LAB University of Applied Sciences

Chatbot-pilotti tekoälyn avulla

LAB-ammattikorkeakoulu
Insinööri (AMK), Tieto- ja viestintätekniikka (ohjelmistotekniikka)
2021
Joni Tammilehto

Tiivistelmä

Tekijä(t)	Julkaisun laji	Valmistumisaika
Tammilehto, Joni	Opinnäytetyö, AMK	2021
	Sivumäärä 25	

Työn nimi

Chatbot-pilotti tekoälyn avulla

Tutkinto

Insinööri (AMK), tieto- ja viestintätekniikka

Toimeksiantajan nimi, titteli ja organisaatio

Axel Tuomala, toimitusjohtaja, Tikion Oy

Tiivistelmä

Opinnäytetyön tavoitteena oli tehdä chatbot-pilotti käyttäen Facebookin omistamaa Wit.ai tekoälypalvelua. Toimeksiantajana toimi perustamisvaiheessa oleva yritys Tikion Oy.

Koneoppiminen on tekoälyn osa-alue, jonka tarkoituksena on kehittää ohjelmiston toimintaa sen saaman datan perusteella. Koneoppimisen prosessiin kuuluu datan kerääminen, sanitointi sekä organisointi. Sanitoitu data jaetaan opetus- ja testidataksi. Opetusalgoritmin valinnan jälkeen aloitetaan opetus opetusdatalla, ja lopuksi algoritmia testataan testidatalla ja tehdään mahdollisia parannuksia.

Big data on suuri datakokoelma, jota voidaan kerätä käytännössä mistä tahansa lähteestä. Big datan analysoinnissa keskitytään poimimaan datakokoelmasta kaikki relevantti informaatio. Big dataa hyödynnetään nykypäivänä paljon muun muassa yritysten toiminnassa.

Chatbot-pilotti kehitettiin käyttäen webbiteknologioita, sekä tekoälypuoli toteutettiin Wit.ai:lla. Pilotti pystyy vastaanottamaan käyttäjän antamia lauseita, ja antamaan JSON-muotoisen vastauksen.

Lopputuloksena saatiin chatbot-pilotti. Käyttäjän kysymyksen perusteella chatbot vastaa käyttäjälle testi-intentioiden määritysten mukaisesti.

Asiasanat

Chatbot, Wit.ai, tekoäly, tekstianalyysi, ai, big data

Abstract

Author(s)	Type of Publication	Published
Tammilehto, Joni	Thesis, UAS	2021
	Number of Pages	
	25	

Title of Publication

Chatbot-pilot with Ai

Name of Degree

Engineer (UAS), Information and Communication Technologies

Name, title and organization of the client

Axel Tuomala, CEO, Tikion Oy

Abstract

The aim of the thesis was to make chatbot-pilot using Facebooks Wit.ai service. The client was Tikion Oy, company that is still in the founding phase.

Machine learning is a subset of artificial intelligence that aims to develop operations of software based on the data it receives. Process of machine learning is to get data, then cleaning and organizing that data. The sanitized data is split to training dataset and test dataset. After choosing learning algorithm, the training is started and finally the algorithm is tested with test data and possible improvements are made if necessary.

Big data is large data set, which can be collected from anywhere. Focus of big data analysis is to take all relevant information from the data set. Companies uses big data analytics a lot nowadays.

The chatbot-pilot was developed using web technologies, and artificial intelligence side was implemented with Wit.ai. The pilot is able to receive the user input and to give JSON response.

Result of the thesis was chatbot-pilot. Pilot is capable of taking user input sentences and responding according to test intentions.

Keywords

Chatbot, Wit.ai, artificial intelligence, text analysis, ai, big data

Sisällys

1 Johdanto

Tekoälyä käytetään enenevässä määrin analysoimaan kaikenlaista tekstipohjaista dataa. Tähän kuuluu suurista datakokoelmista, kuten big datasta, saatava informaatio, sekä chatbottien luonnollisen kielen käsittelyyn pohjautuvat toiminnallisuudet. Chatbotit ovat paljon käytössä asiakaspalvelun tukena sivustoilla. Chatbotit keventävät asiakaspalvelijoiden työtaakka ja helpottaa käyttäjien informaation etsintää sivustoilla. Ne auttavat käyttäjiä yleisimmissä kysymyksissä ja voivat kysyä käyttäjältä tarvittavia tietoja esimerkiksi lomakkeiden tai ilmoituksien tekemiseen. Chatbotit poimivat käyttäjän syötteestä tarpeelliset parametrit ja antaa näiden pohjalta vastauksen.

Toimeksiantaja on Tikion Oy, perustamisvaiheessa oleva yritys, jossa kirjoittaja on myös itse mukana. Yritys tulee tarjoamaan erilaisia digitaalisia palveluja, kuten muun muassa valmiita tuotepaketteja. Päätoimipaikkana on Lahti.

Opinnäytetyön tavoitteena on tehdä chatbot-pilotti käyttäen Wit.ai-tekoälyratkaisua. Pilotti pystyy vastaanottamaan käyttäjän antaman lauseen ja lähettämään JSON-muotoisen vastauksen takaisin perustuen käyttäjän syötteeseen. Työn ohessa tutustutaan pilotin mahdolliseen jatkokehitykseen. Tulevaisuudessa tämä tulisi olemaan valmis tuote, jota pystyttäisiin myymään sellaisenaan.

Tutkimuksen tavoitteena on tarkastella, kuinka tekoälyä käytetään hyväksi tekstintunnistuksessa ja analysoimisessa. Lisäksi käydään läpi, minkälaisia teknologioita tekstianalyysille on, sekä muutamia esimerkkejä mahdollisista tekstianalyysin käyttökohteista.

Teoriaosuudessa käydään läpi koneoppimista sekä tekoälyn hyödyntämistä ja käyttökohteita tekstianalyysissä. Lisäksi käydään läpi big dataa ja sen hyödyntämistä nykypäivänä. Digitalisaation ja datan keräämisen takia nykyään on olemassa suuri markkina datan analysoinnille sekä hyödyntämiselle eri yritysten toiminnassa.

2 Koneoppiminen

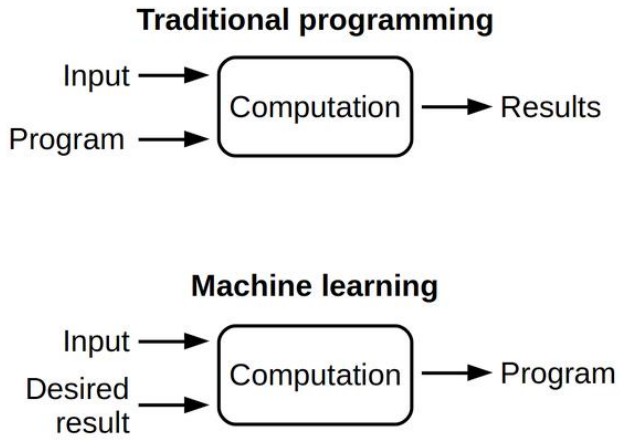
2.1 Yleistä koneoppimisesta

Koneoppiminen (Machine learning) on tekoälyn osa-alue, jonka tarkoitus on kehittää ohjelmistoa toimimaan paremmin ja tarkemmin käyttäjän antamien tietojen ja käskyjen, sekä olemassa olevien pohjatietojen avulla (Lehto ym. 2018, 13). Koneoppiminen on hyvä ratkaisu vaikeammille tehtäville, mihin ihmisen on hankala luoda suoraan toimivaa algoritmia ongelmanratkaisuun (Koleva 2020).

Esimerkiksi tekoäly voi osata tulkita käyttäjän tuottamaa tekstiä, tunnistaa siitä mahdolliset kirjoitusvirheet ja osata tunnistaa sanoille synonyymejä sekä kirjoitusmuotoja. Tekoälyä sovelletaan muun muassa datan analysointiin ja louhintaan, hahmojen tunnistukseen, ja itseohjautuviin järjestelmiin. Yksinkertaisimpia koneoppimisesimerkkejä ovat sähköpostin roskapostisuodatus, verkkokauppojen tuotesuositukset aikaisempien ostojen sekä tuoteselailujen pohjalta, sekä musiikki- ja videopalvelujen suositukset. Google tekee myös paljon erilaisia ehdotuksia internet-käytön perusteella sekä kohdennettua mainontaa.

2.2 Opettaminen

Ohjelmiston opettaminen alkaa siitä, että kerätään dataa, jota voidaan käyttää opetuksessa sekä testauksessa. Tämän jälkeen data sanitoidaan ja organisoidaan, jos on tarvetta, sekä data jaetaan opetusdataksi ja testidataksi. Seuraavaksi valitaan soveltuva opetusalgoritmi, ja aloitetaan opetus. Opetusvaiheessa ohjelmistolle annetaan opetusdataa, sekä kerrotaan haluttu lopputulos, jonka avulla ohjelmisto rakentaa logiikan, millä päästään haluttuun lopputulokseen. Tämä eroaa perinteisestä ohjelmoinnista siten, että perinteisessä ohjelmoinnissa ohjelmistolle määritellään logiikka, jolla saadaan haluttu vastaus. Tämä on esitetty alla olevassa kuvassa 1. Tämän pohjalta ohjelmisto tekee säännöt ongelman ratkaisemiseksi. (Googlen Machine Learning 2018.)



Kuva 1. Perinteinen ohjelmointi vs. koneoppiminen (Futurice 2018)

Lopuksi algoritmia testataan, sekä tehdään mahdollisia parannuksia, kuten esimerkiksi säädetään parametreja (Googlen Machine Learning 2018). Kuvassa 2 on esitetty tyypillinen koneoppimisen opetuksen työnkulku.

Kuva 2. Tyypillinen koneoppimisen työnkulku (Googlen Machine Learning 2018)

2.3 Opetusdata

Opetusdatalla tarkoitetaan dataa, jonka avulla tekoälyä opetetaan ongelmanratkaisuun. Opetusdatan on oltava samankaltaista dataa, mitä tekoäly käy läpi tulevaisuudessakin. Esimerkiksi jos tekoälyn halutaan suodattavan sähköposteista roskapostit pois, opetusdatan on oltava sähköpostiviestejä, jotka sisältävät myös roskapostia. Opetusdata pitää myös olla luokiteltu ja merkitty joissain opetusmalleissa, esimerkiksi itseohjautuville autoille ei riitä vain kuvat autoteistä, vaan kuviin täytyy merkitä missä kuvasta löytyy jalankulkijat, autot ja liikennemerkit. (Appen 2020.) Opetusdatan määrä riippuu kohteesta, jos opetettavat algoritmit ovat todella komplekseja ja lopputuloksen pitää olla todella tarkka, opetusdataa täytyy olla valtavasti. (Christianson 2020). Yleinen ohjeistus on, että opetusdataa tarvitaan enemmän mitä luulisi.

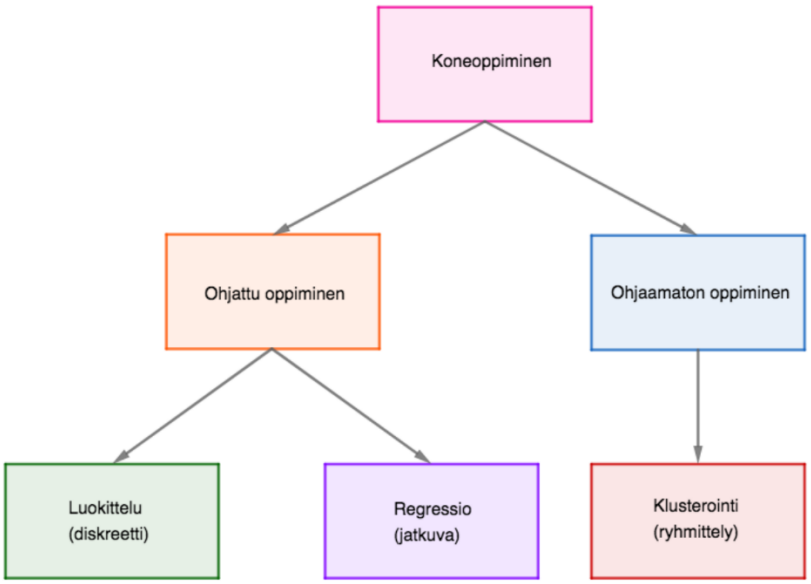
Testidatalla voidaan varmistaa, onko opetusmenetelmä toiminut. Testidata on opetusdatan kaltaista mutta täysin riippumatonta opetusdatasta. Esimerkiksi roskapostisuodattimen

testidatana käytetään sähköpostiviestejä. Nämä sähköpostiviestit eivät voi kuitenkaan olla samoja, mitä on käytetty suodattimen opettamisessa.

2.4 Opetusalgoritmit

Opetusalgoritmit luokitellaan käytettävän opetusdatan mukaan. Nämä voidaan jakaa kuvan 3 esittämällä tavalla kolmeen eri luokkaan oppimisen tyylin perusteella: ohjattu oppiminen, vahvistusoppiminen sekä ohjaamaton oppiminen (Lehto ym. 2018, 13).

Mallien rakentamisessa pitää ottaa huomioon ali- ja ylioppiminen. Alioppiminen tarkoittaa sitä, että opetusmenetelmä on liian yksinkertainen, jolloin ohjelmiston kyky ennustaa ja jakaa dataa luokkiin on heikko. Ylioppiminen on taas sitä, että malli kuvaa liian tarkasti opetusjoukon informaatiota, ja jakaa dataa liian tarkasti omiin luokkiinsa.



Kuva 3. Koneoppimisen luokittelua (Lehto ym. 2018, 14)

2.4.1 Ohjattu oppiminen

Ohjatussa oppimisessa (Supervised learning) annettu opetusdata koostetaan valmiista aineistoista, joista tiedetään haluttu lopputulos. Käytettävä data on ennalta jaettu alkioihin, jotka on luokiteltu valmiiksi. Tämän jälkeen algoritmille annetaan testidata, jonka perusteella voidaan todeta, kuinka hyvin algoritmi on onnistunut. Ohjattu oppiminen voidaan jakaa joko luokitteluun tai regressioon, riippuen käytetyn datan tyypistä. Jos sisään tuleva data pystytään luokittelemaan erillisiin ryhmiin, niin kyse on luokittelusta, jos dataa tulee jatkuvasti, niin kyse on regressiosta. (Lehto ym. 2018, 13.)

2.4.2 Ohjaamaton oppiminen

Ohjaamattomassa oppimisessa (Unsupervised learning) annetusta opetusdatasta ei tiedetä mitään ennalta, vaan ohjelmisto klusteroi alkiot luokkiin riippuen siitä, kuinka paljon ne muistuttavat toisiaan. Klusterointi tarkoittaa datan alkioiden jakamista luokkiin niiden samankaltaisuuksien perusteella. Tämän jälkeen käyttäjä luokittelee klusterit, tästä saadaan luokittelu algoritmille. Esimerkkinä klusteroinnista on asiakasdatasta samankaltaisten asiakkaiden luokittelu samaan ryhmään. Tavoitteena on jakaa data ryhmiin, joiden sisäinen samankaltaisuus on suuri ja ryhmien välinen samankaltaisuus on mahdollisimman pieni. Oikeita valmiiksi määriteltyjä luokkia ei ole. (Lehto ym. 2018, 13.)

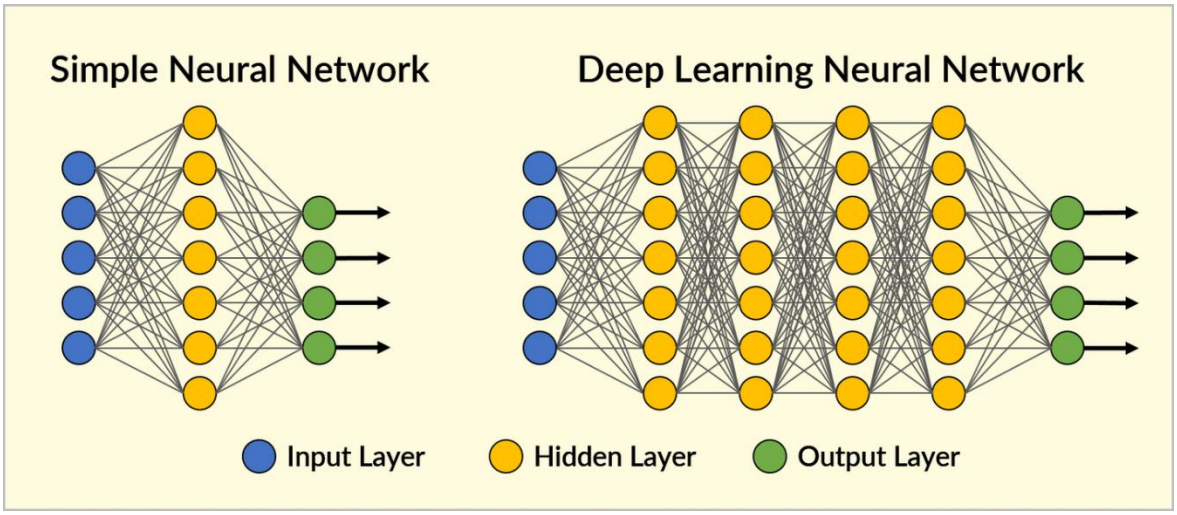
2.4.3 Vahvistusoppiminen

Vahvistusoppiminen (Reinforcement learning) eroaa ohjatusta oppimisesta siten, että kone ei saa valmiita tavoitteita, sekä alkioita ei ole luokiteltu valmiiksi, vaan koneelle annetaan positiivista tai negatiivista palautetta riippuen siitä, miten tarkasti alkioiden luokittelu on onnistunut. Kone pyrkii optimoimaan tekemisensä kohti positiivista palautetta. Esimerkkejä vahvistetusta oppimisesta ovat itseohjautuvat autot ja robotiikka. (Lehto ym. 2018, 13-14.)

2.5 Syväoppiminen ja neuroverkot

Syväoppimisessa (Deep Learning) on tavoitteena luoda algoritmien avulla neuroverkko, joka pystyy ratkaisemaan sille syötetyt ongelmat (Elements of AI). Syväoppimista käytetään sellaisten ongelmien ratkaisemisessa, joissa perinteisemmillä algoritmeilla täytyisi tehdä erittäin monimutkaisia ja hankalia sääntöjä ongelman ratkaisemiseksi. Esimerkiksi puheen, kuvien ja tekstien tunnistamisessa ja käsittelyssä käytetään syväoppimista.

Neuroverkot koostuvat keinotekoisista ”hermoista”, eli neuroneista. Neuronit laskevat numeerisen tuloksen yhdestä tai useammasta numeerisesta arvosta, jotka tulevat joko syötetystä datasta tai muilta neuroneilta, ja antavat sille sisäisen painoarvon. Dataa käsitellään ”yksiköittäin”, joista muodostetaan kerroksia ja niistä verkostoja. Verkostojen syvyys mahdollistaa monimutkaisten sääntöjen oppimisen syötetyn datan pohjalta. Havaintokuva neuroverkoista kuvassa 4. (Elements of AI.)



Kuva 4. Neuroverkot (Bernand 2019)

2.6 Luonnollisen kielen käsittely

Luonnollisen kielen käsittely, eli NLP (Natural Language Processing), on koneoppimisen osa-alue, jossa käytetään tietokoneohjelmistoja sekä tekoälyä luonnollisen puheen sekä tekstin tulkitsemiseen, analysointiin ja tuottamiseen. Asiakaspalvelussa käytetään usein chatbotteja, jotka ovat niin sanottuja ”virtuaalisia asiakaspalvelijoita”. Ne pyrkivät vastaamaan käyttäjän yleisimpiin kysymyksiin automatisoidusti. Luonnollisen kielen käsittely alana sisältää konekääntämisen, automaattisen puheentunnistuksen, puhesynteesin, tekstintunnistuksen, älykkään tekstinsyötön ja puheen kääntämisen osa-alueet. (SAS 2019.)

Yhtenä esimerkkinä missä luonnollisen kielen käsittelyä pystytään hyödyntämään, on automatisoidussa esseiden pisteytyksessä (Automated essay scoring, AES). AES-ohjelmisto toimii poimimalla ominaisuuksia, kuten sanamäärä, sanaston valinta, virheiden määrä ja tiheys, lauseiden pituus sekä kappalerakenteet, joiden perusteella kirjoitus voidaan pisteyttää. Lisäksi AES-ohjelmistot käyttävät nykyisin luonnollisen kielen käsittelyä ja lingvististä tekstianalyysia esseiden sisällön tulkitsemiseen. Tämä tekee esseiden läpikäynnistä nopeaa, sillä tietokone pystyy arvioimaan tekstin muutamassa sekunnissa. Lisäksi AES arvioi esseet objektiivisesti, eli saman kaltaiset tekstit saavat saman arvosanan, eikä oteta huomioon deadlineja taikka kuka esseen on kirjoittanut. (Walker 2020.)

Kritiikkinä tällaisessa menetelmässä on se, että luovuutta ja omaperäisyyttä tekstissä ei oteta samalla tavalla huomioon. Lisäksi opiskelija voi kirjoittaa tekstiä, jossa on paljon tiedesanastoa ja kompleksista tekstiä, jotta algoritmi antaa paremman arvosanan. Ihmisten olisi hyvä tarkistaa AES-ohjelmistojen antamat arvosanat, koska näin varmistutaan tekstin sisällöstä ja voidaan ottaa mahdollisesti luovuutta huomioon arvioidessa. Tällöin saadaan karsittua pois suurin osa ongelmista ja silti kevennettyä työtaakkaa esseiden arvioinnissa.

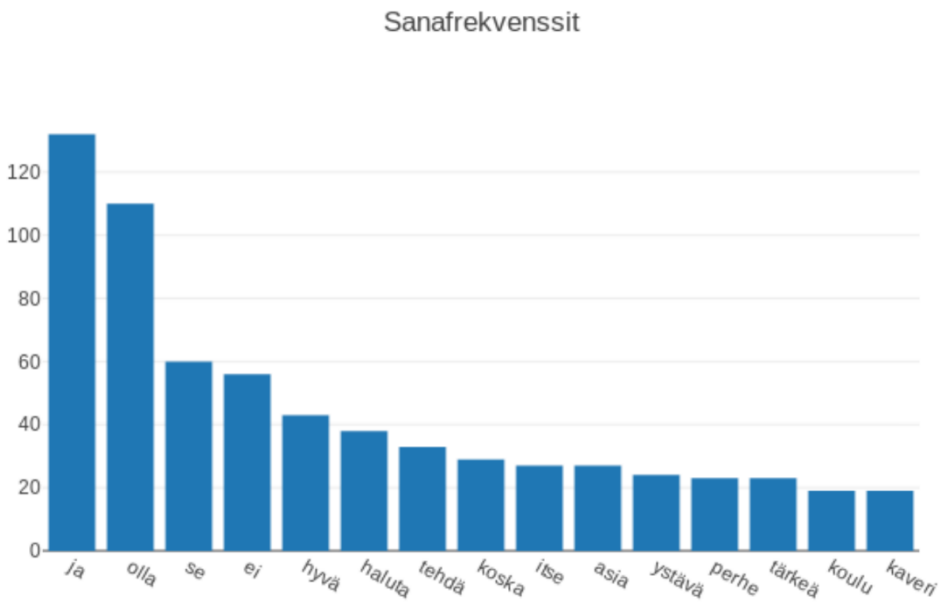
3 Tekstianalyysi

3.1 Tilastollinen ja lingvistinen tekstianalyysi

Tekstianalyysi voidaan perinteisesti jakaa kahteen kategoriaan, tilastolliseen ja lingvistiseen. Tilastollisessa tekstianalyysissa tekstiä analysoidaan helposti mitattavilla suureilla. Lingvistinen tekstianalyysi perustuu kielen rakenteiden tulkitsemiseen. (Lehto ym. 2018, 55.)

3.1.1 Tilastollinen tekstianalyysi

Tilastollinen tekstianalyysi perustuu tekstistä mitattaviin suureisiin, kuten sanojen määrään, sanojen keskipituuteen, esiintyvyyteen sekä sanaston monipuolisuuteen. Tämän takia tilastollinen analyysi ei vaadi esitietoja käsiteltävästä kielestä tai sen rakenteesta. Tilastollisesti voidaan tunnistaa tekstit aihepiireittäin selvittämällä, mitä sanastoa eri aihepiirit sisältävät ja miten ne jakautuvat tekstissä. Kun aihepiirin tekstejä on tulkittu riittävästi ja niistä on pystytty muodostamaan kieliprofiili, pystytään uusia tekstejä vertailemaan tähän ja päättelemään, koskeeko teksti olemassa olevia profiloituja aiheita. Vertailuunkin on paljon erilaisia menetelmiä. Varsinainen avainsanojen esiintyvyys ja frekvenssilistojen vertailu ei ole suuressa mittakaavassa kovin käytännöllinen, vaan tekstien kootuille ominaisuusvektoreille on kehitetty matemaattisia vertailumenetelmiä. (Lehto ym. 2018, 55.)



Kuva 5. Esimerkki frekvenssijakaumasta

Frekvenssijakaumalla voidaan luokitella tekstit sanaston perusteella aihepiireihin. Tilastollisessa menetelmässä on haasteensa, sillä monissa kielissä suurin osa sanoista toistuu

teksteissä aihepiiristä riippumatta. Esimerkiksi suomen kielessä sanat, kuten ”olla”, ”ja”, ”kuin”, ”että”, ovat yleisiä ja esiintyvät lähes poikkeuksetta frekvenssilistojen kärjessä (kuva 5). Ratkaisuna tähän voidaan käyttää sanojen numeerisia ominaisuuksia, eli voidaan jättää pois sanat, jotka ovat liian lyhyitä tai jotka eivät sisällä isoja alkukirjaimia. Lisäksi pystytään käyttämään poistosanatietokantaa, johon voidaan kasata sanoja, jotka sisältävät itsessään hyvin vähän informaatiota ja joita ei haluta käsitellä tilastollisesti. Tämä nopeuttaa tekstin käsittelyä ja säästää tilaa, jos tekstistä ensin leikataan pois kaikki tietokannasta löytyvät sanat. (Lehto ym. 2018, 55.)

Painottaminen frekvenssin suhteen voidaan tehdä yksinkertaisesti raakafrekvenssillä, eli laskemalla sen uniikkien sanojen esiintymismäärät. Hienostuneempi ja erittäin yleinen tekniikka on TF-IDF (Term Frequency - Inverse Document Frequency), joka perustuu numeeriseen arvoon. Arvo kertoo, kuinka tärkeä sana on aihepiirin kokoelman dokumenteissa ja näin saadaan kokoelmakohtainen arvo jokaiselle sanalla. Näiden avulla voidaan luoda numeerinen frekvenssiprofiili aihepiirin dokumenteille. (Lehto ym. 2018, 56.)

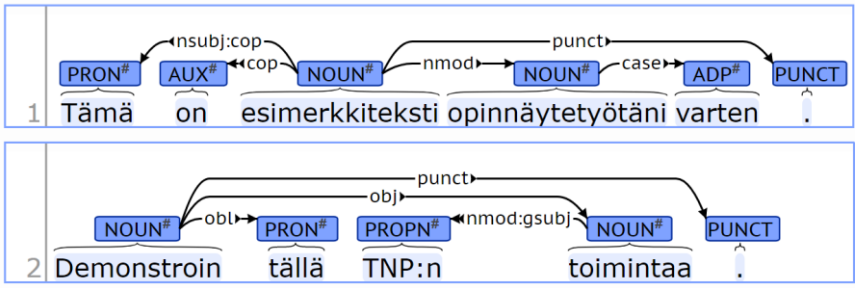
Jos frekvenssien vertaileminen ei tuota haluttua tulosta, voidaan sitä parantaa käyttämällä esimerkiksi kielimalleja, joilla saadaan kielen rakenneosien väliset suhteet muutettua tarkasteltaviksi suureiksi. Kielimallit kertovat esimerkiksi kuinka todennäköisesti tietyt sanat ja lauseet esiintyvät peräkkäin, sekä millaiset rakenneosat rakentavat yleisiä isompia kokonaisuuksia kielen sisällä. (Lehto ym. 2018, 56-57.)

3.1.2 Lingvistinen tekstianalyysi

Lingvistinen, eli semanttinen tekstianalyysi perustuu puhtaasti esitietoon käsiteltävästä kielestä. Suomen kielessä tämä tarkoittaa muun muassa sanojen päätteitä ja taivutusmuotoja, sekä miten nämä liittävät sanoja virkkeen tai lauseen sisällä toisiinsa. Lingvistinen lähestymistapa vaatii kielen merkitysten tunnistamista. Esitietoja kielestä sekä kielen eri sääntöjä voidaan käyttää laskennallisesti hyödyksi. Tällöin pystytään tulkitsemaan, mitä lause tarkoittaa ja pystytään päättelemään yksittäisen sanan merkitys riippuen missä tilanteessa sanaa käytetään, eli voidaan yhdistää samalla tavalla käyttäytyviä ja samoissa konteksteissa olevia sanoja toisiinsa. (Lehto ym, 2018, 55.) Esimerkkinä työkaluista, joita voidaan käyttää semanttisessa analyysissa ovat FinnWordNet, joka on kokoelma eri suomen kielen sanoista ja niiden semanttisista verkoista (Kielipankki 2020). FinnWordNetistä löytyy myös edellä mainittuja poistosanalistatietokantoja suomen kielen sanoista.

FDP (Finnish-dep-parser) on Turun yliopiston ensimmäinen työkalu suomenkielisen tekstin rakenteiden löytämiseen. TNP (Turku Neural Parser) on toinen Turun yliopiston työkalu, joka tekee käytännössä samaa asiaa, mutta on työkaluna kehittyneempi. FDP ja TNP

osaavat esitietojen pohjalta kertoa käyttäjälle sanojen semanttiset riippuvuudet, lauseenjäsenet ja taivutukset valmiiksi, minkä jälkeen niitä voidaan hyödyntää helposti. TNP on käytettävissä yli 50 kielellä (TurkuNLP 2020). Kuvassa 6 on esimerkki TNP:n toiminnasta. Vastaavaa tietoa ei pystytä kertomaan kielestä ilman esitietoja kielen rakenteesta. (Lehto ym. 2018, 56). CoNLL-18 listaus vertailee vastaavia työkaluja keskenään, käyttäen samaa opetus- ja testidataa. TNP on CoNLL-18 Shared Task – tilastoilla kärkipäässä. (CoNLL 2018 Shared Task 2018).



Kuva 6. TNP esimerkki

3.2 Tekstintunnistus teknologiat (OCR ja ICR)

Tekstin tunnistukseen käytettävät teknologiat ovat Optical character recognition (OCR), jolla tunnistetaan koneellisesti tuotettua tekstiä, sekä Intelligent character recognition (ICR), jolla voidaan tunnistaa käsin kirjoitettua tekstiä (Handwriting recognition, HWR). Tekstintunnistus teknologioita voidaan käyttää koneellisesti tai käsin kirjoitetun tekstin tulkitsemiseen.

3.2.1 Toimintaperiaate

Tekstintunnistus teknologioiden toimintaperiaate yksinkertaistetusti perustuu hahmon tunnistukseen, eli skannatusta asiakirjasta verrataan merkin pikseleistä muodostamaa hahmoa merkkeihin ja pyritään tunnistamaan oikea kirjain, numero tai erikoismerkki. Näin teksti voidaan digitalisoida ja muuttaa konelukuiseksi, jolloin siitä pystytään hakea informaatiota sekä tekstiä voidaan muokata helposti. (Rivera 2019.)

OCR perustuu kirjain kerrallaan tekniikkaan, kun taas ICR tulkitsee kokonaisia sanoja. Tämä pienentää yksittäisten kirjainten väärinymmärryksen riskiä, koska on paljon haastavampaa tulkita kaunokirjoituksesta mihin kirjain päättyy ja toinen alkaa. ICR tekniikalla voidaan siis tulkita käsin kirjoitettua tekstiä muun muassa erilaisista paperisista dokumenteista, valokuvista sekä kosketusnäytölle kirjoitetusta tekstistä. (Pay 2015.)

Tekstintunnistusteknologioita hyödynnetään muun muassa, jos asiakirjoja pyritään automatisoidusti indeksoimaan tai hakemaan ja lukemaan niistä löytyvää informaatiota.

Esimerkkeinä tällaisista asiakirjoista on sanomalehdet, laskut, asiakaskyselylomakkeet, markkinatutkimukset, tilaukset tms. mitkä pyritään tallentamaan helposti hallittavaan muotoon. Lisäksi OCR-teknologiaa käyttävät myös jotkin spammibotit, jotka lähettävät roskapostia tai kirjoittelevat joihinkin keskustelupalstoille, sillä OCR:n avulla ne voivat läpäistä CAPTCHA-testit (F-Secure Labs 2019).

OCR-teknologiaa käytetään myös automaattisessa rekisterikilpien tunnistuksessa. Suomeen kilvenlukulaitteisto tuli testikäyttöön vuonna 2014, ja vuoteen 2020 mennessä laitteisto oli käytössä jokaisessa liikenne- ja valvontasektorin poliisiautossa. Lukukamera ilmoittaa muun muassa katsastamattomista autoista, sekä hätäkeskuksesta voidaan ilmoittaa tietoihin esimerkiksi mahdollisesta rattijuoposta tai polttoainevarkaasta, jolloin laite hälyttää, jos kyseinen auto ajaa poliisiautoa vastaan. Suomessa käytössä oleva REVIKA-laitteisto tunnistaa 30 eri maan rekisterikilvet. (Ziemann 2017.)

3.2.2 Tarkkuus

Tarkkuusmielessä ongelmana on yksittäisten merkkien sekaantuminen keskenään, esimerkiksi iso I-kirjain, pieni l-kirjain sekä !-merkki tai numero 8 ja kirjain S voivat sekoittua keskenään (ISMP 2014). Näissä tilanteissa voidaan sanoja ja merkkejä tarkastaa kieliopin, sekä kontekstin avulla. Tunnistusta voidaan myös tarkentaa, jos luettua informaatiota voidaan tarkistaa jollain muulla tavalla. Esimerkkinä on henkilötunnus, passin numero, ajokortin numero, pankkitilin numero tai pankkiviitteen numero tai jokin muu numeerinen arvo, jonka pystyy tarkistamaan esimerkiksi jollain matemaattisella algoritmilla. Myös esimerkiksi viivakoodien lukeminen kameralla voidaan luokitella tähän. Käsialan tunnistuksessa on haasteena tekstin huonolaatuisuus ja erottuvuus taustasta, hyvin uniikit käsialat eri ihmisillä, epälineaariset merkkien paikat ja koot, sekä merkkien erottelu.

Tunnistamisen laatuun vaikuttaa myös skannattujen aineistojen kuvan laatu ja tekstin lukutarkkuus. Kontrasti on yksi oleellinen asia, joka vaikuttaa tekstin tunnistamiseen, eli jos tausta on valkoinen ja teksti itsessään on selkeää ja teräväreunaista, on tunnistaminen helpompaa. Myös skannatun aineiston suoristaminen ja tekstin samansuuntaisuus vaikuttaa lopputuloksen tarkkuuteen. Resoluutio on toinen suuri tekijä ja yleensä optimaalisin lukutarkkuus merkintunnistuksessa on 300 dpi (Dots Per Inch), vaikkakin viivakoodeja lukiessa 200 dpi resoluutiota pidetään riittävänä. Liian suuri resoluutio tuo kuvaan kohinaa, eli taustan värien pisteitä, jotka pehmentävät tekstin reunoja ja tekevät siitä vaikealukuisempaa. Lisäksi huonoon laatuun voi olla syynä käyttäjän kirjoitusvirheet tai ylikirjoitukset, sekä korjaukset alkuperäisessä dokumentissa. (Saraf 2020.)

Vuonna 2009 Australiassa tehdyssä tutkimuksessa digitalisoitiin vanhoja sanomalehtiä 1800-luvulta 1900-luvun puoleenväliin. Tutkimuksessa todettiin sen hetkisten OCR-ohjelmistojen tarkkuuden olevan välillä 71 % - 98.02 %. Tarkkuusmittarina käytettiin yksittäisten merkkien tunnistusta. Yleisesti 98 % - 99 % tarkkuus on hyvä OCR-ohjelmistolle, 90 % - 98 % on kohtalainen ja alle 90 % on huono. Jos tarkkuus on vain 71 %, virheellisiä merkkejä keskiverrossa 500 merkin kappaleessa on 145. Tämä tekee tekstin lukemisesta sekä informaation hakemisesta paikoittain erittäin haastavaa. (Holley 2009.)

4 Big data tekstianalyysissa

4.1 Yleistä big datasta

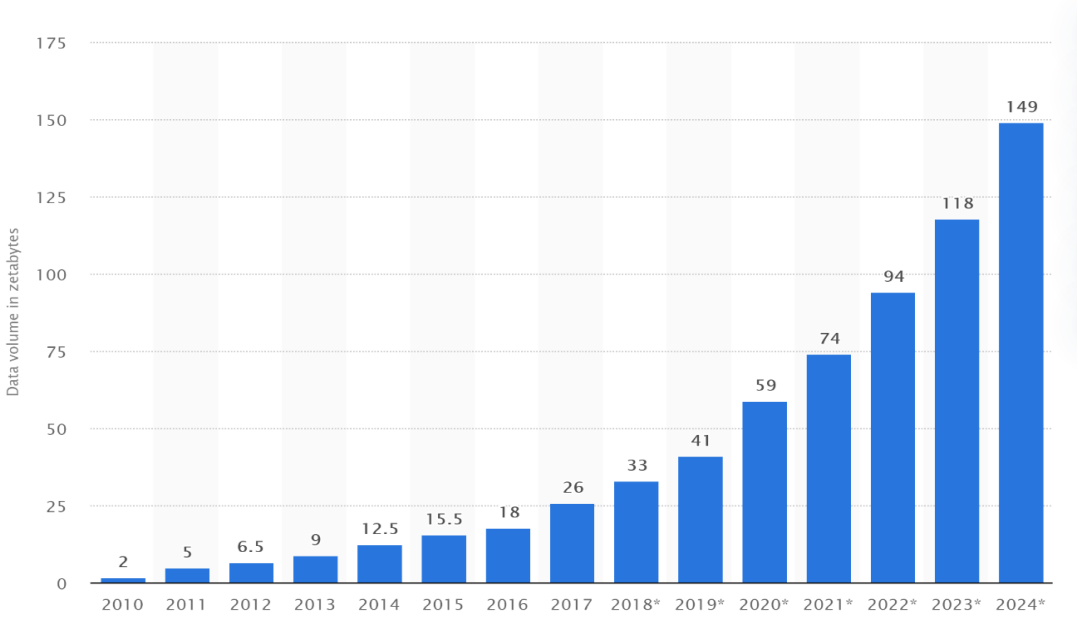
Big data (massadata) on yksinkertaisesti suuri kokoelma dataa. Sen analysointia on poimia tärkeä informaatio suurista datakokoelmista, jotka ovat liian isoja ja komplekseja käsiteltäväksi perinteisillä menetelmillä. Big datalla on seitsemän pääominaisuutta, eli ”seitsemän V:tä”. Nämä ovat suuruus (volume), monimuotoisuus (variety), nopeus (velocity), vaihtelevuus (variability), todenmukaisuus (veracity), visualisointi (visualization) sekä arvo (value) (Curtis 2020).

Suuruus eli datan määrä kertoo, että dataa on paljon. Tosin big datan minimikokoluokan raja on hyvinkin häilyvä, ja mitään oikeaa selkeää rajaa siitä, milloin datakokoelmaa voidaan pitää big datana ei ole.

Monimuotoisuudella tarkoitetaan minkälaista dataa big data sisältää. Nykyisin datakokoelmissa voi olla kaikkea taulukoista ja tietokannoista video- ja äänitiedostoihin. Tällainen monimuotoisuus aiheuttaa omat haasteensa datan analysoinnissa ja tallentamisessa.

Nopeus tarkoittaa tässä kontekstissa sitä, kuinka nopeasti dataa kertyy ja kuinka nopeasti sitä voidaan prosessoida. Dataa kertyy globaalista valtavia määriä jatkuvasti. Hyvänä esimerkkinä big datan keräyksestä on muun muassa osakemarkkinat, jossa New York Stock Exchange luo arviolta 1 TB (teratavu) dataa uudesta osakeliikenteestä päivittäin. Lisäksi sosiaalisen median kanavissa kertyy valtava määrä dataa päivittäin, Facebookissa kertyy uutta dataa yli 4 PB (petatavua) päivittäin (Wiener 2014).

Globaalisti dataa on kertynyt vuoteen 2020 mennessä 59 ZB (tsettatavu) (Statista 2020) ja yksi tsettatavu on miljoona teratavua. Kuvassa 7 on tilasto globaalista datakertymästä vuodesta 2010–2024, joista vuodet 2021–2024 ovat arvioita.



Kuva 7. Globaali datakertymä. (Statista 2020)

Vaihtelevuus viittaa datan epäjohdonmukaisuuteen. Datan merkitys voi muuttua, varsinkin kielen käsittelyssä sanojen merkitykset saattavat muuttua ja kehittyä, tai sanoille voi tulla uusia merkityksiä.

Todenmukaisuus on tärkeää, sillä jos data ei ole tarkkaa ja todenmukaista, se on arvotonta. Tämä on varsinkin silloin erittäin tärkeää, jos dataa syötetään ohjaamattomalla oppimisella opetettavalle tekoälyalgoritmille.

Visualisointi mahdollistaa sen, että suurestakin datamäärästä saadaan ymmärrettävää. On paljon helpompi hahmottaa dataa ja sen sisältämää informaatiota, jos se pystytään visualisoimaan.

Big datan arvo on myös suuressa osassa big datan analyysissa. Suuren datamäärän analysointi on turhaa, jos sitä ei pystytä hyödyntämään. (McNulty 2014.)

4.2 Big datan historia

Ensimmäinen suuri datankeruu projekti luotiin vuonna 1937, kun Yhdysvaltojen presidentti Franklin D. Rooseveltin hallinto laittoi vireille sosiaaliturva lain (Social Security Act, SSA). Valtion piti pystyä pitämään kirjaa yli 26 miljoonasta amerikkalaisesta ja yli 3 miljoonasta työntekijästä. IBM kehitti tätä varten koneen reikäkorttien lukuun.

Yksi ensimmäisistä datan prosessointiin tarkoitetuista koneista tehtiin vuonna 1943. Se oli englantilaisen matemaatikko Max Newmanin keksintö ja sitä käytettiin purkamaan natsien salakirjoitusta toisessa maailmansodassa. Laitteen ensimmäinen versio oli nimeltään Heath

Robinson. Tällä oli kuitenkin ongelmana siinä käytetyt paperiset nauhat, jotka venyivät, kun niitä pyöritettiin kovilla nopeuksilla. Tämä sai nauhat menemään eri tahtiin verrattuna toisiinsa, jolloin laite ei enää toiminut oikein. Robinsonia oli hidas käyttää ja se ei ollut luotettava, jos sitä käytettiin edes hieman suuremmalla nopeudella. Laiteelle kehitettiin jatkoa, joka oli nimeltään Colossus. Se etsi yhtäläisyyksiä viesteistä 5000 kirjainta per sekunti vauhdilla, joka on viisi kertaa nopeampaa kuin mihin Robinson pystyi. Tällä pystyttiin tekemään viikkojen työn muutamaan tuntiin. Colossus käytti vain yhtä paperista nauhaa, jolla koneelle annettiin purettava teksti, sillä loput pystyttiin hoitamaan sähköisesti. (Stanford 2008.)

Vuonna 1944 Fremont Rider, Wesleyan yliopiston kirjastonhoitaja, kirjoitti kirjassaan “The scholar and The future of Research Library”, että arviolta 16 vuoden välein Amerikan yliopistojen kirjastojen koko kaksinkertaistuu. Tällä tahdilla Yale yliopiston kirjastossa olisi vuonna 2040 noin 200 000 000 kirjaa, jotka veisivät 6000 mailia (9656 km) hyllytilaa. Toki Rider ei odottanut digitalisointia, mutta hän ennusti informaatiokasvun. (Press 2013.)

Vuonna 1980 sosiologi Charles Tilly oli ensimmäinen, kuka käytti termiä ”big data” artikkelissaan (Tilly 1980). Termiä ei kuitenkaan käytetty täysin samassa kontekstissa, miten me nykyisin ymmärretään big data.

1990 amerikkalainen tiedemies Peter Denning ymmärsi hyvin mitä on mahdollista tehdä big datalla. Hän kirjoitti artikkelissa:

> *It is possible to build machines that can recognize or predict patterns in data without understanding the meaning of the patterns. Such machines may eventually be fast enough to deal with large data streams in real time. (Denning 1990.)*

Michael Coxin ja David Ellsworthin artikkeli “Application-controlled demand paging for out-of-core visualization” vuodelta 1997 on ensimmäinen artikkeli ACM digitaalisessa kirjastossa (ACM digital library) jossa käytetään termiä big data seuraavasti:

> *Visualization provides an interesting challenge for computer systems: data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk. We call this the problem of big data. When data sets do not fit in main memory (in core), or when they do not fit even on local disk, the most common solution is to acquire more resources. (Cox, Ellsworth 1997.)*

Vuonna 1998 New York Times kertoi artikkelissaan, että John Mashey, SGI:n tutkimusjohtaja, olisi ensimmäinen kuka käytti termiä big data (Verhulst 2013). Vaikka Cox ja Ellsworth

olivat käyttäneen termiä jo vuotta aiemmin artikkelissaan, Mashley oli käyttänyt termiä myös useissa puheissaan. Tämän takia häntä pidetään termin keksijänä (Big Data Framework 2019a).

Tim O'Reilly julkaisi vuonna 2005 artikkelin "What is Web 2.0?", missä käytiin läpi datan tallentamisen tärkeyttä. Tämä aloitti laajemmin big datan analysoinnin. Roger Mougalas O'Reilly Medialta käytti termiä big data kuvaillakseen isoa joukkoa dataa, jota on lähes mahdotonta käsitellä perinteisillä työkaluilla, eli miten termiä käytetään nykyisin. Lisäksi sosiaaliset mediat alkoivat tekemään tuloaan ja dataa alkoi kertymään paljon enemmän kuin aikaisemmin. (O'Reilly 2005.)

Kuten huomataan varsinaisen termin alkuperä nykyisessä kontekstissa on hieman vaikeaselkoinen. Datan kerääminen suurissa määrin ja sen pohjalta analyysien tekeminen on ollut kuitenkin tavoitteena jo pidemmän aikaa. Nykyisin big data terminä tarkoittaa mitä tahansa suurta datajoukkoa, jota ei pystytä perinteisin menetelmin käsittelemään.

4.3 Datarakenteet

Datarakenteet auttavat datan tallentamisessa siten, että niihin pääsee käsiksi ja ne ovat muokattavissa helposti. Datarakenteet voidaan karkeasti jakaa kolmeen eri tyyppiin: jäsennelty data (structured data), jäsentämätön data (unstructured data) sekä osittain jäsennelty data (semi-structured data). (Big Data Framework 2019b.)

Jäsennelty data on järjestetty valmiiksi määritellyn datamallin mukaisesti, jolloin sitä on helppo analysoida. Datamalli määrittelee, miten dataa tallennetaan, datojen väliset suhteet ja miten niihin pääsee käsiksi. Tämän takia datan jokaiseen kenttään pääsee käsiksi erikseen tai muiden kenttien tietojen kautta. Tavallisimpia esimerkkejä jäsennellystä datasta ovat Excel-taulukot sekä SQL-tietokannat. Kaikilla näillä on numeroidut rivit ja sarakkeet, joita pystytään järjestelemään helposti. (Big Data Framework 2019b.)

Yhtenä esimerkkinä jäsennellyn datarakenteen käytöstä on Googlen hakukone. Nettisivuilla pystytään hyödyntämään pientä, yleensä JSON-LD muodossa olevaa datapakettia, jossa kerrotaan oleellista informaatiota sivusta. Esimerkkinä reseptisivulla voidaan kertoa ainesosat, valmistusaika, kalorit ja muuta vastaavaa informaatiota. Google käyttää myös dataa tehdäkseen erikoishakutuloksia, joita ovat kaikki pienet graafiset hakutulokset, kuten pienet kuvat ja videot, tai taulukot. (Google Search Central 2021.)

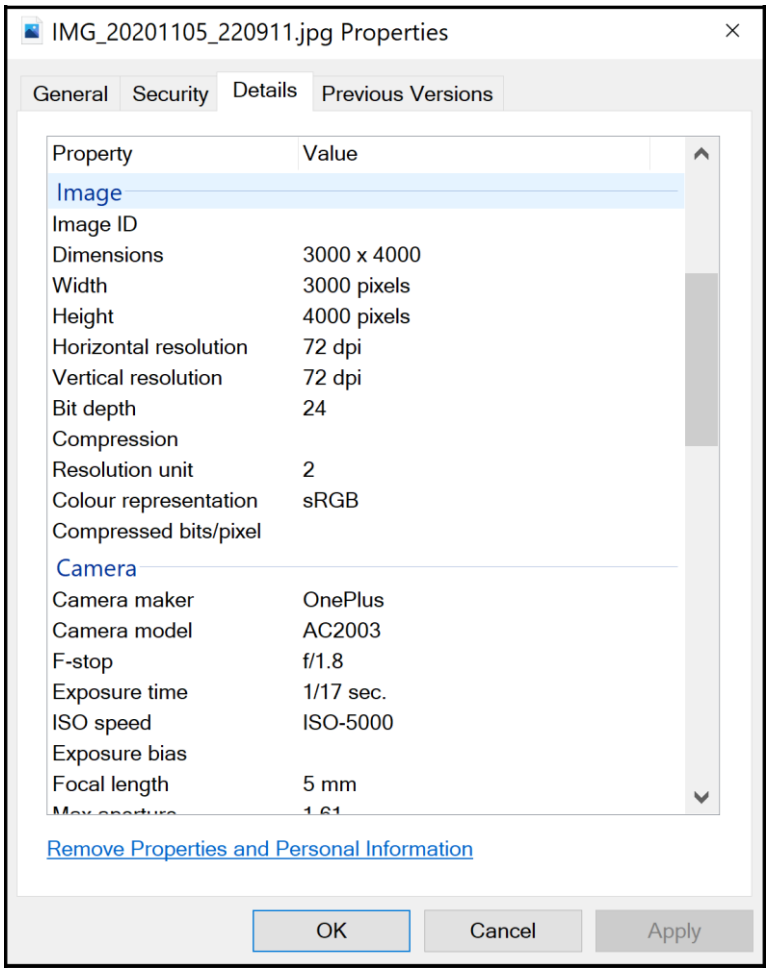
Jäsentämättömällä datalla ei ole datamallia tai sitä ei ole järjestetty millään tavalla. Se on tyypillisesti hyvin tekstipainotteista dataa, mutta siinä voi olla esimerkiksi päivämääriä, numeroita ja muuta informaatiota seassa. Jäsentämätöntä dataa sellaisenaan on vaikea

käsitellä perinteisillä ohjelmilla. Markkinoille on tullut teknologioita, joilla pystytään erikoistumaan erilaisiin datatyyppeihin. Esimerkiksi MongoDB on optimoitu tallentamaan erilaisia dokumentteja. Tyypilliset esimerkit jäsentämättömästä datasta ovat kokoelmat erityyppisiä tiedostoja tai No-SQL tietokannat. (Big Data Framework 2019b.)

Osittain jäsennetyllä datalla on datarakenne, mutta se ei käytä mitään datamallia vaikkakin omaavat merkintöjä ja hierarkioita kenttien välillä. Esimerkkejä näistä ovat JSON ja XML datatyypit. (Big Data Framework 2019b.)

Näiden lisäksi on olemassa metatietoa eli metadataa, joka on dataa datasta. Eli metadatasta saa lisäinformaatiota jostain datakokoelmasta, esimerkiksi tiedostosta. Metadata on tärkeässä osassa big data-analyysiratkaisuissa, sillä metadatan avulla voidaan esimerkiksi suodattaa dokumentteja jonkin metadatasta löytyvän attribuutin avulla. Kuvassa 8 näkyy esimerkki valokuvan metadatasta. Metadatasta löytyy informaatiota missä ja milloin valokuva on otettu, millä kameralla kuva on otettu ja mikä on kuvan resoluutio. Toinen esimerkki on Twitter-sosiaalisenmedian alustan viestit eli twiitit. Twiittien 280 merkin merkkirajasta huolimatta yhdestä twiitistä löytyy metadatasta muun muassa:

- Käyttäjänimi ja käyttäjä ID alkuperäisestä twiitistä mihin vastataan
- Luontipäivä sekä -aika
- Kirjoittajan nimimerkki sekä näyttönimi
- Kirjoittajan bio
- Renderöinti-informaatioita
- Käyttäjän luontipäivä
- Käyttäjän seuraajamäärä sekä suosikkien määrä
- Käyttäjän aikavyöhyke
- Paikan nimi, ID, tyyppi ja maa, josta twiitti on lähetetty
- Applikaatio, josta twiitti on lähetetty. (Krikorian 2010.)



Kuva 8. Valokuvan metadata

4.4 Tekstianalyysi big datasta

Big datan analysoimisen ydin on tehdä jäsentämättömästä datasta jäsenneltyä tai osittain jäsenneltyä dataa sekä analysoida sitä. Datasta halutaan poimia merkityksellinen ja relevantti informaatio ja jäsennellä se siten, että sitä voidaan analysoida jatkossa helpommin. Jos on tiedossa, minkälaista informaatiota tekstistä halutaan, voidaan käyttää muun muassa avainsanahakua. Yleensä ei kuitenkaan tiedetä tarkkaan mitä informaatiota datasta saadaan, jolloin on käytettävä jotain muuta tekniikkaa.

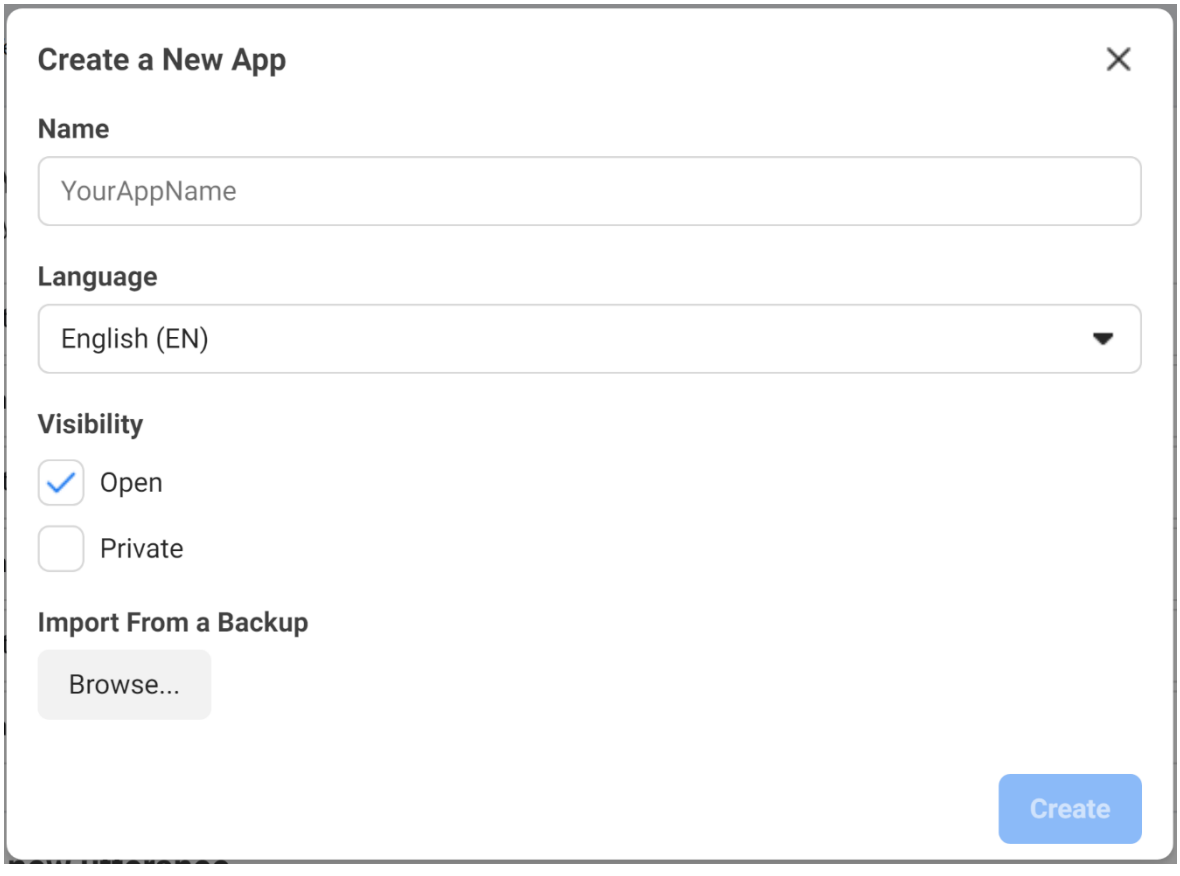
Tekstianalysoidessa big dataa on käytössä useita eri tekniikoita, muun muassa luonnollisen kielen käsittelyä (NLP) ja datalouhinta (data mining). Mikään tekniikka ei ole itsessään kovin tehokas tai nopea, vaan kaikki, mikä auttaa big datan prosessoinnissa, on lähes välttämätöntä analysoinnissa. Varsinainen prosessi on tekstinkäsittelyssä tärkeää, eivät niinkään käytössä olevat tekniikat ja työkalut. Luonnollisen kielen käsittelyä käytetään usein, kun käsittelyssä on jäsentämätöntä dataa. (Linguamatics.)

5 Chatbot-pilotin kehitys

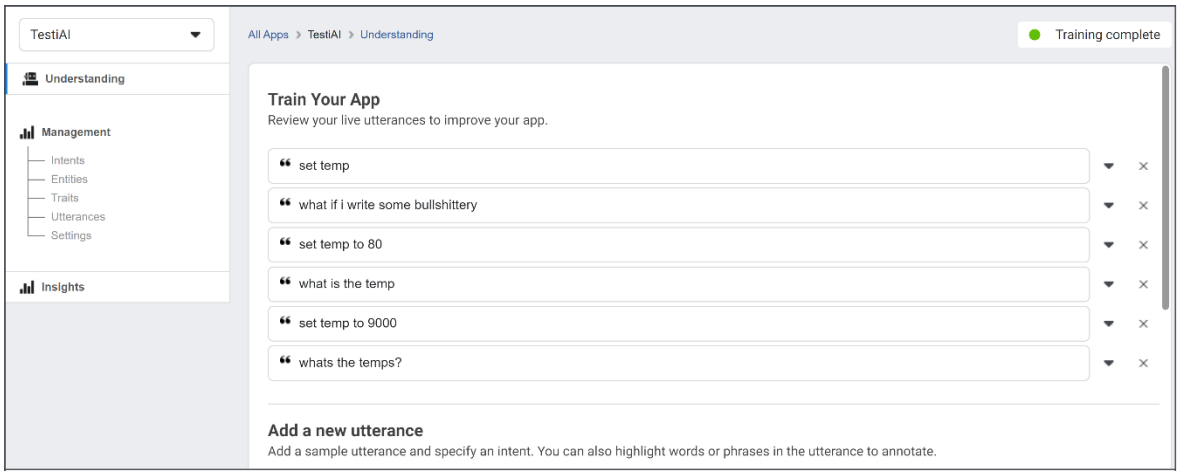
5.1 Chatbot-pilotti

Wit.ai on maksuton, nykyisin Facebookin omistuksessa oleva avoimen lähdekoodin tekoälyratkaisu tekstin ja puheen tulkitsemiseen. Wit.ai tukee lukuisia eri kieliä niin puheen kuin tekstin tunnistuksessa. Wit.ai tarjoaa palveluitaan niin ei-kaupalliseen, kuin kaupalliseen käyttöön ilmaiseksi. Wit.ai ei vaadi rekisteröitymistä, vaan sinne kirjaudutaan Facebook-tunnuksilla. Github-tunnuksilla kirjautuminen poistui käytöstä vuoden 2020 lopussa. (Wit.ai 2021.)

Wit.ai toimii verkkokäyttöliittymän kautta. Käyttöliittymästä pystytään opettamaan ja ylläpitämään chatbotteja (kuva 10). Verkkokäyttöliittymästä uuden sovelluksen voi luoda ”New App” – painikkeesta (kuva 9). Sovellukselle annetaan nimi, luonnollisen kielen käsittelyssä käytettävä kieli, sekä valitaan sovelluksen sisältämän datan yksityisyysasetukset. Nämä asetukset määräävät näkyykö data alustan muille käyttäjille.

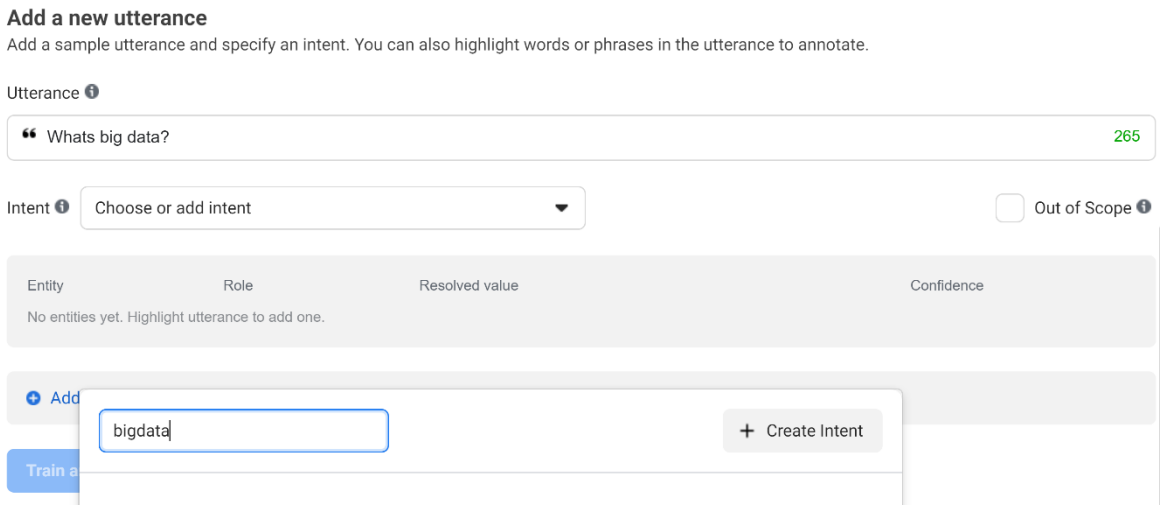


Kuva 9. Uuden projektin luonti



Kuva 10. Wit.ai käyttöliittymä

Wit.ai:n opettaminen tapahtuu siten, että sille kerrotaan esimerkkilauseita (utterance). Kuvassa 11 on esimerkki lauseiden syötöstä. Lauseisiin ei tarvitse kirjoittaa kaikkia mahdollisia tapoja, miten käyttäjä voi asiaa kysyä, sillä Wit.ai pystyy päättelemään asiayhteyden muutamista esimerkeistä. Kuitenkin mitä enemmän esimerkkilauseita kirjoittaa, sitä paremmin Wit.ai ymmärtää käyttäjien lauseita. Tämän jälkeen esimerkkilauseille pitää valita intentiokategoria, jonka perusteella Wit.ai lähettää vastauksen takaisin chatbotille. Varsinaista dialogia ei pysty Wit.ai:n puolella tekemään, vaan vastaukset pitää ohjelmoida manuaalisesti, jolloin hyödynnetään Wit.ai:n antamia intentioita ja entiteettejä.



Kuva 11. Esimerkkilauseiden syöttäminen Wit.ai:lle

Käyttäjän kirjoittamia kysymyksiä chatbotille voidaan käydä läpi kuvan 12 esitetyllä tavalla. Lauseita voidaan käsitellä samalla tavalla, kuin opetusvaiheessa annettuja lauseita. Näille

pystytään tällä tavoin varmistamaan oikea intentio. Jos oikeaa intentiota ei ole olemassa, voidaan luoda uusi intentio.

Kuva 12. Chatbotille annetut lauseet

Wit.ai palauttaa myös luotettavuusluvun (confidence), joka kertoo kuinka luotettavasti Wit.ai pystyi antamaan annetusta lauseesta sopivan intention. Luotettavuusluku on välillä 0-1. Wit.ai palauttaa aina parhaalla luotettavuusluvulla löytyvän intention. Jos halutaan, että tietyn luotettavuusluvun alta ei anneta vastausta, joudutaan itse määrittelemään ohjelmiston puolella luotettavuusluvulle raja-arvo. Tämän jälkeen voidaan manuaalisesti tehdä raja-arvon alle oleville vastauksille jonkinlainen ”Anteeksi, nyt en ymmärtänyt” – viesti. Tähän viestiin pystytään esimerkiksi kertomaan sähköpostiosoite, johon voidaan olla yhteydessä.

Entiteetteihin (entity) voidaan tallentaa parametreja käyttäjän syötteestä. Esimerkiksi jos käyttäjä kysyy tietyn kaupungin liikkeen aukioloaikaa, voidaan tästä tallentaa kaupunki entiteettiin. Tätä entiteettiä voidaan helposti hyödyntää oikean liikkeen aukioloaikojen hakemiseen. Entiteettejä voidaan luoda itse, tai käyttää valmiita sisäänrakennettuja entiteettejä. Esimerkkinä sisäänrakennetusta entiteetistä on wit/temperature, joka etsii ja tallentaa käyttäjän syötteestä lämpötila-arvon, joko celsiuksina tai fahrenheitteina.

5.2 JSON-rakenne

Wit.ai:lle lähetetään http-kutsulla (kuva 13) käyttäjän antama syöte. Wit.ai palauttaa JSON-muodossa datan (kuva 14), jossa on käyttäjän syöte, intention id, nimi ja luotettavuusluku, sekä mahdolliset entiteetit. Wit.ai ei sisällä työkaluja chatbotin julkaisemiseksi, vaan sille lähetetään rajapintakutsuilla käyttäjän syöte, jonka mukaan Wit.ai lähettää vastauksen takaisin.

```
async function getData(){
  q = encodeURIComponent(inputtext);
  uri = 'https://api.wit.ai/message?v=20200513&q=' + q;
  const response = fetch(uri, {headers: {Authorization: auth}}).then(res => res.json());
  return response;
}
```

Kuva 13. Esimerkki Wit.ai:n http-pyynnöstä


```
{
    "text": "'Set temperature to 23C'",
    "intents": [
        {
            "id": "663975967648141",
            "name": "settemp",
            "confidence": 0.782
        }
    ],
    "entities": {
        "wit$temperature:temperature": [
            {
                "id": "397595431640839",
                "name": "wit$temperature",
                "role": "temperature",
                "start": 20,
                "end": 23,
                "body": "23C",
                "confidence": 0.9675,
                "entities": [],
                "unit": "celsius",
                "type": "value",
                "value": 23
            }
        ]
    },
    "traits": {}
}
```


Kuva 14. Esimerkki JSON vastauksesta

5.3 Jatkokehitys

Tulevaisuudessa olisi tarkoitus tehdä valmis chatbot, jota olisi helppo myydä sellaisenaan ja vaivatonta lisätä suurimmalle osalle sivustoista. Tämä tarkoittaa sitä, että olisi valmiina Javascript-kokonaisuus sekä Wordpress-lisäosa. Nämä kattaisivat suurimman osan verkkosivuratkaisuista.

Chatbotti tarvitsee toimiakseen Wit.ai:n määrittelemät uniikit server access tokenin ja client access tokenin. Nämä ovat tarkoitus tallentaa globaaliin tiedostoon, esimerkiksi .env-tiedostoon, josta ne olisivat helposti muutettavissa projektikohtaisesti.

Chatbotista on saatava luotettava, jolloin chatbot vastaa järkevästi käyttäjälle. Tämä vaatii riittävää opetusta Wit.ai:lle. Tätä varten voidaan myös käyttää Wit.ai:n antamaa luotettavuuslukua. Jos luotettavuusarvo on liian alhainen, annetaan vastaukseksi jokin geneerisempi vastaus. Tähän vastaukseen voidaan laittaa muun muassa yhteystietoja tai ohjata

jollekin muulle kanavalle, josta voidaan kysyä vastausta. Tämä luotettavuusluvun raja-arvo olisi hyvä olla muokattavissa, sillä käyttökohteesta riippuen voidaan haluta esimerkiksi vain erittäin luotettavia vastauksia.

Chatbot tarvitsee myös perinteisen keskustelu ulkoasun, jonka ulkonäköä voitaisiin muokata sivustolle sopivaksi ja jossa näkyisi myös keskusteluhistoria. Keskusteluhistoria voi olla hyvin lyhytaikainen. Esimerkiksi historiaa ei tallenneta mihinkään, vaan historia katoaa sivuston päivittäessä. Toinen vaihtoehto on historian tallentaminen selaimen välimuistiin milloin historia pysyy kunnes välimuisti tyhjennetään. Käyttäjän olisi kuitenkin hyvä pystyä selaamaan käytyä keskustelua.

Chatbotille pystytään myös valitsemaan käytetty kieli Wit.ai:n käyttöliittymästä. Muun muassa suomen kieli on tuettuna Wit.ai:ssa, mutta se on merkittynä vielä beta-vaiheeseen. Tämän takia suomen kielen ratkaisu vaatisi testausta.

Lisäksi kannattaisi kerätä dataa käyttäjien yleisimmistä kysymyksistä. Jos sivustolla kysytään usein jotain perustietoja, kuten aukioloaikoja, voidaan tästä päätellä, ettei tämä informaatio ole tarpeeksi selkeästi sivustolla esillä. Lisäksi jos taas jotain kysymyksiä kysytään paljon, ja chatbot ei pysty antamaan kunnollista vastausta siihen, voidaan kysymykselle luoda vastaus erikseen.

6 Yhteenveto

Työn tavoitteena oli tehdä chatbot-pilotti, sekä tutustua Wit.ai käyttöönottoon ja käyttämiseen. Työssä käytiin läpi perinteistä tekoälyn toimintaa, tekstianalyysia tekoälyn avulla sekä big datan analysointia. Työssä perehdyttiin myös esimerkiksi tekstianalyysiin tilastollisella sekä lingvistisellä tasolla, sekä tekstintunnistusteknologioihin. Työssä tutustuttiin myös muutamaan yleiseen tekoälypohjaisen tekstianalyysin käyttökohteeseen.

Tuloksena pilotti pystyy vastaanottamaan käyttäjän syötteen sekä antamaan JSON-muotoisen vastauksen. Vastauksen pohjalta pilotti pystyy vastaamaan muutaman testi-intention perusteella. Testi-intentioita pystytään lisäämään pilotille melko vaivatta. Pilotin jatkokehitys odottaa yrityksen perustamista, joka on kirjoitushetkellä tauolla tämänhetkisen koronapandemian vuoksi.

Varsinainen Wit.ai perehtyminen oli jouhevaa. Mitään suurempaa kokemusta tekoälypalveluntarjoajista ei ollut. Käyttöliittymä on hyvin intuitiivinen ja käyttäjäystävällinen, mutta dokumentaatioon perehtymiseen, sekä tarvittavan ympäristön pystytykseen kului aikaa.

Pilotista onnistuttiin tekemään hyvä pohja tulevaa jatkokehitystä varten, vaikkakaan mitään monipuolisia ominaisuuksia ei saatu aikaiseksi. Vaatii vielä työpanostusta, jotta kaikki halutut chatbotin toiminnot pystyttäisiin helposti lisäämään ja ylläpitämään sivustokohtaisesti. Tällä hetkellä pilottia pystytään hyödyntää hyvin testiympäristönä.

Lähteet

Ajanki, A. 2018. Differences between machine learning and software engineering. Viitattu 04.05.2021. Saatavissa https://futurice.com/blog/differences-between-machine-learning-and-software-engineering

Appen. 2020. What is Training Data? Viitattu 24.05.2020. Saatavissa https://appen.com/blog/training-data/

Bernard, R. 2019. Deep Learning to the Rescue. Viitattu 04.05.2021. Saatavissa https://www.securityinfowatch.com/video-surveillance/video-analytics/article/21069937/deep-learning-to-the-rescue

Big Data Framework. 2019a. Where does 'Big Data' come from? Viitattu 27.05.2021 Saatavissa https://www.bigdataframework.org/short-history-of-big-data/

Big Data Framework. 2019b. Data Types: Structured vs. Unstructured Data. Viitattu 11.12.2020. Saatavissa https://www.bigdataframework.org/data-types-structured-vs-unstructured-data/

Christianson, P. 2020. Billions of Miles of Data: The Autonomous Vehicle Training Conundrum. Viitattu 30.04.2021 Saatavissa https://blog.cloudfactory.com/autonomous-vehicle-training-conundrum

CoNLL 2018 Shared Task. 2018. Viitattu 11.12.2020. Saatavissa https://universaldependencies.org/conll18/results.html

Cox, M., Ellsworth, E. 1997. Application-controlled demand paging for out-of-core visualization. Saatavissa https://www.nas.nasa.gov/assets/pdf/techreports/1997/nas-97-010.pdf

Curtis, B. 2020. YourTechDiet: What are the 7 V's of big data? Viitattu 11.12.2020. Saatavissa https://www.yourtechdiet.com/blogs/7vs-big-data/

Denning, P. 1990. Saving All the Bits. Saatavissa https://ntrs.nasa.gov/api/citations/19910023503/downloads/19910023503.pdf

Elements of AI. Viitattu 21.05.2021 Saatavissa https://course.elementsofai.com/fi/

F-Secure Labs. 2019. CAPTCHA-22: Breaking Text-Based CAPTCHAs with Machine Learning. Viitattu 05.05.2021. Saatavissa https://labs.f-secure.com/blog/captcha22/

Google Machine Learning. 2018. Viitattu 04.05.2021. Saatavissa https://developers.google.com/machine-learning/guides/text-classification

Google Search Central. 2021. Understand how structured data works. Viitattu 25.03.2021. Saatavissa https://developers.google.com/search/docs/guides/intro-structured-data#search-appearance

Holley, R. 2009. Analysing and Improving OCR Accuracy in Large Scale Historic Newspaper Digitisation Programs. Viitattu 15.03.2021. Saatavissa http://www.dlib.org/dlib/march09/holley/03holley.html

ISMP. 2014. Misidentification of Alphanumeric Symbols. Viitattu 24.05.2021. Saatavissa https://www.ismp.org/resources/misidentification-alphanumeric-symbols

Kielipankki. Viitattu 11.12.2020. Saatavissa: https://www.kielipankki.fi/corpora/finnwordnet/

Koleva, N. 2020. When and When Not to Use Deep Learning. Viitattu. 04.05.2021. Saatavissa https://blog.dataiku.com/when-and-when-not-to-use-deep-learning

Krikorian, R. 2010. Map of a Twitter Status Object. Viitattu 28.05.2021 Saatavissa http://www.slaw.ca/wp-content/uploads/2011/11/map-of-a-tweet-copy.pdf

Lehto, M., Neittaanmäki, P., Nyrhinen, R., Ojalainen, A., Pölönen, I., Rautiainen, I., Ruohonen, T., Tuominen, H., Vähäkainu, P., Äyrämö, S. & Äyrämä, S. 2018. Tekoälyn perusteita ja sovelluksia. Informaatioteknologian tiedekunta. Jyväskylä.

Linguamatics. What is Text Mining, Text Analytics and Natural Language Processing? Viitattu 24.05.2021. Saatavissa https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing

McNulty, E. 2014. Understanding big data: the seven V's. Viitattu 11.12.2020. Saatavissa https://dataconomy.com/2014/05/seven-vs-big-data/

O'Reilly, T. 2005. What Is Web 2.0. Viitattu 12.05.2021. Saatavissa https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html

Pay, B. 2015. The Difference Between OCR and ICR and Why It Matters for Organizations Using DMS. Viitattu 05.05.2021. Saatavissa https://www.efilecabinet.com/the-difference-between-ocr-and-icr-and-why-it-matters-for-organizations-using-dms/

Press, G. 2013. A Very Short History Of Big Data. Viitattu 13.05.2021. Saatavissa https://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/?sh=382e154f65a1

Rivera, A. 2019. How Does OCR Work? Viitattu 24.05.2021. Saatavissa https://www.efilecabinet.com/how-does-ocr-work/

Saraf, H. 2020. Pushing the Envelope on ICR Accuracy in Hand-written Forms. Viitattu 24.05.2021 Saatavissa https://www.mantralabsglobal.com/blog/document-parser-icr-intelligent-character-recognition/

SAS 2019. Natural Language Processing: What it is and why it matters. Viitattu 11.12.2020. Saatavissa https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processingnlp.html.

Stanford University. 2008. Viitattu 20.03.2021. Saatavissa https://cs.stanford.edu/people/eroberts/courses/soco/projects/2008-09/colossus/colossus.html

Statista. 2020. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2024. Viitattu 11.12.2020. Saatavissa https://www.statista.com/statistics/871513/worldwide-data-created/

Tilly, C. 1980. The Old New Social History and the New Old Social History. Viitattu 13.05.2021. Saatavissa https://www.jstor.org/stable/40241514?seq=1

TurkuNLP. 2020. Viitattu 11.12.2020. Saatavissa http://turkunlp.org/Turku-neural-parser-pipeline/

Verhulst, S. 2013. Big Data and Academia: the launch of Rennselaer IDEA. Viitattu 12.05.2021. Saatavissa https://blog.thegovlab.org/post/big-data-and-academia-the-launch-of-rennselaer-idea

Walker, N. 2020. Automated Essay Scoring Explained. Viitattu 11.12.2020. Saatavissa https://blog.virtualwritingtutor.com/automated-essay-scoring-explained/

Wiener, J., Bronson N. 2014. Facebook's Top Open Data Problems. Viitattu 20.03.2021 Saatavissa https://research.fb.com/blog/2014/10/facebook-s-top-open-data-problems/

Wit.ai. 2021. Viitattu 05.04.2021 Saatavissa https://wit.ai

Ziemann, M. 2017. Yle. Uutinen. Viitattu 20.03.2021 Saatavissa https://yle.fi/uutiset/3-9658191