistä etsitään avainsanaa *korona* eri kirjoitusasuissa. Kielen ollessa oikea ja avainsanan puuttuessa Esine tallennetaan komennolla `return item`. Muussa tapauksessa aiheutuu poikkeus `DropItem`.

```
from scrapy.exceptions import DropItem

class MycrawlerPipeline(object):
    def process_item(self, item, spider):
        text = item['text']
        header = item['header']
        language = item['language']
        if language == "fi" \
                and "korona" not in text\
                and "Korona" not in text\
                and "KORONA" not in text:
            return item
        else:
            raise DropItem("korona mentioned at "\
                + item['url'])
```

4.5 Pyyntöjen käsittely parse-funktiossa

Scrapy-projektiin luotaessa uusi Hämähäkki Scrapy luo sille luokan ja siihen funktion `parse`. Luokalle luodaan myös muuttujat nimelle, sallituille verkkotunnuksille ja aloitussivuille. Ne määritetään automaattisesti projektinluontikomennon perusteella, mutta niitä voi halutessaan muokata sen jälkeenkin. Hakurobotin suorituksen alkaessa lista aloitussivustoista välitetään Aikatauluttajalle, joka välittää linkit yksitellen ladattavaksi. Työssä toteutettu Hämähäkki on nimeltään `KoronaSpider`.

```
class KoronaSpider(scrapy.Spider):
    name = 'krspider'
    allowed_domains = ['yle.fi']
    start_urls = ['https://yle.fi/uutiset/']
```

Saatuaan vastauksen verkkosivulta hakurobotti kutsuu Hämähäkin parse-funktiota. Sen parametri `response` pitää sisällään verkkosivun vastauksena lähettämän datan. Sitä käsittelemällä css- tai xpath- komennoilla saadaan jäsenneltyä verkkosivun tietoja. [35] `KoronaSpider` jäsentää jokaiselta ladatulta uutissivulta artikkelin tekstikentät (`paragraphs`), otsikon (`h1`), artikkelin julkaisupäivämäärän (`date`), kirjoittajan

(author), sivulta löytyvät linkit (urls) ja kielen (language). KoronaSpider jäsentää tiedot response-parametrista xpath-komennoilla.

```
paragraphs = response.xpath('//p/text()').extract()
h1 = response.xpath('//h1/text()').extract_first()
date = response.xpath(\
    '//*[@class="yle__article__date--published"]\
    /text()').extract_first()
author = response.xpath(\
    '//*[@class="yle__article__author__name__text"]\
    /text()').extract_first()
urls = response.xpath('//a/@href').extract()
language = response.xpath('/html/@lang').extract_first()
```

Parse-funktio kerää myös muuta hyödyllistä tietoa. Se tallentaa URL-osoitteen, josta vastaus on saatu sekä kellonajan, jolloin vastaus on käsitelty:

```
now = datetime.now().time()
this_url = response.url
```

Se myös kokoaa eri tekstikenttien tekstin yhden muuttujan merkkijonoksi text:

```
text = ""
for p in paragraphs:
    text += p
```

Se luo uuden Esineen, eli MycrawlerItem-luokan instanssin, nimeltään site_item. Esineen kenttien arvoksi annetaan aiemmin jäsennellyt muuttujat, sivulta löytyneitä osoitteita lukuun ottamatta. Komento yield lähettää valmiin Esineen Esineputkistoon.

```
site_item = MycrawlerItem()
site_item['url'] = this_url
site_item['publish_date'] = date
site_item['author'] = author
site_item['header'] = h1
site_item['text'] = text
site_item['crawl_time'] = now.strftime("%H:%M:%S")
site_item['language'] = language
yield site_item
```

Lopuksi käsitellään sivulta kerätyt linkit. Luokan allowed_domains -listan määrittely pitää huolen, että muita kuin Yleisradion sivuja ei oteta huomioon. Kuitenkin, jotta

voidaan varmistua, että linkeistä valitaan vain uutissisältöä käsittelevät osoitteet, tarkistetaan, onko niissä polkua `/uutiset/`. Hämähäkin välittämien linkkien on oltava myös kokonaisia, eli alkavan `https://yle.fi`. Niihin, joista alkuosa puuttuu, se lisätään. `Yield`-komento välittää osoitteet Moottorin kautta Aikatauluttajalle.

```
urls_to_visit = []
fullurl = ""

for url in urls:
    if "https://yle.fi/uutiset" in url:
        fullurl = url
        urls_to_visit.append(fullurl)
    elif "/uutiset/" in url:
        fullurl = "https://yle.fi" + url
        urls_to_visit.append(fullurl)

for url in urls_to_visit:
    yield scrapy.Request(url)
```

Hakurobotin suunnittelussa Scrapylla käyttäjän ei tarvitse ottaa huomioon duplikaattiosoitteiden käsittelyä. Scrapyn Dupefilter-luokka pitää kirjaa osoitteista, joilla hakurobotti on jo käynyt, eikä ota niitä sen jälkeen enää huomioon [32].

5. HAUN TULOKSET JA ARVIO

5.1 Ympäristön soveltuvuus

Scrapy tarjoaa toimivan ja helposti käyttöönotettavan kokonaisuuden hakurobotin tekemiseen. Siinä on paljon sisäänrakennettuja ominaisuuksia, joiden käytöstä käyttäjän ei tarvitse huolehtia muuten kuin ottamalla ne käyttöön asetuksista. Myös toiminnallisuuksien muokkaaminen haluamakseen onnistuu asetuksia säätämällä ja omia ratkaisuja toteuttamalla.

Pidempikestoisen haun tekeminen omalta tietokoneelta osoittautui ongelmalliseksi, eikä selvyyttä ja syytä asialle saatu. Vastaavilta ongelmilta vältyttiin ottamalla Scrapinghub käyttöön, jossa hakurobotin ajaminen sujui ongelmitta. Tulosten tutkiminen ja analysointi oli palvelun kautta helppoa – esineet tallentuivat palvelimelle ja ne saatiin ladattua halutussa tiedostomuodossa omalle tietokoneelle. Tunnin käyttökatto yhdelle haulle rajoitti tuloksia, mutta oli kuitenkin tässä tapauksessa riittävä.

5.2 Kerätty data ja sovelletut algoritmit

Taulukossa 1 on esitetty tulokset kahdelle suoritetulle haulle. Molemmat suoritettiin lähes samaan kellonaikaan ja ne kestivät lähes yhtä kauan. Myös pyyntöjä lähetettiin lähes yhtä paljon.

Taulukko 1. *Hakualgoritmien vertailu.*

Haku-algoritmi	Kerättyjä Esineitä	Valideja Esineitä	Pyyntöjä	Kesto	Päivämäärä	Aloitusaika
Leveyteen ensin	4	0	615	01:03:39	23.4.2020	11:58:06 UTC
Syvyyteen ensin	5	4	620	01:03:02	20.4.2020	12:03:41 UTC

Merkittävin ero hakujen välillä huomataan, kun tarkastellaan niiden keräämiä Esineitä. Molemmissa hauissa alle prosentilla haetuista sivuista ei ollut uutistekstissä mainintaa koronaviruksesta. Kerättyjen Esineiden määrä on molemmissa hauissa likimain sama. Ero tulee esiin vasta, kun tarkastellaan kerättyjen Esineiden sisältöä. Syvyyteen ensin -haussa yhtä lukuun ottamatta jokainen Esine on oikea uutisartikkeli, kun taas leveyteen ensin -haussa yksikään ei ole. Virheelliset tulokset johtuvat pääasiassa sivuista, joiden toteutus poikkeaa normaalista, eikä tällaisiin hakurobotin suunnittelussa ollut varauduttu. Ne oli toteutettu käyttäen interaktiivisia elementtejä, joista tekstin haku samoilla xpath-

komennoilla ei onnistunut. Yksi leveyteen ensin -haun Esineistä oli osoitteesta https://yle.fi/saa/. Lokitietoja tarkastelemalla se on ainut linkki, joka poikkeaa /uutiset/ -polusta. Syytä poikkeamalle ei selvinnyt. Syvyyteen ensin -haussa vastaavia poikkeamia ei ollut.

Tuloksista voidaan todeta hypoteesi syvyyteen ensin -haussa esiintyvästä suuremmasta aihepiirien varianssista todeksi. Sen löytämät koronavirusta käsittelemättömät sivut olivat julkaistu alkuvuodesta 2020 tai loppuvuodesta 2019. Leveyteen ensin -haussa ladatut sivut olivat aihepiiriltään ja julkaisuajaltaan enemmän samankaltaisia. Siten sen latasi enemmän tuoreita uutisia, joiden joukossa todennäköisyys löytää ehtoja vastaavia tuloksia oli pienempi.

5.3 Sovelletut käytänteet

Leveyteen ensin -haussa hieman pidemmästä kokonaiskestosta huolimatta pyyntöjä lähetettiin viisi vähemmän, mikä voisi olla merkki automaattisen viiveensäädön käytöstä. Toisin sanoen 23.4.2020 hakurobotti saattoi arvioida verkkosivun olevan ruuhkaisempi kuin toisen haun aikaan. Ero on kuitenkin varsin pieni, eikä varmaa johtopäätöstä voida sen perusteella vetää. Molemmissa tapauksissa hakurobotti lähetti noin 60 minuutissa noin 600 pyyntöä, mistä voidaan laskea keskimääräisen viiveen olleen noin 10 sekuntia. Automaattinen viiveenkorjain oli siis tulkinnut minimiviiveeksi määritetyn viisi sekuntia liian pieneksi.

Hakurobotin käyttäjän on kuitenkin vaikea saada todenmukaista kuvaa määritettyjen minimiviiveen ja automaattisen viiveenkorjaimen vaikutuksesta palvelimelle. Scrapyn kehittäjät kuvaavat korjaimen määrittämää viivettä optimaaliseksi [3]. Kuitenkaan todellista palvelimelle aiheutunutta kuormaa ei voi arvioida ilman ylläpitäjän arviota tilanteesta, jollaista ei tätä työtä varten saatu.

Kuten algoritmien tapauksessa, Scrapyn asetustiedosto tarjoaa helpon alustan kohteliaan hakurobotin määrittämiselle. Automaattiselle viiveenkorjaimelle on monia muitakin asetuksia esimerkiksi rinnakkain ajettaville hakuroboteille, joita ei tässä työssä hyödynnetty [3]. Scrapinghub tarjoaa maksullisen ominaisuuden automatisoituun uudelleenvierailuun. Hakurobotille voi asettaa säännöllisen ajoajan, jolloin palvelu ajaa ohjelman määritetyin intervallein. [30] Säännöllistä uudelleenvierailua ei tässä työssä käytetty vaan haut aloitettiin manuaalisesti.

6. JOHTOPÄÄTÖKSET

Hakukoneista on tullut kiinteä osa internetin käyttökontekstia. Sitä myöten myös hakurobottien rooli yksittäisenkin internetin käyttäjän kannalta on huomattava. Tarjolla on valtavia määriä informaatiota, joka hakurobottien keräämänä ja hakukoneiden jäsentämänä on tuotu jokaisen internetyhteyden päässä olevan saavutettavaksi. Palveluntarjoajien näkökulmasta hakuroboteilla voi olla sekä positiivisia että negatiivisia vaikutuksia. Koska sivustojen tarjoajille on usein tavoiteltavaa, että mahdollisimman moni käyttäjä päätyy sivustolle, on myös ajantasaisin tiedoin hakukoneiden hakutuloksissa esiintyminen eduksi. Kuitenkin jokainen, niin ihmisen kuin robotinkin, lähettämä pyyntö palvelimelle kuormittaa sitä ja aiheuttaa kustannuksia. Onkin tavoiteltavaa, että hakurobottien suunnittelijat ja palveluntarjoajat löytävät molempia hyödyntävän tasapainon siinä, kuinka tiheään robotit sivustoa kuormittavat. On tehtävä kompromisseja ajantasaisen tiedon saatavuuden ja sekä palvelimiin että hakurobottein käyttäjiin kohdistuvien kustannusten välille.

Vaikka hakukoneyhtiöt ovat sekä toimintalaajuudelta että taloudellisesti mitattuna hakurobottien käytön suurimpia toimijoita, on etenkin kontekstiltaan rajatulla tiedonkeräämisellä todettu olevan merkittäviä käyttökohteita. Hakuroboteista on tullut monialaisen tutkimuksen työkaluja, joilla voidaan tutkia monimutkaisiakin yhteiskunnallisia ilmiöitä. Dataa saadaan kerättyä paljon lyhyessä ajassa, mikä mahdollistaa suurten kokonaisuuksien ajantasaisen analysoinnin ja siten myös kehityksen analysoinnin ajan kuluessa. Muutosten perusteella voidaan muodostaa malleja ja ennustaa kehityksen kulkua tulevaisuuteen. Kaupallisten sovellutusten kirjo on hyvin laaja, sillä hakurobotteja voidaan räätälöidä kunkin tahon tarpeisiin tapauskohtaisesti.

Oli hakurobotin motiivi kerätä tietoa sitten hakukonetta, muuta kaupallista toimintaa tai tutkimuskohdetta varten, ovat keskeisimmät sovellettavat suunnitteluperiaatteet samoja. Tärkeimmät suunnitteluperiaatteet ovat valinnat, kohteliaisuus, rinnasteisuus uudelleenvierailu, turvallisuus sekä eettisyys. On tapauskohtaista, miten paljon huomiota kullekin tulee antaa. Esimerkiksi hakukoneiden indeksointia varten uudelleenvierailu on keskeinen kustannuksiin vaikuttava tekijä, mutta kertaluontoisessa tiedonkeruussa sillä ei luonnollisesti ole merkitystä.

Suunnitteluperiaatteiden soveltamista tutkittiin Scrapylla. Viitekehykseen ohjelmoitiin hakurobotti, joka hakee Yleisradion uutissisältöä rajaten sisältöä pois avainsanojen

perusteella. Kontekstin määrittämisessä tärkeimmät seikat liittyivät tietoturvaan sekä eettisyyteen, mikä johti rajatun, julkisen toimijan tarjoaman kohdeaineiston valitsemiseen. Yksi merkityksellisimmistä seikoista oli tutkia, kuinka kirjallisuudessa esitettyjä keskeisimpiä suunnitteluperiaatteita voidaan tässä kontekstissa soveltaa. Kysymyksistä yksi merkityksellisimmistä oli tässä tapauksessa valintakäytäntöjen määrittäminen. Se määräsi näkyviltä osin haun tulosten suunnan ja siten myös kerätyn tiedon laadun ja määrän. Myös kohteliaisuus oli työssä suuressa osassa, vaikka sen osuus tuloksissa onkin näkymätön, tai näkyy pikemminkin tulosten suppeudessa.

Keskeistä oli kuitenkin saavutettu käsitys hakurobottien merkityksestä palveluntarjoajan kannalta. Tässä tapauksessa Yleisradio ei saavuta edes sitä hyötyä, mitä hakukoneiden robottien sallimisesta aiheutuu, mutta silti he antoivat luvan tutkimuskäyttöön. Heidän saavuttamansa hyöty lieneekin jotain huomattavasti vähemmän konkreettista, kuten asiakaspalvelun kautta saavutettu luottamussuhteen vankistuminen ja brändiarvon kohoaminen tai ylläpito. Palveluntarjoajalle lienee selvää, että hyväntahtoiset sivuston käyttäjät ovat heille eduksi. Hakurobotin kehittäjän näkökulmasta kohtelias ja lupaa kunnioittava toiminta on ensisijaista, sillä palveluntarjoajalla on mahdollisuus evätä robotin pääsy sivustolle ja siten keskeyttää harjoitettu toiminta välittömästi. Havainnot tukevat ajatusta, että kohteliaisuus on periaate, jota molempien osapuolien on syytä vaalia.

Koska hakurobotin suunnittelu on aina tapauksesta ja käyttökontekstista riippuvaa, ei tämänkään työn hakurobotti tarjoa muuta kuin yhden esimerkin periaatteiden soveltamisesta valitussa kontekstissa. Se ei kuitenkaan tarkoita, etteikö työssä saavutetuilla tuloksilla olisi merkitystä missään muussa tilanteessa. Sovelletut suunnitteluperiaatteet eivät alkujaankaan ole määritetty työn kaltaiseen kontekstiin, vaan niitä on tutkittu enimmäkseen hakukoneiden kannalta. Sekään ei tarkoita, etteivätkö ne olisi merkityksellisiä muutakin, kuin indeksointia tehtäessä. Vaikka hakurobottien käyttökontekstit olisivat aina tapauskohtaisia, on niissä aina myös huomattavasti samankaltaisuuksia. Jo se, että ne kaikki toimivat osana asiakas-palvelinmallia noudattavaa internetiä, tekee käyttökontekstista, ja siten myös siihen sovellettavista suunnitteluperiaatteista, huomattavan yhdenmukaisen kokonaisuuden.

LÄHTEET

[1] About us - Best Web Scraping Services Agency, Hir InfoTech, verkkosivu. Saatavissa (viitattu 1.4.2020): https://hirinfotech.com/about/

[2] Architecture, Scrapy Documentation. Saatavissa (viitattu 17.4.2020): https://docs.scrapy.org/en/latest/topics/architecture.html

[3] Autothrottle, Scrapy Documentation. Saatavissa (viitattu 20.4.2020): https://docs.scrapy.org/en/latest/topics/autothrottle.html

[4] Billig, J., Danilchenko, Y., & Frank, C. E. (2008, September). Evaluation of google hacking. In *Proceedings of the 5th annual conference on Information security curriculum development* (pp. 27-32).

[5] Boldi, P., Codenotti, B., Santini, M., & Vigna, S. (2004). Ubicrawler: A scalable fully distributed web crawler. *Software: Practice and Experience*, *34*(8), 711-726.

[6] Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine.

[7] Castillo, C. (2005, June). Effective web crawling. In *Acm sigir forum* (Vol. 39, No. 1, pp. 55-56). New York, NY, USA: Acm.

[8] Cho, J. (2001). Crawling the web: discovery and maintenance of large-scale web data. *Computer science. Stanford University*.

[9] Clement, J. (28.2.2020) Global mobile data traffic 2017-2022, Statista. Saatavissa (viitattu 30.3.2020): https://www.statista.com/statistics/271405/global-mobile-data-traffic-forecast/

[10] Cohen-Almagor, R. (2013). Internet history. In *Moral, ethical, and social dilemmas in the age of technology: Theories and practice* (pp. 19-39). IGI Global.

[11] Dallmeier, V., Burger, M., Orth, T., & Zeller, A. (2013, January). Webmate: Generating test cases for web 2.0. In *International Conference on Software Quality* (pp. 55-69). Springer, Berlin, Heidelberg.

[12] Data Mining, Georanker, verkkosivu. Saatavissa (viittattu 6.5.2020): https://www.georanker.com/data-mining

[13] Farooq, B., Husain, M. S., & Suaib, M. (2018). CRAWLING OF JAPANESE REAL-ESTATE WEBSITES USING SCRAPY. *International Journal of Advanced Research in Computer Science*, *9*(Special Issue 2), 64.

[14] Floridi, L. (2009). The information society and its philosophy: Introduction to the special issue on “the Philosophy of Information, its Nature, and future developments”. *The Information Society*, *25*(3), 153-158.

[15] Frequently Asked Questions, Scrapy Documentation. Saatavissa (viitattu 20.4.2020): https://docs.scrapy.org/en/latest/faq.html

[16] General Data Protection Regulation GDPR. Saatavissa (viitattu 8.4.2020): https://gdpr-info.eu/

[17] Heydon, A., & Najork, M. (1999). Mercator: A scalable, extensible web crawler. *World Wide Web*, *2*(4), 219-229.

[18] Käyttöehdot, Rapala. Saatavissa (viitattu 8.4.2020): https://www.rapala.fi/content/rapala-legal/customer-terms-conditions.html

[19] Meet the Scrapy pros, Scrapy. Saatavissa (viitattu 9.4.2020): https://scrapy.org/companies/

[20] Moshchuk, A., Bragin, T., Gribble, S. D., & Levy, H. M. (2006, February). A Crawler-based Study of Spyware in the Web. In *NDSS* (Vol. 1, p. 2).

[21] Najork, M., & Wiener, J. L. (2001, April). Breadth-first crawling yields high-quality pages. In *Proceedings of the 10th international conference on World Wide Web* (pp. 114-118).

[22] Nemeslaki, A., & Pocsarovszky, K. (2012). Supporting e-business research with web crawler methodology. *Society and Economy*, *34*(1), 13-28.

[23] Nisar, T. M., & Yeung, M. (2018). Twitter as a tool for forecasting stock market movements: A short-window event study. *The journal of finance and data science*, *4*(2), 101-119.

[24] Overview, Scrapy Documentation. Saatavissa (viitattu 9.4.2020): https://docs.scrapy.org/en/latest/intro/overview.html

[25] Pant, G., Srinivasan, P., & Menczer, F. (2004). Crawling the web. In Web Dynamics (pp. 153-177). Springer, Berlin, Heidelberg.

[26] Rader, M. H. (2002). Strategies for teaching internet ethics. *INSTITUTION Delta Pi Epsilon Society, Little Rock, AR.*, 116.

[27] Robotin rajauustandardi, Rapala. Saatavissa (viitattu 8.4.2020 https://www.rapala.fi/robots.txt

[28] Robotin rajauustandardi, Yleisradio. Saatavissa (viitattu 26.4.2020): https://www.yle.fi/robots.txt

[29] Sanoma Media Finlandin yleiset käyttöehdot, Sanoma Media. Saatavissa (viitattu 8.4.2020): https://oma.sanoma.fi/asiakastuki/yleiset/kayttoehdot

[30] Scheduling Periodic Jobs (15.4.2020), Scrapinghub Support Center. Saatavissa (viitattu 5.5.2020): https://support.scrapinghub.com/support/solutions/articles/22000200419-scheduling-periodic-jobs

[31] Scrapy Cloud Pricing, Scrapinghub. Saatavissa (viitattu 20.4.2020): https://scrapinghub.com/scrapy-cloud#pricing

[32] Settings, Scrapy Documentation. Saatavissa (viitattu 20.4.2020): https://docs.scrapy.org/en/latest/topics/settings.html

[33] Shi, Z., Shi, M., & Lin, W. (2016, December). The Implementation of Crawling News Page Based on Incremental Web Crawler. In *2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD)* (pp. 348-351). IEEE.

[34] Short History of Early Search Engines, The History of SEO. Haettu 30.3.2020 osoitteesta http://www.thehistoryofseo.com/The-Industry/Short_History_of_Early_Search_Engines.aspx

[35] Spiders, Scrapy Documentation. Saatavissa (viitattu 20.4.2020): https://docs.scrapy.org/en/latest/topics/spiders.html

[36] Sun, Y. (2008). A comprehensive study of the regulation and behavior of web crawlers.

[37] Total number of Websites, Internet Live Stats. Saatavissa (viitattu 31.3.2020): https://www.internetlivestats.com/

[38] Yleiskatsaus Googlen indeksointiroboteista (käyttäjäagenteista), Google. Saatavissa (viitattu 30.3.2020): https://support.google.com/webmasters/answer/1061943?hl=fi