

Generatiivisen tekoälyn hyödyntäminen datan rikastuttamisessa

LAB-ammattikorkeakoulu

Insinööri (YAMK)

2024

Jake Horn

Tiivistelmä

Tekijä(t) Jake Horn	Julkaisun laji Opinnäytetyö, YAMK	Valmistumisaika 2024
	Sivumäärä 42	
Työn nimi Generatiivisen tekoälyn hyödyntäminen datan rikastamisessa		
Tutkinto ja koulutusala Insinööri (YAMK), IoT:stä tekoälyyn		
Tiivistelmä <p>Tämän opinnäytetyön tavoitteena oli selvittää, miten generatiivista tekoälyä voidaan hyödyntää datan strukturoinnissa ja rikastamisessa. Lisäksi työssä haluttiin selvittää, kuinka tekstintunnistusta voidaan hyödyntää datan strukturoinnin tukena.</p> <p>Kehittämistyönä toteutetun opinnäytetyön tutkimusmenetelmänä käytettiin suunnitellutieteen mallia, jossa luotiin konseptintodistus malli, joka toimi työn konstruktiona. Konseptintodistus mallia hyödyntämällä pystyttiin vastaamaan työlle asetettuihin tutkimuskysymyksiin ja lisäämään tietopääomaa.</p> <p>Työ saavutti sille asetetut tavoitteet. Tehdyn työn perustella voitiin todeta, että generatiivista tekoälyä voidaan hyödyntää strukturoimattoman tiedon strukturoinnissa ja datan rikastuttamisessa. Lisäksi voitiin todeta, että tekstintunnistamista voidaan hyödyntää tukemaan strukturoimattoman tiedon strukturoimista. Työssä luotua konseptintodistus mallia on mahdollista käyttää uusissa tutkimuksissa, joissa hyödynnetään oikeaa dataa ja datamalleja.</p>		
Asiasanat Tekoäly, generatiivinen tekoäly, GPT, datan strukturointi		

Abstract

Author(s) Jake Horn	Type of Publication Thesis, UAS	Published 2024
	Number of Pages 42	
Title of Publication Utilizing generative artificial intelligence for data enrichment		
Degree, Field of Study Master of Engineering, From IoT to AI		
Abstract <p>The purpose of this thesis was to explore how generative artificial intelligence can be utilized for data structuring and enrichment. Another objective of this thesis was to find out how text recognition can be used to support data structuring.</p> <p>The used research method was design science. The proof-of-concept (PoC) model was created using this the method. The proof-of-concept served as an artifact of this work. By using the proof-of-concept model, it was possible to answer the research questions and increase knowledge capital.</p> <p>This thesis achieved its set goals. Based on the conducted work, it was concluded that generative artificial intelligence can be utilized in structuring unstructured data and enriching dataset. Additionally, it was concluded that text recognition can support structuring unstructured data. Created proof-of-concept model can be utilized in new research projects where real data and data models are in use.</p>		
Keywords Artificial intelligence, generative artificial intelligence, GPT, data structuring		

Sisällys

1	Johdanto.....	1
1.1	Työn tausta ja tavoitteet.....	1
1.2	Tutkimuskysymykset ja rajaus	1
1.3	Tutkimusmenetelmä.....	2
2	Tekoäly.....	4
2.1	Älykkyyys	4
2.2	Luokittelu.....	5
2.3	Koneoppiminen	6
2.4	Syväoppiminen	7
2.5	Luonnollisen kielen käsittely	8
2.6	Haasteet.....	10
3	Generatiivinen tekoäly	12
3.1	GPT.....	12
3.2	Kehotesuunnittelu	13
3.3	Omistajuus	13
4	Tekstintunnistus	15
4.1	Tekstintunnistamisen vaiheet.....	15
4.2	Päätelaitteella vai palvelimella.....	16
4.3	Palvelut ja ohjelmistot	17
4.4	Käyttökohteita	18
5	Data.....	19
5.1	Struktuurit.....	19
5.2	Datan laatu.....	20
6	Toteutus	22
6.1	Työn toteutuksen vaiheet.....	22
6.2	Testitapaus 1: Vahingon tunnistaminen vapaasta tekstistä.....	28
6.3	Testitapaus 2: Tietojen tunnistus kuitista otetusta kuvasta.....	29
6.4	Testitapaus 3: Tietojen tunnistaminen pdf-tiedostosta	31
6.5	Testitapaus 4: GPT-4 mallin tekstin tunnistus kuvasta	34
7	Yhteenveto ja pohdinta.....	36
	Lähteet.....	39

1 Johdanto

1.1 Työn tausta ja tavoitteet

Data ja datamallit ovat jo pitkään ohjanneet tiedonkeruuta. Datan halutaan olevan tietyssä struktuurissa, jotta sen käyttö jatkossa olisi helpompaa ja kerättävän datan laatu olisi parempaa. Tämä tuo usein haasteita siinä sovelluksen kehitysvaiheessa, jossa tulee ottaa huomioon loppukäyttäjä. Jotta haluttu tieto saadaan haluttuun datamallin muotoon, päädytään usein tilanteeseen, jossa loppukäyttäjältä kysytään vaikeita ja spesifejä kysymyksiä. Näihin kysymyksiin on esimerkiksi tehty erinäisiä valintalistoja, joista käyttäjän tulisi paras vaihtoehto valita. Tämän valinnan tekeminen voi joissakin tapauksissa olla loppukäyttäjälle haastavaa ja aiheuttaa turhautumista, jolloin valinta saatetaan tehdä "summamutikassa". Näissä mahdollisesti turhautuneena tehdyissä valinnoissa vaikutus taas näkyy suoraan kerättävän datan laadussa. Mitä enemmän on vastattu "vähän sinnepäin", sitä huonompi on datan laatu.

Tekoäly on kehittynyt viime vuosina kovaa vauhtia. Tekoälyn käyttökohteet ovat lisääntyneet ja uusia tekoälyä hyödyntäviä palveluita tulee jatkuvasti lisää. Lisäksi tekoälyn käytöstä on tullut helposti lähestyttävämpää ja sen käyttöönotto sovelluksissa ja palveluissa on nykyään helppoa. Myös yrityksillä on kova kiinnostus tekoälyä kohtaan ja uusia käyttökohteita pyritään jatkuvasti löytämään. (Hjelt 2023; State of AI in Finland 2020.)

Työn aikana toteutettiin konseptitodistuksen malli (PoC), jolla kokeiltiin tiedon strukturointia generatiivista tekoälyä hyödyntäen. Mallin avulla tehtiin kokeiluja, jossa vapaasta tekstistä sekä kuvasta tunnistettua tekstistä saatua tietoa pyrittiin samaan halutun datamallin muotoon. Käyttöliittymällä tehdyillä toiminnoilla oli tarkoitus simuloida mahdollisen loppukäyttäjän tekemiä toimintoja.

Tämän kehittämistyön tavoitteena oli saada konseptitodistus mallin avulla vastauksia esitettyihin tutkimuskysymyksiin ja tämän lisäksi kartuttaa uutta tietoa generatiivisen tekoälyn hyödyntämisestä tietyllä sovellusalueella.

1.2 Tutkimuskysymykset ja rajaus

Tässä opinnäytetyössä vastattiin seuraaviin tutkimuskysymyksiin:

1. Miten generatiivista tekoälyä voidaan hyödyntää datan strukturoinnissa?
2. Miten generatiivisella tekoälyllä voidaan rikastuttaa dataa?
3. Miten tekstintunnista voidaan hyödyntää datan strukturoinnin tukena?

Tässä kehittämistyössä keskitytään toteuttamaan generatiivisen tekoälyä ja tekstintunnista hyödyntävä konseptitodistuksen malli. Työstä on rajattu pois generatiivisen tekoälyn sekä tekstintunnistukseen tarkoitettujen sovellusten ja palveluiden vertailu. Konseptitodistuksen mallista saatuja tuloksia käsitellään sanallisesti, eikä työssä tulla luomaan omia luotettavuusmittareita. Konseptitodistuksen mallissa luodut tietomallit sekä käytettävä tieto ovat yleispäteviä, eivätkä pohjaa minkään tietyn yrityksen tai sovelluksen käyttämiin tietomalleihin tai tietoon.

1.3 Tutkimusmenetelmä

Table 1. Design-Science Research Guidelines	
Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Taulukko 1. Suunnittelutieteen soveltamisen periaatteet (Hevner ym. 2004)

Tämän työn tutkimusmenetelmänä käytetään suunnittelutieteen mallia. Suunnittelutieteen lähtökohtana on löytää ratkaisu käytännön ongelmaan hyödyntäen olemassa olevaa tietopääomaa, sekä tietopääoman kasvattamien ongelmaratkaisutyön aikana syntyneestä uudesta tiedosta. Suunnittelutieteen hyödyntäminen vaatii systemaattisuutta, jotta työn aikana syntynyt tieto on myös muiden, kuin tekijän itsensä hyödynnettävissä. Tämän opinnäytetyön suunnittelussa käytettiin taulukon 1 mukaista mallia.

Periaate 1 on tuottaa konstruktio (Design as an Artifact). Konstruktio voi olla esimerkiksi laite, järjestelmä, tietokoneohjelma tai -ohjelmisto. Tämän työn ”Toteutus” osassa luodaan konseptitodistuksen malli, joka toimii tämän työn konstruktiona. (Hevner ym. 2004, 82–84.)

Periaate 2 on todellinen ongelma (Problem Relevance). Työn tulee ratkaista jokin todellinen tiedossa oleva ongelma luodun konstruktion avulla. Tässä työssä esitetyt tutkimuskysymykset toimivat tämän työn todellisina ongelmina. Tässä työssä konstruktion avulla pyritään samaan vastauksia esitettyihin tutkimus kysymyksiin, sekä lisäksi kasvattamaan olevassa olevaa tietopääomaa. Tutkimuskysymysten asettelussa on huomioitu lähtökohdat, tehty rajaukset sekä kuvattu ongelmat, joihin pyritään löytämään ratkaisuja. (Hevner ym. 2004, 84–85.)

Periaate 3 on tulosten arviointi (Design Evaluation). Tässä työssä luodun konstruktion toimivuus voidaan todeta, kun luotu konseptitodistuksen malli on toiminnallisesti käytettävä ja esitettyihin tutkimus kysymyksiin on sen avulla saatu vastauksia. (Hevner ym. 2004, 85–87.)

Periaate 4 on tieteellinen arvo (Research Contributions). Tässä työssä luodun konstruktion avulla saatujen vastausten perusteella voidaan asettaa uusia tutkimuskysymyksiä uusille tutkimuksille. Tämän opinnäytetyön ”Toteutus” osiossa dokumentoidaan konseptitodistuksen mallin toteutus vaiheet, joiden avulla samankaltainen sovellus voidaan toteuttaa ja sen pohjalta voidaan tehdä jatkokehitystä ja jatkotutkimusta. (Hevner ym. 2004, 87.)

Periaate 5 on systemaattisuus (Research Rigor). Tämän opinnäyte työn suunnittelussa on käytetty suunnittelutieteen soveltamisen periaatteita, jotka ovat tässä työssä kuvattuina. Tässä työssä tehtävän konstruktion vaatimusmäärittelynä on tehdä sovellus, jolla kyetään samaan vastauksia esitettyihin tutkimuskysymyksiin tehdyt rajaukset huomioiden. Saadut tulokset tullaan käsittelemään ja arvioimaan ”Yhteenveto ja pohdinta” osuudessa. (Hevner ym. 2004, 87–88.)

Periaate 6 on tunnetun tiedon soveltaminen (Design as a Search Process). Tässä tutkimustyössä tutustutaan aiheeseen liittyvään teoriaan, jonka avulla pyritään luomaan selkeä kuva kehittämistyön aihealueeseen. Työssä tehtävän konstruktion toimivuutta ja siihen tehtyjä valintoja tullaan testaamaan useammalla testitapauksella ja niistä saatuja tuloksia tullaan vertaamaan asetettuihin tavoitteisiin. (Hevner ym. 2004, 88–90.)

Periaate 7 on tulosten julkaisu (Communication of Research). Tästä työstä saadut tulokset ovat osana tätä kehittämistyötä. Työn tavoitteiden, toteutuksen ja tulosten kuvauksissa on otettu huomioon se, että lukija kykenee sen perustella arvioimaan työn tulosten hyödynnettävyyttä ja johdonmukaisuutta. (Hevner ym. 2004, 90.)

2 Tekoäly

Niin kauan kuin on ollut tietokoneita, on myös ollut tekoälyn kehitystä. Jo 1950-luvulla heti ensimmäisten tietokoneiden käyttöönoton jälkeen alkoivat ensimmäisten tekoälymenetelmien kehittäminen. Uutena tieteenalana ja terminä tekoäly voidaan katsoa syntyneen vuonna 1956 järjestetyssä työpajassa. Kyseisessä työpajassa tehtiin ensimmäinen tekoälysovellus, sekä vakuututtiin siitä, että tulevaisuudessa tietokoneet kykenevät oppimaan ja ymmärtämään kirjoitettua kieltä ja tarjoamaan vastauksia esitettyihin kysymyksiin. Tekoälyn kehitys ei kuitenkaan ole ollut suoraviivaista, vaan vasta 2000-luvulla tekoäly sovellukset ovat alkaneet vaikuttamaan ihmisten arkeen. (Kolari & Kallio 2023, 18.)

Tekoälyllä tarkoitetaan tietokoneiden kykyä tehdä päätöksiä perustuen annettuun tietoon eli dataan. Sillä tarkoitetaan tietokoneiden suorittamia toimintoja, jotka vaativat ihmismäistä älykkyyttä. Ohjelmistot, jotka pohjautuvat tekoälyyn perustuvat ihmisten luomiin algoritmeihin, jotka kykenevät myös oppimaan sekä kehittämään omaa toimintaansa. Pääpiirteisä tekoäly voidaan jakaa kahteen luokkaan: heikkoon ja vahvaan tekoälyyn. Heikossa tekoälyssä tietokone voi suorittaa vain sille määriteltäviä tehtäviä, kun taas vahva tekoäly kykenee toimimaan ja ajattelemaan, koska sillä on laajempi ymmärrys asioista. Tietoisuuden omaavaa tekoäly voidaan kutsua supertekoälyksi, joka kykenee kehittymään ja luomaan itsestään jatkuvasti parempia versioita. Ensimmäiset enemmän hyödynnetyt tekoälysovellukset perustuivat konenäön ja kuvien analysoinnin hyödyntämiseen, mutta viime aikoina hyödyntäminen on laajentunut tekstin analysointiin sekä muihin käyttökohteisiin. (Kolari & Kallio 2023, 14; Kalmi 2021, 14.)

2.1 Älykkyys

Älykkyuden määrittäminen itsessään on haastavaa, koska älykkyuden määritelmä ei ole yksiselitteinen ja itsestään selvä. Kuitenkin vertailemalla ihmisen ja tekoälyn toimintaa voidaan saada, jonkin näköinen käsitys tekoälyn älykkyudesta ja toimintakyvystä. Esimerkiksi vertailemalla tekoälyn ja ihmisen suorittamia kielen käännöksiä havaitaan, että tekoäly suoriutuu käännöstehtävästä huomattavasti ihmistä nopeammin, mutta ihmisen tekemässä käännöksessä on paremmin otettu huomioon konteksti, sekä kulttuurilliset erot. Tekoäly kykenee siis kääntämään minkä tahansa kielen, mille tahansa kielelle, mutta se ei välttämättä kykene samanlaiseen monipuoliseen kokonaisuuden tarkasteluun kuin ihminen. On kuitenkin tärkeää muistaa, että vertailulla ei kuitenkaan saada välttämättä täydellistä kuvaa tekoälyn kyvykkyyksistä. (Toivonen 2023, 17–18.)

2.2 Luokittelu

		AI taxonomy	
		AI domain	AI subdomain
Core	Reasoning		Knowledge representation
			Automated reasoning
			Common sense reasoning
	Planning		Planning and Scheduling
			Searching
			Optimisation
	Learning		Machine learning
Communication		Natural language processing	
Perception		Computer vision	
		Audio processing	
Transversal	Integration and Interaction		Multi-agent systems
			Robotics and Automation
			Connected and Automated vehicles
	Services		AI Services
	Ethics and Philosophy		AI Ethics
		Philosophy of AI	

Taulukko 2. Tekoälyn luokat ja alaluokat (Samoili ym. 2020)

Tekoälyyn löytyy useita luokittelutapoja. Taulukossa 2 on esitetty yksi tapa tekoälyn luokitteluun. Tämä luokittelutapa pitää sisällään kahdeksan tekoälyluokkaa, joilta löytyvät omat alaluokkansa.

Taulukon ensimmäisessä luokassa on päättely (Reasoning) ja sen alaluokat ovat tiedon esittäminen, automatisoitu päättely ja arkijärjen päättely. Tähän luokkaan menevät ne tilanteet, joissa tietokone muuttaa olemassa olevasta datasta tietoa. (Samoili ym. 2020, 12.)

Toisen luokka taulukossa on suunnittelu (Planning), joka alaluokat ovat suunnittelu ja aikataulut, haku ja optimointi. Tämän luokan tekoälyn pystyy suorittamaan tehtäviä erinäisten strategioiden avulla. Hyvänä esimerkkinä tämän luokan tekoälystä voidaan pitää älykästä robottia, joka kykenee muokkaamaan toimintaansa vaihtelevissa tilanteissa. (Samoili ym. 2020, 12.)

Kolmannessa luokassa on oppiminen (Learning). Tähän luokka yhdistää koneoppimisen alaluokan, joka tarkoittaa kykyä oppia, ennustaa ja päätellä muutoksia ilman tekoälyn uudelleen ohjelmointia. Tämä tekoälyn luokka on yleisesti aina vahvasti edustettuina muissa tekoälyn luokissa. On hyvä huomioida, että myös koneoppimisen saralla oppimista on

erilaista. Tekoälyn kohdalla oppiminen voidaan jaotella ohjattuun oppimiseen, ohjaamattomaan oppimiseen ja vahvistusoppimiseen. (Samoili ym. 2020, 12.)

Neljäntenä luokkana taulukosta löytyy kommunikaatio (Communication), jonka alaluokka on luonnollisen kielen käsittely. Luonnollisen kielen käsittelyllä tarkoitetaan erinäisiä kommunikaatiotapoja tietokoneen ja ihmisen välillä. Näistä voidaan pitää esimerkkeinä tekstin luomista, puheeksi muuttamista, sekä luokittelua. (Samoili ym. 2020, 12–13.)

Viides tekoälyn luokka taulukossa on havaitseminen (Perception), jonka alakategorioita ovat kuvantunnistus ja äänenkäsittely. Tähän luokkaan kuuluvia tekoälyn osa-alueita ovat esimerkiksi kasvojen tunnistus sekä muiden objektien tunnistaminen kuvista tai videoista. (Samoili ym. 2020, 13.)

Kuudennessa luokassa on integraatio ja vuorovaikutus (Integration and Interaction), joka sisältää alaluokat monitoimija-järjestelmä, robotiikka ja automaatio sekä automatisoidut ajoneuvot. Tämän luokan tekoälyt hyödyntävät erinäisiä kombinaatio- ja yhteistyömalleja, joiden avulla se pystyy hyödyntämään samanaikaisesti useista lähteistä tulevaa dataa toimintansa muokkaamiseen. Näistä esimerkkinä itseohjautuvat ajoneuvot pystyvät useiden data lähteiden pohjalta päättelemään tuleeko ajoneuvon kiihdyttää, jarruttaa vaiko jatkaa matkaa nykyisellä nopeudella. (Samoili ym. 2020, 13.)

Seitsemäs luokka pitää sisällään palvelut (Services). Näillä tarkoitetaan infrastruktuuria, sovelluksia ja alustoja, jotka tarjoavat tekoälypalveluita tai sovelluksia esimerkiksi pilviympäristöissä. (Samoili ym. 2020, 13–14.)

Kahdeksas luokassa on etiikka ja filosofia (Ethics and Philosophy). Tämä luokka on saanut viime aikoina paljon huomiota tekoälyn viimeaikaisen nopean kehityksen takia. Tässä luokassa painotetaan erityisesti tekoälyn vaikutusta yhteiskuntaan. (Samoili ym. 2020, 14.)

2.3 Koneoppiminen

Koneoppiminen on yksi tekoälyn osa-alueista, jossa koneoppimismallin on tarkoitus oppia itsenäisesti datasta sen sijaan, että mallille annettaisiin valmiiksi määriteltyjä ohjeita toimia eri tilanteissa. Sen perusajatuksena on, että mallit oppivat tekemään päätöksiä ja löytämään säännönmukaisuuksia dataa hyödyntämällä. Tällöin jokaista mahdollista tilannetta varten ei tarvita ohjelmoida omaa ratkaisua. Käytettävissä olevan datan ja koneen laskentatehojen nousu ovat lisänneet koneoppimisen käyttöä sovelluksissa. (Expert.ai Team 2022.)

Koneoppimismalleja opettaessa voidaan hyödyntää kolmea eri tapaa. Nämä opetustavat ovat ohjatusti, ohjaamattomasti ja vahvistetusti. Ohjelmistojen avulla mallit pystyvät tekemään johtopäätöksiä havaitsemaan ympäristöään ja tallentamaan tietoa. Aineistoa

tulkitseamalla mallit voidaan opettaa tulkitsemaan isoja ja pieniä asioita, pelaamaan erilaisia pelejä, seuraamaan muutoksia ilmastossa sekä säätämään kodin järjestelmiä kuten lämmitystä. (Merilehto 2018, 19.)

Ohjatussa oppimisessa käytetään opetusdataa. Opetusdata pitää sisällään oikeat vastaukset, jonka perusteella mallin opetus toteutetaan. Tässä mallissa koneet oppivat ennustamaan oikeita vastauksia uusista syötteistä. Ohjatun oppimisen mallia voidaan esimerkiksi käyttää asiakaskäyttäytymisen ennustamiseen aiempien tietojen perusteella. (The MathWorks, Inc 2022, 7.)

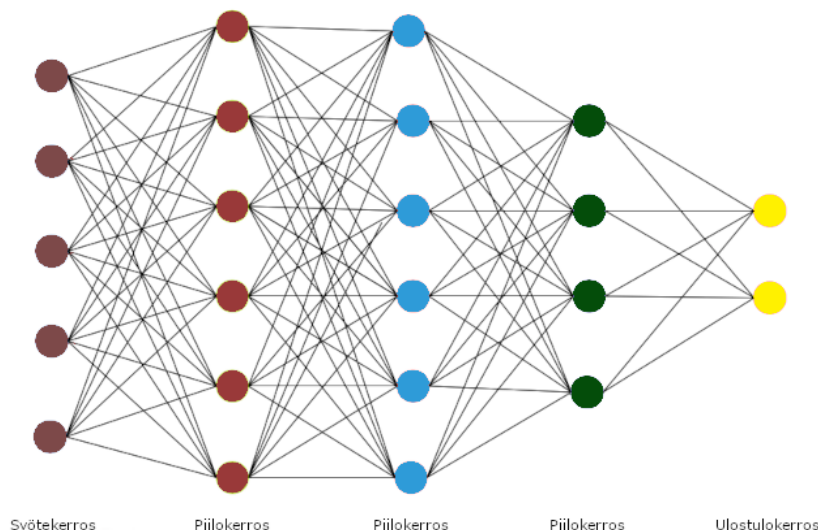
Ohjaamattomassa oppimisessa koneoppimismallit ei ole valmiita vastauksia, vaan se oppii tunnistamaan säännönmukaisuuksia ja suhteita datassa. Tämän tyyppisiä koneoppimisen malleja käytetään datan klusteroinnissa ja ryhmittelyssä, jotka auttavat tunnistamaan luonnollisia ryhmiä datassa. Tällaisia ovat esimerkiksi asiakassegmentit. (The MathWorks, Inc 2022, 8.)

Vahvistetussa oppimisessa koneoppimismallit ovat vuorovaikutuksessa ympäristönsä kanssa. Mallia opetetaan siten, että oikeista päätöksistä palkitaan ja virheistä rangaistaan. Tätä mallia voidaan esimerkiksi hyödyntää peleissä, joissa mallit saavat palautetta suorituksista ja optimoivat toimintaansa tämän perusteella saavuttaakseen tavoitteet mahdollisimman tehokkaasti. (Margaret Rouse 2023.)

Mallien opettamiseen löytyy paljon monipuolisia työkaluja päätöksentekoon, datan analysointiin ja ennustamiseen. Oppimismallin ja työkalun valintaan vaikuttavat usein käytettävissä oleva data ja mitä tavoitteita sen hyödyntämiselle on asetettu. Esimerkiksi ohjattu oppiminen soveltuu hyvin ennustamiseen, ohjaamaton oppiminen datan säännönmukaisuuksien löytämiseen ja vahvistettu oppiminen vuorovaikutteisiin tehtäviin, kuten pelit. Oppimismallien ymmärtäminen on keskeinen asia tehokkaan koneoppimisen toteuttamisessa. (Saumya 2024.)

2.4 Syväoppiminen

Syväoppiminen on koneoppimisen osa-alue, joka perustuu neuroverkkoihin, joilla on yritetty jäljitellä ihmisaivojen toimintaa. Neuroverkko on menetelmien joukko, joka on saanut innoituksensa biologiasta. Se koostuu toisiinsa kytketyistä neuroneista, joiden läpi syötetty data kulkee neuroneiden välisien yhteyksien kautta. Syväoppiminen on ollut merkittävä tekijä 2010-luvun tekoälyn kehityksessä, jolloin neuronikerroksia kyettiin lisäämään muutamista kerroksista, kymmeneen jopa satoihin. (Kolari & Kallio 2023, 131.)



Kuvio 1. Neuroverkon rakenne

Syväoppiminen vaatii paljon dataa ja laskenta tehoa neuroverkon koulutusta varten. Neuroverkko rakentuu kuvion 1 mukaan. Koulutettu neuroverkko toimii siten, että data puretaan numeroarvoiksi verkon syöteneuroneihin. Syöteneuroneista arvot välitetään askel kerralla syvempiin kerroksiin, joita kutsutaan piilokerroksiksi. Piilokerroksia yhdistellään siten, että lopulta verkon ulostulokerroksen neuroneille välittyy oikea lopputulos. Neuroverkoissa arvojen yhdistely perustuu neuronien välissä olevien yhteyksien painojen säätelyyn. Näitä arvoja voidaan säätää neuroverkon koulutuksen yhteydessä ja niillä jokaisella on vaikutusta saatuihin lopputuloksiin. (Kolari & Kallio 2023, 131–132.)

2.5 Luonnollisen kielen käsittely

Luonnollisen kielen käsittely on tekoälyn alaluokka, joka pitää sisällään koneen kykyä ymmärtää, prosessoida ja tuottaa ihmiskieltä. Siitä käytetään lyhennettä NLP, joka tulee englanninkielisistä sanoista ”Natural Language Processing”. NLP:tä hyödynnetään käännösten tekemisessä, yhteenvetojen luomisessa sekä yleisesti tekstin tuottamisessa. Yhteistä näillä kaikilla on tarkoitus luoda hyödyllistä tietoa. Esimerkkinä voidaan pitää tilannetta, jossa käännös halutaan kokonaisesta lauseesta. Yksittäisen sanan kääntämiseen ei NLP:tä tarvita, mutta lauseiden kääntämisessä sovelluksen tulee tulkita koko lausetta käännöksen tekemiseen. (Aditya ym. 2018, 161–162.)



Kuvio 2. Luonnollisen kielen käsittelyn kehityssykli (Hagiwara 2021)

Useiden NLP-sovellusten tai järjestelmien kehityksessä voidaan käyttää kuvion 2 mukaista interaktiivista kehitysmallia. Malli koostuu tutkimuksen, kehityksen ja käyttöönoton vaiheista. Sovellusten kehitysvaiheessa on tavallista siirtyä edestakaisin vaiheiden välillä kokeilujen ja virheiden kautta. (Hagiwara 2021.)

Tutkimusvaihe pitää sisällään datan keräämiseen sekä analyysin ja kokeilun alavaiheet. Useissa NLP-mallien ja sovellusten perustana on koneoppiminen, jossa hyödynnetään saatavilla olevaa dataa. Dataa voidaan kerätä useista eri lähteistä ja sitä käytetään kehityksessä NLP-mallin koulutukseen, validointiin ja testaukseen. Datan keräämisen jälkeen voidaan seuraavaksi suorittaa kokeiluja ja niiden analysointia. Näiden kokeilujen tavoitteena on löytää lupaavimmat lähestymistavat ja selvittää niiden toimivuutta. Näin kehitysvaiheessa säästytään turhalta ajankäytöltä, kun kokeiluja ja analysointia on jo suoritettu pienemmillä datamassoilla ja tiedetään mitkä lähestymistavat ovat lupaavimpia. (Hagiwara 2021.)

Kehitysvaiheessa ensimmäisenä on NLP-mallin opetus. Tässä vaiheessa on jo selvä ajatus lähestymistavasta ja mallin koulutuksen datamääriä sekä laskentatehoa voidaan alkaa nostamaan. Tässä vaiheessa mallien opettamisessa voi mennä päivä tai jopa viikkoja, joten datan määrän ja mallin kokoa on syytä nostaa asteittain. Kevyemmin ja vähemmällä datalla koulutettu malli voi toimia yhtä hyvin, jopa paremmin kuin suuret ja paljon dataa koulutetut mallit toimivat. Tässä vaiheessa on myös hyvä huomioida mallin koulutukseen käytettävien putkien toistettavuus, koska mallia saatetaan joutua uudelleen kouluttamaan myöhemmissä

vaiheessa uudestaan esimerkiksi parametrimuutosten takia. Koulutusvaiheen jälkeen malli tulee saada tuotantovalmiuteen. Tämä vaihe mukailee normaalin sovelluskehityksen parhaita käytäntöjä, kuten yksikkötestien kirjoittamista ja koodin siistimistä. (Hagiwara 2021.)

Käyttöönottovaiheessa sovellus julkaistaan loppukäyttäjälle käyttöön. Tehtyä NLP-mallia voidaan käyttää, kuin mitä tahansa muuta palvelua ja sen julkaisuprosessissa on hyvänä käytäntönä käyttää jatkuvan julkaisun periaatetta. NLP-mallin käyttöönoton jälkeen sen toimintaa tulee monitoroida. Monitoroinnin tarkoituksena on tarkkailla, toimiiko tehty NLP-malli siten, kuin se oli suunniteltu. Esimerkiksi, jos tuotannossa tuleva data poikkeaa koulutusdatasta, NLP-malli tuottaa vääränlaisia vastauksia ja koulutus tulisi suorittaa uudestaan tuotannosta saadulla datalla. (Hagiwara 2021.)

Useat sovellukset ja työkalut hyödyntävät jo NLP:tä, useasti jopa siten, että käyttäjä eivät sitä itse havaitse. Esimerkiksi YouTubessa on käytössä automaattinen tekstitys, joka kääntää videolla kuullun puheen englanniksi ja Gmail kykenee tunnistamaan sähköpostien joukosta roskapostit. NLP on nopeasti kehittyvä tieteenala ja uusia käyttökohteita syntyy jatkuvasti. (Hagiwara 2021; Kolari & Kallio 2023, 40.)

2.6 Haasteet

Tekoälyyn ja sen käyttöön liittyviä haasteita ovat muun muassa yksityisyyden heikkeneminen, tekoälyn väärinkäyttö, työmarkkinoiden muutokset, ammattien katoaminen, vastuukysymykset ja tekniset haasteet. Lähitulevaisuudessa nykyisiä työtehtäviä tulee poistumaan, koska tekoäly tekee ne paremmin, nopeammin ja halvemmalla. Toisaalta tekoälyn kehitys kuitenkin avaa uusia mahdollisuuksia ja uusia työtehtäviä. Näihin muutoksiin yhteiskunnan täytyy keksiä ratkaisuja niihin sopeutumiseen. (CSC.)

Tekoälyn nopeassa kehityksessä useimmin huomio keskittyy tekoälyn teknisiin ominaisuuksiin ja samalla jää huomioimatta sen vaikutukset ympäristöön, yhteiskuntaan ja yksilöön. Tekoälyllä saadaan monet toteutukset näyttämään helpoilta, vaikka usein se on vain harhaa siitä, että kaikki olisi selvää sekä hallinnassa. Tekoäly heijastaa sille syötetyn datan asenteita ja asemaa, josta muodostuu haaste, koska tekoälyn kehitystä hallitsevat valkoiset miehet. Esimerkiksi tekoäly saattaa suositella naiselle alempipalkkaista työtä, tai samasta työstä vähemmän palkkaa, kuin miehelle. Tämä ilmiö esiintyy myös muun muassa autojen kehittämisessä, joka on johtanut siihen, että naisten riski loukkaantua kolareissa on miestä suurempi. Toisena esimerkkinä voidaan pitää kasvojen tunnistusta, jossa on havaittu joidenkin algoritmien tunnistavan paremmin valkoihaisia, koska tekoäly on opetettu valkoihisten kuvia käyttäen. Syrjivien asenteiden vaikutukset yhteiskunnan kehitykseen ovat

haitallisia, ja ne aiheuttavat ihmisten eriarvoista kohtelua. (Toivonen 2023, 19–20; Yhdenvertaisuusvaltuutettu.)

Disinformaatio sekä erinäiset huijaukset ovat jo suuria uhkia yhteiskunnalle. Sosiaalinen media on luonut niille otollisen alustan levitä sekä toteuttaa huijauksia ja näiden toteuttamiseen voidaan myös hyödyntää tekoälyä. Tekoälyllä muokattujen tekstien, kuvien ja videoiden tunnistaminen on haastavaa, joka vaikuttaa luotettavan tiedon löytämisen. Näitä manipulointikeinoja hyödyntämällä voidaan muun muassa ajaa tiettyä poliittista agendaa tai käyttää sitä rikollisiin tarkoituksiin. Yleisimpiä tekoälyä hyödyntäviä huijauksia ovat äänenkloonaus, deepfake-videot (syvävääreännökset), chatbot-huijaukset ja sosiaalinen manipulointi. Esimerkiksi äänenkloonaus huijaukseen tarvitaan vain lyhyt nauhoite, josta saadaan tekoälyä hyödyntämällä luotua hyvinkin aidon kuuloista ääntä, joka saadaan sanomaan, mitä ikinä halutaan. Tämä yhdistettynä nopeaa toimintaa vaativaan viestiin on uhrin vaikea tunnistaa huijausta. (F-Secure 2024.)

3 Generatiivinen tekoäly

Generatiivinen tekoäly on tekoälyn omana erikoisalueensa, jolla pystytään luomaan esimerkiksi tekstiä, kuvia, videoita tai musiikkia. Sen avulla voidaan luoda novelli, joka perustuu tietyn kirjailin tyyliin, luoda realistinen kuva ihmisestä, jota ei ole olemassa, säveltää sinfonia kuuluisan säveltäjän tyyliin tai luoda video yksinkertaisen tekstin pohjalta. Generatiivinen tekoäly toimii koneoppimisen periaatteilla, eli ne kykenevät oppivat datasta. Perinteistä koneoppimisesta poiketen generatiivinen tekoäly kuitenkin kykenee luomaa uusia datainstansseja, jotka jäljittelevät syötettyjentietojen ominaisuuksia. (SAP.)

Generatiivisen tekoälyn nopean kehityksen takia on markkinoille tullut useita generatiivisen tekoälyn sovelluksia. ChatGPT, DALL-E ja Github Copilot ovat hyvin tunnettuja sovelluksia ja niiden käyttö on jo runsasta. Näillä generatiivisen tekoälyn sovelluksilla voidaan luoda tekstiä, koodikieltä ja kuvia. Myös isot teknologiayritykset kuten Microsoft on tuonut markkinoille omat generatiivista tekoälyä hyödyntäviä sovelluksia, kuten Copilot. Generatiivista tekoälyä hyödyntäviä sovelluksia tulee jatkuvasti lisää ja niiden on jo nyt huomattu vaikuttavan työntekoon. (Stallbaumer 2023; Brynjolfsson ym. 2023, 4.)

3.1 GPT

GPT tulee sanoista ”Generative Pre-trained Transformer”, joka on OpenAI tutkimuslaboratorion kehittämä generatiivinen malli. Siitä löytyy jo useita eriversioita kuten GPT-3.5 ja GPT-4. Käytettävän mallin mukaan se pystyy ottamaan vastaan tekstiä ja kuvia ja tuottamaan niistä uusia tekstejä. GPT-mallit ovat yleiskäyttöisiä kielen prosessointimalleja, joiden käytön kuka tahansa voi aloittaa. Näiden mallien käyttäminen on helppo omaksua ja niiden käyttämiseen ei tarvita syvällistä ohjelmointitaitoa. (OpenAI.)

GPT-mallit ovat pääpiirteittäin samanlaisia. Suurimpana erona eri versioilla on koulutukseen käytetty data. GPT-3.5 mallin koulutukseen uusimmat datat ovat helmikuulta 2021, kun taas GPT-4 mallin uusimmat datat ovat huhtikuulta 2023. GPT-4 mallilla on myös muita ominaisuuksia, jota GPT-3.5-mallista ei löydy kuten kuvan vastaanottaminen ja suuremmat datan sisään ja ulostulo määrät. GPT-malleissa käytetään taustalla tilastollista mallia, jolla lasketaan todennäköisyyksiä eri sanojen sekvenssien välillä. Käytössä se siis arvaa lauseeseen seuraavaksi parhaiten sopivan sanan. (OpenAI.)

ChatGPT on OpenAI:n julkaisema verkkosovellus, joka hyödyntää GPT-3.5-mallia. Sovellus on chatbot tyyppinen ratkaisu, jossa voidaan esittää kysymyksiä GPT mallille. ChatGPT:stä löytyy myös maksullinen versio, joka käyttää GPT-4-mallia. ChatGPT:n ja GPT-mallien käytössä nähdään paljon hyvää, mutta myös negatiivisia asioita. Esimerkiksi mallit

saattavat tuottaa väärällä tavalla painotettua tai vääristynyttä tietoa. Koska GPT-mallit ovat koulutettu ihmisten kirjoittamilla teksteillä, käytetystä datasta voi muodostua epätasapainoinen kuva esimerkiksi ikäluokkien tai sukupuolten suhteen. ChatGPT:n ja GPT-mallien nähdään myös vaikuttavan negatiivisesti työpaikkojen määrään sekä tiettyihin ammattiryhmiin. On kuitenkin hyvä muistaa, että uudet teknologiat myös synnyttävät uusia työpaikkoja ja ammatteja. Esimerkiksi GPT-mallit tarvitsevat toimiakseen kehoitesuunnittelua, mallien oikein toimivuuden varmistamiseksi. Lisäksi GPT-mallien hyödyntämismahdollisuudet ovat laajat ja tutkimus vasta alussa. (OpenAI; Kolari & Kallio 2023, 43–47.)

3.2 Kehotesuunnittelu

Kielimalleja käyttäviä tekoälysovelluksia, kuten GPT-malleja ohjataan kehoitteiden (prompt) avulla. Käytettävät kehoitteet koostuvat tyypillisesti luonnollisella kielellä kirjoitetuista ilmaisuvoimaisista lauseista, jonka perusteella tekoälysovellus suorittaa sille annetun tehtävän. Suoritettavan tehtävän onnistuminen riippuu paljon annetun kehoitteen laadusta, sekä vuorovaikutuksesta tekoälysovelluksen kanssa. Kokeilevalla kehoitteiden käyttämisellä päästään alkuun, mutta parempien lopputulosten saavuttamiseksi tarvitaan systemaattista ja suunnitelmallisempia lähestymistapoja. Tekstistä tekstiä kehoitteiden lisäksi on olemassa kehoitteet muun muassa tekstistä kuvaksi ja tekstistä ääneksi. (Arvola 2023.)

Käytettävät kehoitteet voidaan jakaa manuaalisiin ja automaattisiin kehoitteisiin. Manuaaliset kehoitteet ovat yleensä ihmisen luomiomina. Näiden kehoitteiden avulla mallille kerrotaan, minkä tyyppiseen tietoon sen tulee keskittyä ja miten annettua tehtävää tulisi lähestyä. Luotettavan ja laadukkaan kehoitteen suunnittelu vaatii syvää osaamista sekä tietoa. Lisäksi tarvitaan aikaa, koska pienikin muutos kehoitteessa voi saada suuria muutoksia aikaan mallin toiminnassa. Manuaalisien kehoitteiden haasteitten takia rinnalle on kehitetty automaattisia kehoitteita. Automaattiset kehoitteet ovat yleistyneet niiden tehokkuuden takia. Niiden tehokkuus kylläkin riippuu käytetystä algoritmista ja mallin representaatiosta. Automaattiset mallit ovat yleisesti joko erilliskehoitteita tai jatkuvia kehoitteita. Erilliskehoitteissa turvautaan ennalta määrättyihin kategorioihin vastausten luomiseen, kun taas jatkuvissa kehoitteet harkitsevat olemassa olevan kehoitteen avulla kontekstin ja pyrkivät sitä hyödyntäen luomaan vastauksen. (Wang ym. 2021.)

3.3 Omistajuus

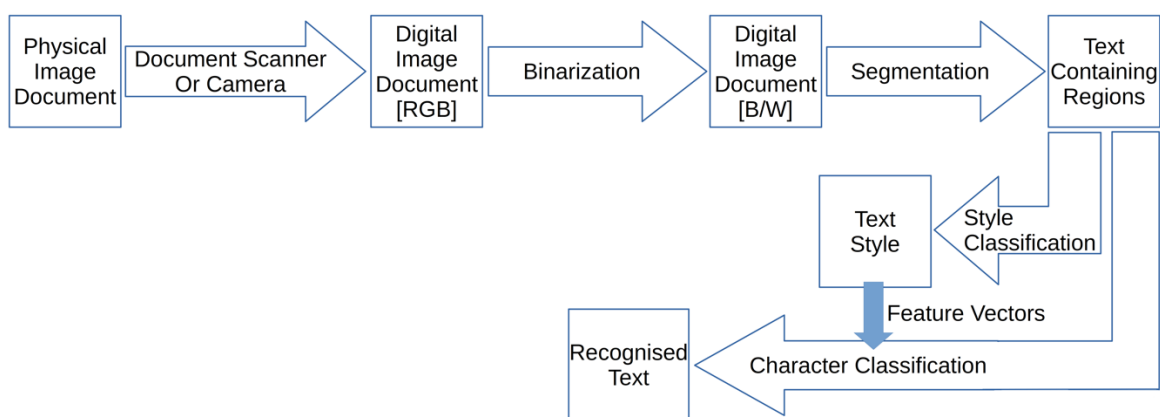
Generatiivisen tekoälyn luomaan sisältöön liittyy ongelma liittyen tuotoksen omistusoikeuksiin. Esimerkiksi perinteisessä mainonnassa käytetyt tekstit, kuvat ja videot ovat olleet tekijänoikeuksien piirissä ja niiden käytöstä on sovittu esimerkiksi asiakkaan ja mainostoimiston välillä. Tekoälyn luomien sisältöjen kohdalla tämä sama järjestelmä ei päde, koska

vakiintuneen käsityksen mukaan tekijänoikeuksia ei voida myöntää ilman luonnollista henkilöä tekijänä. Tämä johtaa siihen, että tekoälyllä luodut tekstit, kuvat ja videot eivät ole tekijänoikeuksien piirissä, jolloin kyseinen sisältö on kaikkien vapaassa käytössä. Sisältö, johon on käytetty ihmisen luomaa sisältöä, sekä tekoälyllä luotua materiaali tulee dokumentoida siten, että siitä selviää, mikä on ihmisen tekemää ja mikä luotu tekoälyllä. Dokumentoinnin perustella voidaan määritellä mitkä osat sisällöstä ovat tekijänoikeuksien piirissä ja mitkä eivät. (Ahava 2023.)

4 Tekstintunnistus

Tekstintunnistuksesta käytetään lyhennettä OCR, joka on lyhenne sanoista “Optical Character Recognition”. Se on optista merkkientunnistamista, joka mahdollistaa käsin kirjoitetun ja painetun tekstin muuttamisen koneluettavaan muotoon. OCR hyödyntämällä kuvista tunnistettua tekstiä voidaan jatkossa työstää koneita hyödyntäen. Esimerkiksi, jos arkistoitavat dokumentit ovat vain kuvatiedostoja ei niistä pysty tekemään tekstihakuja tai muokkaamaan tekstiä. OCR tekniikkaa hyödyntäen dokumenteista saadaan tallennettua tekstit, jolloin taas teksti on muokattavissa, siitä voidaan etsiä tietoa ja laskea kirjaimia ja sanoja. (AWS 2024.)

4.1 Tekstintunnistamisen vaiheet



Kuvio 3. Tekstintunnistamisen vaiheet (DrMax 2022)

Kuviossa 3 on kuvattuna tekstintunnistamisen vaiheet. OCR tekniikan ensimmäisessä vaiheessa fyysinen dokumentti tulee saada digitaaliseen muotoon ottamalla dokumentista kuva tai skannaamalla se digitaaliseen muotoon. Dokumentin digitoimisen jälkeen kuvalle voidaan tehdä esikäsittelyä. Tätä on muun muassa kontrastin säätäminen, kohinan vähentäminen ja kuvan suoristaminen. Näillä toimenpiteillä dokumentissa olevia merkkien tunnistamisen tarkkuutta saadaan parannettua. Tämän jälkeen digitaalisesta dokumentista tehdään koneellisesti mustavalkoinen versio. Saatua mustavalkoista kuvaa analysoidaan valo- ja tumma-alueiden suhteen, jossa tummat alueet tunnistetaan merkeiksi ja vaaleat alueet taustaksi. Tämän jälkeen tummat alueet tunnistetaan numeroiksi ja aakkosiksi. Tämä vaihe voidaan tehdä, joko merkki, sana tai tekstilohko kerrallaan ja siihen voidaan käyttää kuvio- tunnistus- tai piirteentunnistusalgoritmia. (DrMax 2022.)

Kuvion tunnistusalgoritmissa sovellukselle on syötetty esimerkki tekstejä eri fonteista ja niiden muodoista. Tätä vaihetta voidaan myös kutsua sovelluksen ”opetus” vaiheeksi. Tämän jälkeen tarkasteltavasta dokumentista saatuja merkkejä voidaan vertailla sovellukselle ”opettuihin” merkkeihin. (DrMax 2022.)

Piirteiden tunnistetta hyödyntäessä sovellus soveltaa tiettyjä sääntöjä kirjain ja numeroiden merkkien tunnistamiseen. Näitä piirteitä ovat esimerkiksi kulmien määrä, risteilevät linjat tai kaaret merkissä. Erimerkiksi iso kirjain "A" luetaan kahtena viistona viivana, jotka kohtaavat vaakasuoran viivan keskellä. Tämän jälkeen tunnistettu merkki konvertoidaan sen ASCII-koodi (American Standard Code for Information Interchange) muotoon, jota järjestelmät käyttävät edelleen käsittelyssä. (IBM Cloud Education 2022.)

OCR-ohjelmistoa voidaan myös hyödyntää asiakirjan rakenteen analysointiin. Ohjelmalla voidaan jakaa ladattu dokumentti osiin, kuten tekstilohkoihin, taulukoihin tai kuviin. Yksittäiset rivit voidaan jakaa sanoiksi taikka merkeiksi. Dokumentista saadun tiedon näyttämiseen ja tallentamiseen voidaan esimerkiksi käyttää JSON tietomuotoa. JSON muodossa tieto on ymmärrettävässä muodossa niin ihmiselle kuin myös koneelle. (Tesseract-ocr.)

4.2 Päätelaitteella vai palvelimella

Tekstintunnistusta on mahdollista suorittaa niin käyttäjän päätelaitteella kuin myös palvelimen päässä. Kummassakin toimintamallissa on omat hyötynsä ja haittansa. Sopivan tekstintunnistus työkalun valintaan vaikuttavat sen käyttötarkoitus, tietoturva vaatimukset sekä käytettävissä olevat resurssit. (Peters.)

Käyttäjän päätelaitteella tehtävän tekstintunnistuksen hyötyjä ja haittoja ovat:

- Yksityisyys ja tietoturva: kun tekstintunnistus tehdään käyttäjän päätelaitteella niin tieto myös pysyy käyttäjän päätelaitteella. Tällöin dokumenttia ei lähetetä esimerkiksi kolmannelle osapuolelle analysoitavaksi. Tämä voi olla oleellinen asia, kun käsitellään luottamuksellisia tietoja kuten henkilötietoja.
- Käyttö ilman verkkoyhteyttä: tekstintunnistusta on mahdollista tehdä käyttäjän päätelaitteella ilman verkkoyhteyttä, jos käyttäjän käyttämä sovellus löytyy käyttäjän päätelaitteelta.
- Vähentää tietoliikennettä: kun tekstintunnistus tehdään käyttäjän päätelaitteella niin silloin kuvaa ei tarvitse lähettää taustapalvelulle käsiteltäväksi. Tämä vähentää tietoliikennettä käyttäjän ja palvelimen välillä.
- Latenssi: käyttäjän päätelaitteella tehtävä tekstintunnistus voi lisätä sovelluksen viiveaikaa. Tämä voi korostua varsinkin heikompi tehoisilla laitteilla.
- Laite yhteensopivuus: käyttäjillä voi olla erilaisia laitteita ja sovelluksen toimivuus voi olla vaikea todentaa niillä kaikilla. (Peters.)

Palvelimen päässä tehtävän tekstintunnistuksen hyötyjä ja haittoja ovat:

- Yhdenmukaisuus: palvelimen päässä tehtävän tekstintunnistus ei tule laite eroista johtuvia poikkeamia tuloksiin.
- Suorituskyky: palvelin päässä laitteiston tehoja voidaan helposti nostaa ja laskea tarpeen mukaan. Tällä saadaan aikaiseksi tarkempia ja luotettavampia tuloksia.
- Ylläpidettävyys: sovelluspäivitysten yhteyksissä laite testausta ei tarvitse tehdä usealla laitteella.
- Tietoliikenne: tekstintunnistuksen toteuttaminen palvelimen päässä lisää tietoliikennettä, joka voi lisätä kustannuksia.
- Skaalautuvuus: palvelin päässä sovellusta voidaan skaalata horisontaalisesti käsittelemään useampi kutsuja samanaikaisesti. (Peters.)

4.3 Palvelut ja ohjelmistot

Nykypäivänä tekstintunnistuksen suorittamiseen löytyy useita palveluita sekä ohjelmistoja. Palvelujen ja ohjelmistot hyödyntäminen mahdollistavat helpomman OCR:n käyttöönoton sekä soveltamisen. Palveluja ja ohjelmistoja hyödyntämällä voidaan säästää aikaa ja keskittyä muihin sovelluksen keskeisiin toimintoihin. Myös jokaiselta suurimmalta pilvipalvelun tuottajalta löytyy omat tekstintunnistus palvelunsa. Näitä palveluita ovat muun muassa Google Cloud Vision OCR, Amazonilla Textract ja Microsoft Azure OCR. Näiden lisäksi löytyy myös useita muita maksullisia palveluntarjoajia.

Tekstintunnistukseen löytyy myös avoimen lähdekoodin projekteja. Näistä yksi käytetyimmistä on Tesseract, jonka on alun perin kehittänyt Hewlett-Packard. Tesseract tukee useita ohjelmistokieliä ja tätä kautta se on yksi keskeisimmistä tekstintunnistus työkaluista. (Tesseract-ocr.)

4.4 Käyttökohteita



Kuva 1. Tekstintunnistusta voidaan käyttää rekisterinumeron tunnistamiseen

OCR käytetään jo useilla toimialoilla muun muassa toiminnan automatisointiin, käyttökokemuksen parantamiseen sekä tiedon lukemiseen ja tallentamiseen:

- Kommunikaatio: OCR hyödyntämällä voidaan esimerkiksi kääntää kirjoitettua tekstiä reaaliaikaisesti toiselle kielelle. Tämä parantaa ihmisten välistä kommunikaatiota, vaikka yhteistä kieltä ei aina löytyisikään.
- Turismi: Henkilödokumenttien automaattista tunnistamista voidaan hyödyntää esimerkiksi hotelliin tai lennolle kirjautumisen yhteydessä.
- Logistiikka: Paketeissa ja kirjeissä olevien tekstien automaattinen tunnistaminen ja rekisteröinti. OCR- hyödyntämällä mahdolliset virheet vähenevät ja prosessia nopeutettua.
- Tunnistaminen: Yksi yleisimmistä tekstintunnistuksen käyttökohteista on rekisterinumeron tunnistaminen. Tätä käytetään esimerkiksi pysäköintipalveluissa ajoneuvon automaattisessa tunnistamisessa. Kuvassa 1 esimerkki rekisteritunnuksen tunnistamisesta. (Richard 2023.)

5 Data

Data itsessään on merkityksetöntä. Vasta dataa tulkitsemalla sen merkitys ja konteksti voidaan selvittää. Tällöin voidaan vasta puhua datan sisältämästä tiedosta. Dataa voidaan tulkita monin eri tavoin, joka voi johtaa tuloksiin, joita dataa kerätessä ei vielä osattu odottaa. Datan määrä maailmassa kasvaa nopeaa vauhtia. Tallennetun datan määrä sanotaan kaksinkertaistuvan, joka toinen vuosi. Esimerkiksi vuonna 2022 arkistoiitiin noin 3,5 tsettatavua dataa ja arvioiden mukaan 2030 arkistoitavan datan määrä olisi 18 tsettatavua. 1 tsettatavu muunnettuna gigatavuihin on 1 000 000 000 000 gigatavua. Tämä tallennettu data pitää sisällään muun muassa tekstistä ja symboleista koostuvia dokumentteja, kuvia, videoita, taulukoita ja tietokantoihin kerättyjä tietoja. (Heiskanen 2023; AVOINDATA.FI 2023.)

5.1 Struktuurit



Kuvio 4 Datan pääluokat

Tallennettu data voidaan jakaa kuvion 4 mukaisesti kolmeen pääluokkaan strukturoitu, epästrukturoitu ja puolistrukturoitu data.

Dataa pidetään strukturoituna, kun sille on olemassa selkeä tietomalli. Tietomalli kuvaa, miten dataa voidaan käsitellä, tallentaa ja käyttää. Tietomalli on taulukkojärjestelmä, jossa rivireillä ja sarakkeilla on määritelty suhde. Tietomallissa jokainen kenttä on yksilöllinen ja siihen voidaan viitata erikseen tai tietoja voidaan yhdistää muiden kenttien kanssa. Strukturoidun datan käsittely on tehokasta, joka mahdollistaa datan helpon analysoinnin. Datan tallentaminen strukturoidussa muodossa pidetään perinteisenä tiedon tallennus tapana, sillä jo varhaisimmat tiedonhallintajärjestelmät pystyivät käsittelemään, tallentamaan ja käyttämään strukturoitua dataa. Esimerkkejä strukturoidusta datasta ovat SQL-tietokannat ja Excel-tiedostot. Näissä kummassakin on selkeästi ennalta määritelty rakenne, jossa rivit ja sarakkeet voidaan lajitella. (BIS, Inc.)

Epästrukturoidussa datassa tieto ei ole ennalta määritellyssä tietomallissa, formaatissa tai skeemassa. Epästrukturoitu data pitää yleensä sisällään paljon tekstiä, jossa voi lisäksi olla osioita, jotka sisältävät numeroita, päivämääriä tai muita lukemia. Tämä tekee datasta epäjohdonmukaista ja epäselvää, jolloin datan hallinta ja analysointi on haastavampaa verrattuna strukturoituun dataan. Lisäksi epästrukturoitu tieto voi olla kuva-, ääni- ja tiedostomuodossa. Epästrukturoitu data voidaan tallentaa ei-relaationaalisiin tietokantoihin (NoSQL). Epästrukturoidun tiedon käsittely on yrityksille erityisen tärkeää, koska 80–90 % tiedosta on epästrukturoitua. (BIS, Inc.)

Puolistrukturoitu data on strukturoitua dataa, joka ei noudata relaatiotietokantojen muodollista rakennetta. Puolistrukturoidun datan rakenteessa on kuitenkin tyypimäisiä merkkejä datan sisällöstä. Puolistrukturoitu data ei ole yhtä helppoa analysoida kuin mitä strukturoitua dataa on, mutta se on kuitenkin helpompi käsitellä ja analysoida, kuin strukturoimatonta data. Yleisiä puolistrukturoitujen tiedonmalleja ovat JSON sekä XML. (BIS, Inc.)

5.2 Datan laatu

Datan laatua voidaan tarkastella monesta näkökulmasta. Se tarkoittaa kerätyn datan tarkkuutta, johdonmukaisuutta, täydellisyyttä, luotettavuutta, säilytystä sekä sen liittyvyyttä tiettyyn kontekstiin. Korkealaatuinen data on tärkeää esimerkiksi yrityksen päätöksenteossa, analyysien tekemisessä sekä palveluiden ja strategioiden kehittämisessä. Datan laatuun vaikuttaa useita tekijöitä, kuten datan kerääminen, syöttäminen, säilytys sekä integraatiot. (Jones 2023.)

Datan laatua voidaan tarkastella kuuden peruspilarin kautta. Nämä peruspilarit ovat:

- **Tarkkuus:** tällä tarkoitetaan sitä, että kuinka hyvin data vastaa todellisia arvoja. Tarkkuuden varmistaminen vaatii virheiden tunnistamista ja niiden korjaamista. Tarkkuutta voidaan parantaa esimerkiksi tekemällä datan validointi sääntöjä.
- **Täydellisyys:** sillä tarkastellaan sitä, että sisältääkö data kaikki ne tiedot ilman puuttuvia arvoja. Täydellinen datasetti mahdollistaa tarkemmat analyysit. Datan täydellisyyttä voidaan parantaa puuttuvien tietojen täydentämisellä, useiden tietolähteiden tai ulkoisten lähteiden käyttämisellä.
- **Ajantasaisuus:** tällä tarkastellaan datan ajantasaisuutta. Vanhentunut data voi johtaa virheellisiin päätöksiin ja analyysihin.
- **Yhdenmukaisuus:** mittaa sitä, kuinka hyvin datasetit ovat yhtenäisiä sekä yhteensopivia toistensa kanssa. Datasettien yhteen sopivuutta keskenään voidaan parantaa datan standardoimisella.
- **Yksilöllisyys:** datassa tiedot tulisi olla yksilöllisiä ja duplikaatti tiedot tulisi poistaa.

- Yksityiskohtaisuus ja relevanttius: näissä tarkastellaan kuinka yksityiskohtaista ja relevanttia kerätty data on. Liian yksityiskohtaisesti kerätty data voi johtaa liialliseen monimutkaisuuteen datan keräämisessä, kun taas riittämätön voi hankaloittaa analyysien tekemistä. (Jones 2023.)

Puutteellinen datan laatu voi johtua monista seikoista. Yleisimmät ongelmat johtuvat datan konvertoimisesta muodosta toiseen. Tiedon muuttaessa muotoa voidaan ajautua virhetilanteeseen, jossa data korruptoitua ja siitä tulee käyttökelvotonta. Puutteellinen datan laatu voi myös johtua datan keräämisen epäjohdonmukaisuudesta, datan migraatiosta, datan rappeutumisesta tai monistetusta datasta. Huono datan laatu saattaa johtaa yritystä muun muassa rahan menetykseen, tehottomuuteen, mainehaittoihin sekä menetettyihin mahdollisuuksiin. (Foote 2023.)

6 Toteutus

Tämän kehittämishankkeen toteutusvaiheessa tehtiin konseptitodistuksen malli. Kyseisestä mallista haluttiin tehdä helppokäyttöinen, joten työssä päädyttiin luomaan käyttöliittymä. Sovellukselle haluttiin luoda käyttöliittymä, jonka avulla sovelluksen tehtäviä toiminnallisuuksia voitaisiin helposti kokeilla sekä tehdä visuaalisia ja selkeitä esityksiä. Lisäksi käyttöliittymälle luodulla toiminnallisuudella voidaan simuloida loppukäyttäjän toimintaa, sekä nopeuttamaan myöhemmissä vaiheissa tehtäviä testejä. Tällä lähestymistavalla lisäksi ajateltiin, että saadaan parempi ymmärrys toteutuksesta tehdyn konseptin hyödyntämisestä todellisissa sovelluksissa. Sovellukselle luotiin versiohallinta GitHub:iin. Tällä varmistettiin tehtävän työn säilyvyys, sekä mahdollisuus tehdä sovelluskehitystä eri kehityskoneilla. Tässä luvussa kuvataan työn toteutuksen kulkua sen eri vaiheissa, sekä toteutetaan testitapauksia luotua toteutusta hyödyntäen.

6.1 Työn toteutuksen vaiheet

Tämän kehitystyön pohjaksi valittiin Next.js verkkosovelluskehitys, jotta kehitystyössä päästiin hyvin liikkeelle. Se edesauttoi kehitysympäristön nopean pystyttämisen ja ympäristön pyörittämisen kehityskoneella. Lisäksi Next.js:n käyttöönotto kehitystyössä toi mahdollisuuden tutustua sen käyttöön yleisesti projektityössä. Tämän työn kannalta Next.js ei osoittautunut oleelliseksi asiaksi, vaan sen sijaan olisi voitu käyttää mitä muuta tahansa sovelluskehitystä tai jopa rakentaa sovellus tyhjästä. Next.js pitää sisällään TypeScript työkalun, jolla voidaan parantaa koodin luettavuutta ja vähentää virheiden mahdollisuuksia tehtävässä toteutuksessa. (Vercel, Inc 2024)

```

interface Page {
  blocks: Block[] | null;
  confidence: number;
  lines: Line[];
  oem: string;
  osd: string;
  paragraphs: Paragraph[];
  psm: string;
  symbols: Symbol[];
  text: string;
  version: string;
  words: Word[];
  hocr: string | null;
  tsv: string | null;
  box: string | null;
  unlv: string | null;
  sd: string | null;
  imageColor: string | null;
  imageGrey: string | null;
  imageBinary: string | null;
  rotateRadians: number | null;
  pdf: number[] | null;
  debug: string | null;
}

interface Line {
  words: Word[];
  text: string;
  confidence: number;
  baseline: Baseline;
  bbox: Bbox;
  paragraph: Paragraph;
  block: Block;
  page: Page;
  symbols: Symbol[];
}

```

Kuva 2. Tesseract.js vastauksen tyyppitykset

Sovelluksen ensimmäiseksi ominaisuudeksi toteutettiin tekstintunnistus kuvasta. Tekstintunnistus toteutettiin tesseract.js JavaScript kirjastoa hyödyntäen. Tesseract.js valikoitui käytettäväksi tekstintunnistamisen tavaksi, sen helpon käyttöönoton Nextjs:illä luodussa sovelluspohjassa, sekä tesseract.js:ssä käytettävän kehityskielen takia.

Kuvassa 2 on tesseract.js kirjaston tekstintunnistuksesta saatavan JSON vastauksen tyyppitykset. Lisäksi siinä on nähtävissä, että Tesseract.js tuottaa kuvasta tunnistetusta tiedosta puolistrukturoidun JSON-vastauksen. Saatu JSON-vastaus pitää sisällä paljon informaatiota, joka on jaoteltu erinäisiin objekteihin. Näistä oleellisimpina objekteina tämän työn osalta voidaan pitää seuraavia:

- text: sisältää koko tunnistetun tekstin merkkijono (string) muodossa
- confidence: tunnistuksen luottamusarvo numeraalisessa (number) muodossa
- lines: lista objekteja, joka sisältää muun muassa text ja confidence objectit per tunnistettu rivi

Työssä hyödynnettiin tesseract.js JSON vastauksesta löytyvää text objektia. Tämän työn osalta lines objektiin listatut tekstit per rivi tiedot eivät tuoneet lisäarvoa työn seuraavien vaiheiden kannalta.

```
1  import { createWorker } from 'tesseract.js';
2
3  export const readTextFromImage = async (
4    file: File,
5  ): Promise<{ text: string; confidence: number }> => {
6    const worker = await createWorker('fin');
7    const ret = await worker.recognize(file);
8
9    await worker.terminate();
10   return { text: ret.data.text, confidence: ret.data.confidence };
11  };
```

Kuva 3. Tesseract.js kirjaston kutsuminen

Tesseract.js kirjaston kutsuminen toteutettiin tesseract.js:n dokumentaatiosta löytyvän esimerkin perusteella. Kuvassa 3 näkyy funktio, jolla käyttöliittymässä lisätty kuva lähetetään tesseract.js kirjastolle tekstin tunnistukseen. Tekstintunnista varten luodaan worker, jonka parametriksi tässä tapauksessa on määritelty 'fin'. Kyseinen parametri on tunnistuksessa käytettävä kieli ja tässä tapauksessa se on Suomi. Suomen kieli valikoitui käytettäväksi kieleksi, koska tunnistettavat kuvat pitävät sisällään suomenkielisiä sanoja, joissa esiintyy esimerkiksi Ää, Öö ja Åå. Englannin kieltä tunnistuskielenä käytettäessä nämä edellä mainitut esimerkit tunnistettaisiin olevan Aa tai Oo. Tunnistuksen jälkeen käyttöliittymälle palautettiin saadun vastauksen text ja confidence objektit jatkokäytön ja tarkempaa tarkastelua varten. (Tesseract-ocr.)

```
node.js v   
1 import OpenAI from "openai";  
2  
3 const openai = new OpenAI();  
4  
5 async function main() {  
6   const completion = await openai.chat.completions.create({  
7     messages: [  
8       {  
9         role: "system",  
10        content: "You are a helpful assistant designed to output JSON.",  
11      },  
12      { role: "user", content: "Who won the world series in 2020?" },  
13    ],  
14    model: "gpt-3.5-turbo-0125",  
15    response_format: { type: "json_object" },  
16  });  
17  console.log(completion.choices[0].message.content);  
18 }  
19  
20 main();
```

Kuva 4. OpenAI:n rajapinta kutsusta

Sovelluksen toisena ominaisuutena toteutettiin generatiivisen tekoälyn rajapintakutsu. Tässä työssä käytettäväksi rajapinnaksi valikoitui OpenAI:n tarjoama GPT rajapinta. Kyseisen rajapinnan käyttöön päädyttiin sen helpon käyttöönoton ja yleisen tunnettavuuden vuoksi. Tämän vaiheen toteuttamiseen hyödynnettiin kuvan 4:n mukaista OpenAI:n dokumentaatiosta löytyvää esimerkkiä rajapintakutsun toteuttamiseen. Huomion arvoista on, että kutsujen tekeminen vaatii rekisteröinnin OpenAI:n palveluihin ja palvelussa tulee luoda Api key. Tätä luotua Api key:tä tulee käyttää omassa sovelluksessa kutsuja tehdessä. Rajapinnan kutsumista varten luotiin kehote. Kehotteessa määriteltiin, kuinka GPT-mallin tulisi toimia tilanteessa ja millainen vastauksen tulisi olla. Kuvan 4 esimerkissä GPT-mallille kerrottiin sen olevan avulias assistentti, joka tuottaa JSON vastauksen. Kehotteessa määriteltiin myös käyttäjä, joka esitti kysymyksen, johon vastaus haluttiin saada. Tämän lisäksi määriteltiin käytettävä GPT-malli sekä vastauksen muoto, jossa vastaus haluttiin. Tässä esimerkki kehotteessa pyydetty vastauksen muoto oli JSON. Tämä oli hyödyllinen tapa saada vastaus datan jatkohyödyntämistä ajatellen. (OpenAI.)

Kokeilu- ja kehitysvaiheessa päätettiin käyttää OpenAI:n GPT-3.5-turbo-mallia. Tähän päädyttiin, koska jokaisen kutsun tekeminen palveluun maksaa ja GPT-3.5-turbon kutsujen hinta on huomattavasti halvempi, kuin mitä GPT-4-mallin kutsuminen on. Tässä vaiheessa työtä GPT-4 mallin kokeilut päätettiin jättää myöhemmälle vaiheelle ja jatkaa kehitystä GPT-3.5-turboa käyttäen.

Kolmannessa vaiheessa työtä tavoitteena oli saada käyttöliittymältä GPT:lle esitettyyn kysymykseen vastaus itse laaditun JSON skeeman muodossa. Tässä vaiheessa tuli päättää millaista kysymystä haluttiin esittää, ja missä muodossa vastauksen tulisi olla. Tässä työssä ja tässä vaiheessa työtä päädyttiin kysymykseen ”Mikä vahinko on sattunut kyseisen kuvauksen perusteella?”.

```
functions: [
  {
    name: 'response_schema',
    description: 'JSON response schema for accident type and description',
    parameters: {
      type: 'object',
      properties: {
        description: {
          type: 'string',
          description: 'Short incident description',
        },
        incidentType: {
          type: 'string',
          enum: ['injury', 'vehicle', 'illness', 'travel', 'luggage'],
          description: 'Incident type',
        },
      },
    },
  },
],
```

Kuva 5. Kehotteeseen luotu skeema vahinkovastauksen saamiseksi

Vastauksen skeema haluttiin pitää yksinkertaisena, jotta luodun kehotteen toimivuus voidaan todeta ja vastaus helposti validoida. Kehotteeseen lisättiin kuvan 5 mukainen ”functions” objekti, jossa skeeman määrytykset yleisesti tehdään. Vastauksessa pyydettiin antamaan lyhyt kuvaus tapahtuneesta, sekä määrittelemään vahinkotyyppi annetun arvolistan perusteella. Vahingon arvolistaan valittiin arvot yleisten vahinkolajien perusteella.

```

const productsSchema = {
  description: {
    type: 'string',
    description: 'Short description of products',
  },
  store: {
    type: 'string',
    description: 'Store name',
  },
  products: {
    type: 'array',
    description: 'List of products',
    items: {
      type: 'object',
      properties: {
        type: {
          type: 'string',
          enum: Object.values(ProductCategory),
          description: 'Product type',
        },
        name: {
          type: 'string',
          description: 'Product name',
        },
        price: {
          type: 'string',
          description: 'Price of product',
        },
        date: {
          type: 'string',
          format: 'date',
          description: 'Date of purchase',
        },
      },
      required: ['type', 'name', 'price', 'date'],
    },
  },
};

```

Kuva 6. Tuotteet-skeema

Työn neljännessä vaiheessa luotiin kuvan 6 mukainen monimutkaisempi skeema. Skeemassa päädyttiin listamaan ostettuja tuotteita. Pohjana skeeman suunnitteluun käytettiin kassakuitissa olevia tietoja. Lisäksi skeemaan haluttiin tuoda kohtia, joita kuitista ei suoraan löydy. Tässä skeemassa tuotteista haluttiin tietää nimi, hinta, ostopäivä ja tuotteen tyyppi. Tuotteen tyyppille on määritelty arvolista, jossa on esitelty erinäisiä tuote kategorioita, kuten "Groceries", "Toys", "Tools" ja "Shoes". Kyseisen tuotekategorialistan luomiseen käytettiin ChatGPT:tä, eikä se perustu minkään yrityksen tai sovelluksen käyttämiin

tuotekategorioidiin. Tuotetietojen lisäksi haluttiin tietää kauppa, josta tuotteet olivat hankittu, sekä lyhyt kuvaus tapauksesta. Tätä skeemaa käytettäessä tulee kuvassa 5 esitetyn kehotteen ”properties” kohta korvata kuvan 6 skeemalla.

The screenshot shows a web form with the following elements:

- Valitse schema**: A section with two radio buttons: "Vahinko" (unselected) and "Tuotteet" (selected).
- Lisää kuva**: A section with a "Choose file" button and the text "No file chosen".
- Kuitista saatu teksti**: A large, empty text input field.
- Lähetä**: A button at the bottom of the form.

Kuva 7. Sovellukseen tehty käyttöliittymä

Sovellukseen tehtyä käyttöliittymää kehitettiin muiden vaiheiden rinnalla. Kuvan 7 mukainen käyttöliittymä haluttiin pitää yksinkertaisena ja tehdä siitä sellainen, että se tukee testitapausten tekemistä. Käyttöliittymällä tehtiin mahdolliseksi valita skeema, jota halutaan testata.

Testitapaukseen oli valittavissa ”Vahinko” tai ”Tuotteet” skeemat. Molemmissa tapauksissa käyttöliittymällä oli tekstikenttä, johon tekstiä voitiin syöttää. Lisäksi, kun oli valittu ”Tuotteet” skeema, käyttöliittymältä löytyi vaihtoehto ”Lisää kuva”.

”Lisää kuva” toimintoa käyttämällä oli mahdollista tunnistaa kuvasta tekstiä, joka tuodaan tekstikenttään. Lähetä nappia klikkaamalla teksti lähetettiin GPT-mallille analysoitavaksi, jonka jälkeen saatu vastaus tuotiin JSON muodossa käyttöliittymälle tarkasteltavaksi.

6.2 Testitapaus 1: Vahingon tunnistaminen vapaasta tekstistä

Ensimmäisessä testitapauksessa kokeiltiin vahinkolajien tunnistamista vapaasta tekstistä. Testi tapauksissa luotiin lyhyitä kuvauksia sattuneista vahingoista, jotka lähetettiin GPT-mallille analysoitavaksi. Testitapauksissa käytettiin kuvan 5 mukaista skeemaa, joka lähetettiin kehotteen mukana. Kun analysointi oli saatu valmiiksi, saatu vastaus tuotiin merkkijonona käyttöliittymälle tarkastelua varten. Tämän vaiheen testitapauksissa käytettiin GPT-3.5-turbo-mallia.

Kuvaile vahinkoa Autosta hajosi rengas ja matka jäi kesken <pre>{ "description": "Autosta hajosi rengas ja matka jäi kesken", "incidentType": "vehicle" }</pre>	Kuvaile vahinkoa Kaaduin pyörällä ja jalka meni poikki <pre>{ "description": "Kaaduin pyörällä ja jalka meni poikki", "incidentType": "injury" }</pre>
Kuvaile vahinkoa Laukku varastettiin matkalla hotellille <pre>{ "description": "Laukku varastettiin matkalla hotellille", "incidentType": "luggage" }</pre>	Kuvaile vahinkoa Minulle tuli kuumetta ja kurkku tuli kipeäksi <pre>{ "description": "Kuume ja kurkkukipu", "incidentType": "illness" }</pre>
Kuvaile vahinkoa Lomamatka keskeytyi sairastumisen takia <pre>{ "description": "Lomamatka keskeytyi sairastumisen takia", "incidentType": "illness" }</pre>	Kuvaile vahinkoa Lomamatka keskeytyi <pre>{ "description": "Lomamatka keskeytyi", "incidentType": "travel" }</pre>

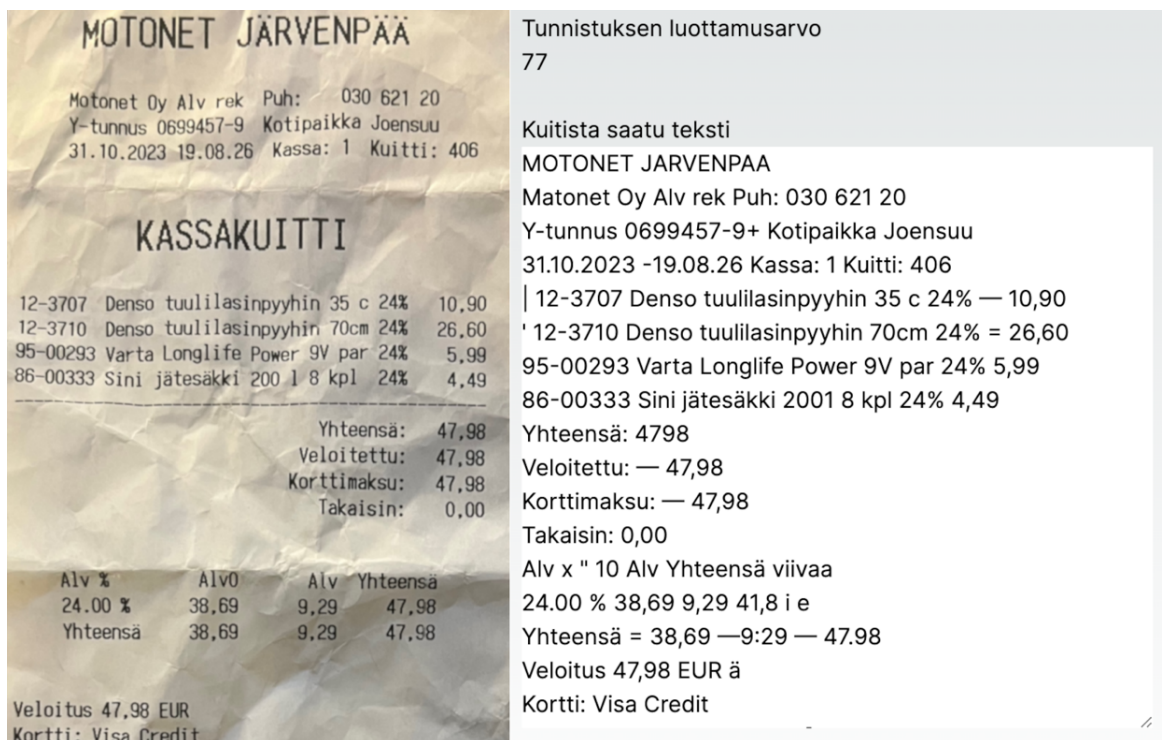
Kuva 8. Esimerkkejä kysymyksistä ja niihin saaduista vastauksista

Kuvassa 8 on nähtävillä esimerkkejä kokeiluista teksteistä ja niihin saaduista vastauksista. Tapauksissa voidaan todeta, että jos syötetyssä tekstissä on jokin selkeä vahinkolajiin viittaava sana, onnistuu tunnistus hyvin sen perusteella. Esimerkiksi, jos tekstissä on sana "Auto" voidaan siitä suoraan olettaa kyseessä olevan ajoneuvovahinko, jolloin palautettava vahinko tyyppin arvo on "vehicle". Haastavimpia ovat tapaukset, joissa syötetty teksti sopii kahteen tai useampaan vahinkolajiin, kuten "Lomamatka keskeytyi sairastumisen takia". Tässä tapauksessa vastaukseksi on saatu vahinko tyyppin arvoksi "illness", jota voidaan pitää oikeana vastauksena. Mutta kyseinen tekstistä voisi myös palauttaa arvon "travel", koska sairastuminen on tapahtunut lomamatkalla.

Saatujen vastausten pohjalta voidaan todeta GPT-3.5-turbo-malli onnistuneen selkeissä ja helppoissa tunnistustehtävissä hyvin. Huomion arvoista on, että vaikka käytetty kieli testitapauksissa on ollut Suomi, ovat vastaukset olleet näissä kokeiluissa kuitenkin osuvia. Tässä vaiheessa ei nähty tarpeellista tehdä kokeiluja GPT-4-mallilla, koska GPT-3.5-turbo saadut vastaukset vastasivat hyvin olettamuksia.

6.3 Testitapaus 2: Tietojen tunnistus kuitista otetusta kuvasta

Toisessa testitapauksessa kokeiltiin kuvan tunnistuksen toimivuutta, sekä siitä saadun tekstin lähettämistä GPT-mallille analysoitavaksi. Kyseisessä testitapauksessa käyttöliittymältä tuli valita käytettäväksi skeemaksi "Tuotteet" ja käytettävän kuvan tulee pitää sisällään tuotteita listattuna, mistä käy ilmi tuotteen nimi ja hinta. Lisäksi kuvasta tulee löytyä päivämäärä, sekä liikkeen nimi, josta tuotteet ovat hankittu.



Kuva 9. Kuitti ja siitä saatu teksti tekstintunnistusta hyödyntäen

Tässä testitapauksessa käytettiin kuvan 9 olevaa kauppakuitista otettua kuvaa. Kuvassa on kuitti Motonetistä tehtyjen ostoksista, jossa näkyy testaukseen vaadittavat tiedot. Käytetty kuva ladattiin sovellukseen, jossa siitä tunnistettiin teksti, joka palautettiin käyttöliittymälle tunnistuksen luottamusarvon kanssa. Tekstintunnistuksesta saadut vastaukset näkyvät kuvassa 9.

Vaikka testitapauksessa käytettävän kuvanlaatu on heikko, voidaan saatua tunnistustulosta pitää hyvänä. Tesseract.js antoi tunnistetulle tekstille luottamusarvoksi 77, jota voidaan myös pitää hyvänä arvona ja yleisellä tasolla tunnistusta voidaan pitää riittävänä. Tunnistetun tekstin joukossa kylläkin löytyy jotain erikoisesti tunnistettuja merkkejä sekä yksittäisiä kirjaimia, jotka on tunnistettu väärin. Esimerkiksi kuitin toisella rivillä oleva "Motonet Oy" on tunnistettu "Matonet Oy", kolmannelle riville on tullut ylimääräinen "+-merkki" ja sana "Kassakuitti" on jäänyt kokonaan tunnistamatta. Työn kannalta kuitenkin kaikki oleellinen tieto on saatu tunnistettu oikein ja tietojen jatkokäyttö on mahdollista.

```

{
  "description": "Products purchased from MOTONET JARVENPAA",
  "store": "MOTONET JARVENPAA",
  "products": [
    {
      "type": "Automotive",
      "name": "Denso tuulilasinpyyhin 35 cm",
      "price": "10.90",
      "date": "2023-10-31"
    },
    {
      "type": "Automotive",
      "name": "Denso tuulilasinpyyhin 70cm",
      "price": "26.60",
      "date": "2023-10-31"
    },
    {
      "type": "Electronics",
      "name": "Varta LongLife Power 9V par",
      "price": "5.99",
      "date": "2023-10-31"
    },
    {
      "type": "Home Appliances",
      "name": "Sini jätösäkki 2001 8 kpl",
      "price": "4.49",
      "date": "2023-10-31"
    }
  ]
}

```

Kuva 10. Testitapaus 2:ssä GPT-mallilta saatu vastaus

Testitapauksen seuraavassa vaiheessa tekstintunnistuksesta saatu teksti lähetettiin GPT-mallille analysoitavaksi. Kuvassa 10 nähdään GPT-mallin luoma JSON vastaus sille annetusta tekstistä. Vastauksessa näkyy lyhyt kuvaus tapauksesta, liikkeen nimi ja listatut tuotteet, jotka on ostettu. Saatujen vastausten vertaamalla kuitista otettuun kuvaan voidaan todeta, että saadut vastaukset vastaavat suurilta osin olettaa. Tuotteiden nimi, hinta ja päivämäärä ovat kohdillaan sekä lisäksi tuotteiden tyypit on saatu valittua sopiviksi. Tässä tapauksessa esimerkiksi tuulilasinpyyhkiöiden tuotetyypiksi on tullut "Automotive", pariston tuotetyypiksi on tullut "Electronics" ja jätösäkkien tuotetyypiksi on tullut "Home Appliances". Jätösäkeille tullut tuotekategoria ei ole täysin oikea ja yksittäin kokeiltuna sen tuotekategoriaksi saatiin muun muassa "Cleaning".

6.4 Testitapaus 3: Tietojen tunnistaminen pdf-tiedostosta

Kolmannessa testi testitapaus oli sama, kuin testitapauksessa kaksi. Poikkeuksena toiseen testitapaukseen tässä käytettiin tiedostona pankin palvelusta saatua pdf-muodossa olevaa kuittia.


```

{
  "description": "Products purchased from Prisma Jumbo",
  "store": "Prisma Jumbo",
  "products": [
    {
      "type": "Groceries",
      "name": "SINIHOMEJUUSTO",
      "price": "2.99",
      "date": "2.3.2024"
    },
    {
      "type": "Groceries",
      "name": "VADELMA MONIVILJAPATUKKA 5X29G",
      "price": "2.79",
      "date": "2.3.2024"
    },
    {
      "type": "Groceries",
      "name": "PUNAJUURIPYÖRYKÄT",
      "price": "3.29",
      "date": "2.3.2024"
    },
    {
      "type": "Groceries",
      "name": "KEVYTMAITO",
      "price": "0.89",
      "date": "2.3.2024"
    },
    {
      "type": "Groceries",
      "name": "MUOVIKASSI PR HARMAA",
      "price": "0.26",
      "date": "2.3.2024"
    }
  ],
}

```

```

{
  "type": "Groceries",
  "name": "NAUTA JAUHELIHA 10%",
  "price": "3.99",
  "date": "2.3.2024"
},
{
  "type": "Health",
  "name": "SANA-SOL MONIVITAMIINI",
  "price": "1.85",
  "date": "2.3.2024"
},
{
  "type": "Groceries",
  "name": "KIRSIKATOMAATTI 250G",
  "price": "1.35",
  "date": "2.3.2024"
},
{
  "type": "Groceries",
  "name": "TUTTIFRUTTI HEDMAK JUOMA",
  "price": "0.45",
  "date": "2.3.2024"
},
{
  "type": "Apparel",
  "name": "JR T-PAITA KUUMAA",
  "price": "26.91",
  "date": "2.3.2024"
}
]

```

Kuva 12. Tuotteiden listaustauksen JSON

Tässä testitapauksessa saamaa tekstiä lähetettiin useamman kerran GPT-mallille. Kuvassa 12 näkyy yksi saaduista vastauksista. Tässä kyseisessä tuotteet nähdään hyvin listattuina halutun skeeman mukaisesti. Lisäksi tuotetyppi, nimi, hinta ja päivämäärä tiedot ovat oikeita. Elintarvikkeet ja muovipussi saivat tässä testitapauksessa tuotetyppi arvoksi "Groceries", monivitaminivalmiste sai arvoksi "Health" ja t-paita arvon "Apparel". Muovipussin kohdalla "Groceries" arvo oli ehkä vähän sinnepäin, mutta voidaan joissain tilanteissa pitää oikeana arvona.

Testitapausta useampaa kertaa kokeillessa saatiin myös toisia tuloksia. Esimerkiksi t-paita ja terveys tuotteet saattoivat, joskus saada myös arvon "Groceries". Tämä osittain johtunee käytettävän kategorialistassa olevien päällekkäisyyksien takia. Tässä testi tapauksessa päädyttiin tekemään kokeilu uudestaan, kun kategorialistauksesta on poistettu "Groceries".

```

{
  "description": "Products purchased from PRISMA JUMBO",
  "store": "HOK-Elanto Liiketoiminta Oy",
  "products": [
    {
      "type": "Dairy",
      "name": "SINIHOMEJUUSTO",
      "price": "2.99",
      "date": "2024-03-02"
    },
    {
      "type": "Snacks",
      "name": "VADELMA MONIVILJAPATUKKA 5X29G",
      "price": "2.79",
      "date": "2024-03-02"
    },
    {
      "type": "Frozen Foods",
      "name": "PUNAJUURIPYÖRYKÄT",
      "price": "3.29",
      "date": "2024-03-02"
    },
    {
      "type": "Dairy",
      "name": "KEVYTMAITO",
      "price": "0.89",
      "date": "2024-03-02"
    },
    {
      "type": "Household",
      "name": "MUOVIKASSI PR HARMAA",
      "price": "0.26",
      "date": "2024-03-02"
    }
  ],
}

{
  "type": "Meat",
  "name": "NAUTA JAUHELIHA 10%",
  "price": "3.99",
  "date": "2024-03-02"
},
{
  "type": "Vitamins",
  "name": "SANA-SOL MONIVITAMIINI",
  "price": "1.85",
  "date": "2024-03-02"
},
{
  "type": "Fruits",
  "name": "KIRSIKKATOMAATTI 250G",
  "price": "1.35",
  "date": "2024-03-02"
},
{
  "type": "Beverages",
  "name": "TUTTIFRUTTI HEDMAK JUOMA",
  "price": "0.45",
  "date": "2024-03-02"
},
{
  "type": "Apparel",
  "name": "JR T-PAITA KUUMAA",
  "price": "26.91",
  "date": "2024-03-02"
}
}

```

Kuva 13. GPT-mallin vastaus ilman 'Groceries' kategoriaa

Kuvassa 13 näkyy GPT:lta saadut vastauksen ilman 'Groceries' tuotekategoriaa. Vastauksista voidaan todeta, että ne saadut kategoriat kuvaukset ovat tarkempia, mitä aikaisemmassa vastauksessa oli saatu.

6.5 Testitapaus 4: GPT-4 mallin tekstin tunnistus kuvasta

Viimeisessä testitapauksessa otettiin käyttöön GPT-4 malli. Koska GPT-4 malli tukee kuvien vastaanottamista, päätettiin kyseinen ominaisuus ottaa sovelluksessa käyttöön. Tämä vaati muutoksia GPT:n rajapinta kutsuun, sekä analysoitavan kuvan muuttamista base64 koodattuun muotoon.

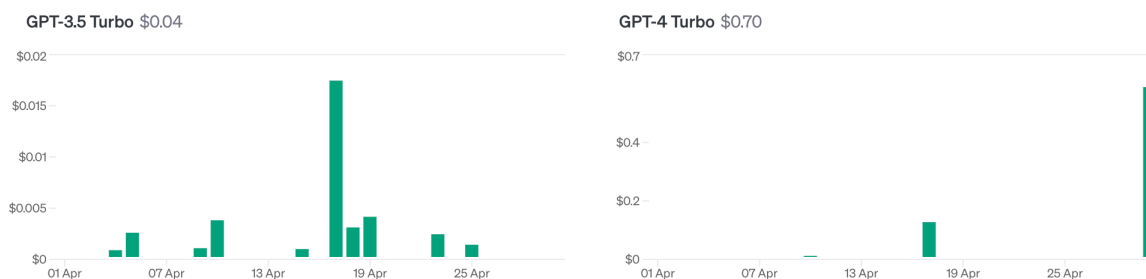
```

model: 'gpt-4-turbo',
messages: [
  { role: 'system', content: 'Provide output in valid JSON from image' },
  {
    role: 'user',
    content: [
      {
        type: 'image_url',
        image_url: {
          url: image,
        },
      },
    ],
  },
],

```

Kuva 14. muutokset GPT rajapinnan kutsuminen kuvan kanssa

Rajapinta kutsuun tehdyt muutokset näkyvät kuvassa 14. Käytettävä GPT-malli on vaihdettu olemaan "gpt-4-turbo", mallia on pyydetty antamaan vastaus JSON muodossa annetusta kuvasta ja käyttäjän mallille lähetettävä sisältö on kuva. Tässä tapauksessa kuva on base64 koodattuna, mutta se voi myös olla suora verkko-osoite kuvatiedostoon.



Kuva 15 GPT-mallien käyttöön menneet kulut

Testivaiheessa käytettiin samaa kuvaa kuin mitä testitapaus 2:ssa käytettiin. Kuva lähetettiin useampaan kertaan GPT-mallille analysoitavaksi ja malli kykeni tuottamaan halutun JSON vastauksen lähetetystä kuvasta. Saadut vastaukset vaihtelivat hieman toisistaan, esimerkiksi paristo sai vaihtelevasti arvoja "Electronics" ja "Batteries". Yleisessä tasolla GPT-malli onnistui annetussa tehtävässä hyvin. On kuitenkin hyvä muistaa, että GPT-4 mallin käyttäminen on huomattavasti kalliimpaa kuin GPT-3.5 mallin. Kuvassa 15 näkyy sovelluksen kehitys- ja testausvaiheen mallien käyttöön menneet kulut. Vaikka GPT-4 malli on vain käytetty muutamia kertoja, on sen kulut moninkertaiset verrattuna GPT-3.5 malliin.

7 Yhteenveto ja pohdinta

Ensimmäinen tutkimuskysymys kehittämishakkeessa oli, miten generatiivista tekoälyä voidaan hyödyntää datan strukturoinnissa. Uusimpia generatiivisen tekoälyn rajapintoja hyödyntämällä generatiiviselta tekoälyltä saatu vastaus voidaan pyytää olemaan JSON muotoa. Lisäksi vastauksen struktuuri voidaan määrittää vastaamaan sitä tietomallia, mitä vastauksen halutaan olevan. Esimerkkejä tämän kaltaisen toteutuksen tekemiseen löytyy generatiivisen tekoälyn dokumentaatiosta. Tässä työssä tehdystä konseptitodistuksen mallista löytyy yhteensä neljä testitapaus esimerkkiä datan strukturoinnista GPT-mallia hyödyntäen. Tehtyjen testitapausten tulosten pohjalta voidaan todeta, että käytetyt GPT-mallit suoriutuvat hyvin ei-strukturoidun tekstidatan strukturoimisessa.

On kuitenkin hyvä ottaa huomioon, että tässä opinnäytetyössä tehdyt konseptitodistus mallia hyödyntävät testitapaukset ovat olleet selkeitä, melko yksinkertaisia ja käytetyt datamallit eivät vastanneet minkään tietyn yrityksen tai sovelluksen käyttämiä datamalleja. Lisäksi on huomioitava, että tehdyissä testitapauksissa kaikki tarvittava tieto oli GPT-mallilla saatavissa datamallin täyttämiseen. Tässä työssä ei tutkittu miten generatiivinen tekoäly käyttäytyy, jos kaikkea tietoa ei ole tekstistä saatavissa.

Toisena tutkimuskysymyksenä oli, miten generatiivisella tekoälyllä voidaan rikastuttaa dataa. Tässä työssä tehdyt kokeilut GPT-malleja hyödyntäen voidaan todeta, että generatiivista tekoälyä voidaan käyttää datan rikastuttamisessa. Tehdyissä testitapauksissa GPT-mallit saivat tunnistettua tuotteiden kategoriat hyvä tasoisesti. Huomattavaa oli kuitenkin se, että käytetyn tuotekategorialistaus oli myös ChatGPT:llä generoitu ja se sisälsi päällekkäisyyksiä. Tämän takia eri kokeilukerroilla GPT-mallilta saaduissa vastauksissa tuotekategorioissa oli vaihtelua.

Datan rikastuttamisessa on tärkeää ottaa huomioon ne arvojoukot, jotka GPT-mallille määritellään. Esimerkiksi tuotekategorialistaus tulisi suunnitella siten, että siinä ei olisi päällekkäisyyksiä. Lisäksi kehotteessa tulisi määritellä millaisia tuotteita kyseiseen kategoriaan kuuluu, joka on erittäin tärkeä osa kehotesuunnittelua. Tässä työssä ei paneuduttu kategorioiden suunnitteluun, vaan keskityttiin tekemään testitapauksia tiedon rikastuttamista varten.

Kolmantena tutkimuskysymyksenä oli, miten tekstintunnistamista voidaan hyödyntää datan strukturoinnin tukena. Tehtyjen testien perusteella voidaan todeta, että kuvasta tekstintunnistus on hyvällä tasolla ja siitä saatua tekstiä voidaan hyödyntää datan jatkokäsittelyssä. Suoraan tekstintunnistussovelluksesta saatua dataa on haastava käsitellä, joten sen jatko-prosessointiin tarvitaan aina, jokin toinen työkalu tai itse kirjoitettu ohjelma.

Testitapaus 4:n pohjalta voidaan todeta, että GPT-4 malli suoriutuu hyvin tekstintunnistamisessa kuvasta ja tätä on mahdollista käyttää tietolähteenä datan strukturoinnissa. Tätä ominaisuutta hyödyntämällä voitiin prosessissa jättää yksi toiminto kokonaan tekemättä. Jos työssä olisi käytetty pelkästään GPT-4 mallia, olisi tesseract.js toteutus voitu jättää tästä työstä kokonaan pois.

Tekoälyn ja siihen liittyvien työkalujen kehitys on tällä hetkellä erittäin nopeaa ja uusia työkaluja tulee koko ajan lisää. Myös tämän työn edetessä asioita ilmeni uusia näkökulmia, kuin mitä työn ensimmäiset lähtökohdat olivat. Esimerkiksi tämän opinnäytetyön aihe ei välttämättä olisi vielä ollut ajankohtainen puolta ennen työn aloittamista. Tämä mielestäni kuvastaa hyvin tekoälyn ja varsinkin generatiivisen tekoälyn nopeaa kehitystä. Useista palveluista ja nopeasta kehityksestä johtuen tässä työssä keskityttiin suorittamaan kokeiluja yhden palveluntarjoajan generatiivisen tekoälyn malleilla. Lisäksi alkuperäisistä suunnitelmista poiketen kuvantunnistus jätettiin pienempään rooliin, koska työssä oleellisempina tarkastelun kohteina olivat datamallien luominen ja datan rikastuttaminen generatiivista tekoälyä hyödyntämällä.

Tekstintunnistamisen hyödyntäminen on jo laajassa käytössä eri toimialoilla ja siihen löytyy useita palveluita ja sovelluksia. Työn edetessä käytettävä tesseract.js kirjasto päivittyi. Vaikka työn tarkoitus ei ollut tehdä vertailuja, niin kirjaston päivityksen yhteydessä huomattiin sen parantunut luotettavuus tunnistuksessa. Ensimmäiset kokeilut kuvassa 9 olevalla kuitilla antoivat huonompia tuloksia, jossa esimerkiksi tunnistetut rivit olivat sekaisin. Uudemmassa versiossa nämä olivat poistuneet ja voitiin todeta, että saatua tekstiä voitiin paremmin hyödyntää seuraavissa vaiheissa.

Konenäköä hyödynnettäessä tulee ottaa huomioon luotettavuusaste. Konenäöllä päästään harvemmin täysin luotettaviin tuloksiin, mutta kuitenkin tarpeeksi hyviin, jotta toimintaa voidaan jatkaa. Saatua vastauksia on myös hyvä validoida käyttäjällä ennen toiminnan jatkamista. Tällöin säästytään väärän informaation lähettämisessä eteenpäin.

Työssä hyödynnetty tesseractjs tarjosi vastauksen puolistrukturoidussa JSON muodossa. Saatua vastausta oli helppo tulkita ja sieltä löytää halutut tiedot jatkokäsittelyä varten. Kyseinen JSON vastaus ei kuitenkaan vastannut sellaisenaan siihen, mitä tässä työssä tavoiteltiin. Esimerkiksi kuvassa 9 olevasta kuitista saadusta vastauksesta tuotteiden listaaminen olisi ollut ohjelmallisesti hankalaa. Tämän lisäksi pelkästään kuvan tunnistusta käyttämällä saatua dataa ei voida saada suoraan haluttuun datamallin muotoon. Dataa ei myöskään saada rikastettua ilman ohjelmallisia toimintoja.

Tässä kehittämistyössä tehtyjen testitapausten mukaan voidaan todeta, että generatiivista tekoälyä ja tekstintunnistusta on mahdollista hyödyntää käyttöliittymän käytön tukena, jossa

loppukäyttäjä täyttää lomaketta. Sen avulla pystytään parantamaan datan laatua ja vähentämään loppukäyttäjän tarvetta syöttää tietoja manuaalisesti. Tässä työssä käytetyt esimerkkitapaukset olivat itse valittuja ja keksittyjä. Lisäksi työn kokeiluissa käytettiin itse luotua skeemaa, joka ei vastannut minkään yrityksen tai sovelluksen käyttämää datamallia.

Generatiivista tekoälyä hyödyntävien lomakesovellusten toiminnallisuutta ajatellen, on hyödyllistä tehdä lisätutkimusta siitä, miten loppukäyttäjät suhtautuvat generatiivisen tekoälyn käyttöön lomakkeiden täyttämässä. Sen lisäksi tarvitaan miettiä, mitä prosesseja tarvitaan, jotta kerättyä tietoa voidaan välittää eteenpäin jatkoprosessointia varten.

Tämän työn pohjalta on jo suunniteltu tehtäväksi diplomityö, jossa tutkitaan generatiivisen tekoälyn hyödyntämistä datan rikastuttamisesta käytännön sovelluskohteessa, käyttäen yrityksen todellista dataa ja datamalleja.

Lähteet

Aditya J, Gandhar K & Vraj S. 2018. Natural Language Processing. International Journal of Computer Sciences and Engineering. Viitattu 3.5.2024. Saatavissa https://www.ijcseonline.org/pub_paper/28-IJCSE-VC-109.pdf

Ahava A. 2023. Generatiivinen tekoäly markkinoinnissa – kuka omistaa tulokset, ja voivatko ne loukata kolmansien oikeuksia? Berggren. Viitattu 30.4.2024. Saatavissa <https://www.berggren.eu/blogi/generatiivinen-tekoaly-markkinoinnissa-kuka-omistaa-tulokset-ja-voivatko-ne-loukata-kolmansien-oikeuksia>

Arvola P. 2023. Tulevaisuuden tieto(palvelu)työ on kehofesuunnittelua. Tampereen yliopisto. Viitattu 29.4.2024. Saatavissa <https://blogs.tuni.fi/wildfire/yleista/tulevaisuuden-tietotyö-on-kehofesuunnittelua/>

AVOINDATA.FI. 2023. Mitä on avoin data?. Digi- ja väestötietovirasto. Viitattu 3.5.2024. Saatavissa <https://www.avoindata.fi/fi/tietoa-avoimesta-datasta/mita-on-avoin-data#avoimen-datan-taustaa>

AWS. 2024. What is OCR (Optical Character Recognition)?. AWS. Viitattu 30.4.2024. Saatavissa <https://aws.amazon.com/what-is/ocr/>

BIS, Inc. Integrate Unstructured Data. BIS, Inc. Viitattu 3.5.2024. Saatavissa <https://www.bisok.com/analytics-and-business-intelligence/unstructured-data/>

Brynjolfsson E, Li D & Raymond LR. 2023. GENERATIVE AI AT WORK. NBER. Viitattu 28.4.2024. Saatavissa https://www.nber.org/system/files/working_papers/w31161/w31161.pdf

CSC. Tekoälyn eettisiä haasteita. CSC – TIETEEN TIETOTEKNIIKAN KESKUS OY. Viitattu 3.5.2024. Saatavissa <https://www.csc.fi/-/tekoalyn-eettisia-haasteita>

DrMax. 2022. How Does Optical Character Recognition Work. Baeldung. Viitattu 30.4.2024. Saatavissa <https://www.baeldung.com/cs/ocr>

Expert.ai Team. 2022. What Is Machine Learning? A Definition. Viitattu 22.4.2024. Saatavissa <https://www.expert.ai/blog/machine-learning-definition/>

Foote KD. 2023. The Impact of Poor Data Quality (and How to Fix It). DATAVERSITY. Viitattu 6.5.2024. Saatavissa <https://www.dataversity.net/the-impact-of-poor-data-quality-and-how-to-fix-it/>

- F-Secure. 2024. Miten äänikloonauus ja tekoälyhuijaukset toimivat?. F-Secure. Viitattu 3.5.2024. Saatavissa <https://www.f-secure.com/fi/articles/ai-scam-calls>
- Hagiwara M. 2021. Real-World Natural Language Processing. Manning Publications. Viitattu 2.5.2024. Saatavissa <https://livebook.manning.com/book/real-world-natural-language-processing>
- Heiskanen M. 2023. ”Datan määrä tuplaantuu noin joka toinen vuosi” – Mihin ihmeeseen kaikki se säilötään?. Tekniikan maailma. Viitattu 3.5.2024. Saatavissa <https://tekniikanmaailma.fi/lehti/11b-2023/mihin-ihmeeseen-sailotaan-kaikki-tama-data/>
- Hevner AR, March ST, Park J & Ram S. 2004. Design Science in Information Systems Research. Viitattu 24.4.2024. Saatavissa https://wise.vub.ac.be/sites/default/files/thesis_info/design_science.pdf.
- Hjelt Y. 2023. Tekoäly voi pian auttaa täyttämään excel-tilukot työpaikalla – nopea kehitys yllätti tutkijankin. Yle. Viitattu 8.5.2024. Saatavissa <https://yle.fi/a/74-20056377>
- IBM Cloud Education. 2022. Optical character recognition (OCR) technology is an efficient business process that saves time, cost and other resources by utilizing automated data extraction and storage capabilities. IBM. Viitattu 30.4.2024. Saatavissa <https://www.ibm.com/blog/optical-character-recognition/>
- Jones E. 2023. 6 Pillars of Data Quality and How to Improve Your Data. IBM. Viitattu 6.5.2024. Saatavissa <https://www.ibm.com/blog/6-pillars-of-data-quality-and-how-to-improve-your-data/>
- Kalla D & Smith N. 2023. Study and Analysis of Chat GPT and its Impact on Different Fields of Study. International Journal of Innovative Science and Research Technology Viitattu 29.4.2024. Saatavissa https://www.researchgate.net/publication/369506216_Study_and_Analysis_of_Chat_GPT_and_its_Impact_on_Different_Fields_of_Study
- Kalmi R. 2021. Tekoälyn uudet kielimallit hämmästyttävät. Viitattu 30.4.2024. Vaasan yliopisto. Saatavissa https://issuu.com/universityofvaasa/docs/vox_1_2021_issuu?fr=sZWEzMTM5Mjg5OTc
- Kolari J & Kallio A. 2023. Tekoäly 123. Jyväskylä: Docendo.
- Margaret Rouse. 2023. Reinforcement Learning. Techopedia. Viitattu 3.5.2024. Saatavissa <https://www.techopedia.com/definition/32055/reinforcement-learning-rl>
- Merilehto A. 2018 Tekoäly matkaopas johtajalle. Helsinki: Alma Talent.

OpenAI. Viitattu 28.4.2024. Saatavissa <https://platform.openai.com/docs/overview>

Peters B. Server Side & Client Side Processing (OCR A Level Computer Science).

SaveMyExams. Viitattu 30.4.2024. Saatavissa <https://www.savemyexams.com/a-level/computer-science/ocr/17/revision-notes/3-exchanging-data/3-4-web-technologies/server-side-and-client-side-processing/>

Richard M. 2023. THE TOP USE CASES OF OPTICAL CHARACTER RECOGNITION (OCR) TECHNOLOGY. ShuftiPro. Viitattu 30.4.2024. Saatavissa

<https://shuftipro.com/blog/the-top-use-cases-of-optical-character-recognition-ocr-technology/>

Samoili, S, López Cobo, M, Gómez, E, De Prato, G, Martínez-Plumed, F, & Delipetrev, B 2020. AI Watch Defining Artificial Intelligence. Viitattu 30.4.2024. Saatavissa

https://publications.jrc.ec.europa.eu/repository/bitstream/JRC118163/jrc118163_ai_watch._defining_artificial_intelligence_1.pdf

SAP. What is generative AI?. SAP. Viitattu 11.5.2024. Saatavissa

<https://www.sap.com/products/artificial-intelligence/what-is-generative-ai.html>

Saumya. 2024. How to Choose the Right Machine Learning Tool. Appypie. Viitattu

11.5.2024. Saatavissa <https://www.appypie.com/blog/how-to-choose-right-machine-learning-tool>

Stallbaumer C. 2023. Introducing Microsoft 365 Copilot – A whole new way to work.

Microsoft. Viitattu 7.8.2023. Saatavissa <https://www.microsoft.com/en-us/microsoft-365/blog/2023/03/16/introducing-microsoft-365-copilot-a-whole-new-way-to-work/>

State of AI in Finland. 2020. Raportti tekoälyn tilasta Suomessa: julkinen sektori loistaa ja yli 1200 yritystä käyttää tekoälyä päivittäin – kasvun jarruina datan ja osaamisen puute.

Teknologiateollisuus. Viitattu 8.5.2024. Saatavissa

<https://teknologiateollisuus.fi/fi/ajankohtaista/tiedote/raportti-tekoalyn-tilasta-suomessa-julkinen-sektori-loistaa-ja-yli-1200>

Tesseract-ocr. Tesseract OCR. Tesseract-ocr. Viitattu 30.4.2024. Saatavissa

<https://tesseract-ocr.github.io/tessdoc/>

The MathWorks, Inc. 2022. Machine Learning with MATLAB. Viitattu 3.5.2024. Saatavissa

<https://se.mathworks.com/content/dam/mathworks/ebook/gated/machine-learning-ebook-all-chapters.pdf>

Toivonen H. 2023. Mitä tekoäly on?. Helsinki: Teos.

Vercel, Inc. 2024. Introduction Wellcome to the Next.js documentation!. Vercer, Inc.
Viitattu 20.4.2024. Saatavissa <https://nextjs.org/docs>

Wang J, Shi E, Yu S, Wu Z, Ma C, Dai H, Yang Q, Kang Y, Wu, Hu H, Yue C, Zhang H,
Liu Y, Li X, Ge B, Zhu D, Yuan Y, Shen D, Liu T & Zhang S. 2021. Prompt engineering for
healthcare: Methodologies and applications. Viitattu 29.4.2024. Saatavissa
<https://arxiv.org/pdf/2304.14670>

Yhdenvertaisuusvaltuutettu. Tekoäly ja yhdenvertaisuus. Yhdenvertaisuusvaltuutettu.
Viitattu 3.5.2024. Saatavissa <https://yhdenvertaisuusvaltuutettu.fi/tekoaly>