

Iina Hyytiäinen

TEKOÄLY JA LUONNOLLISEN KIELEN KÄSITTELYN MIKROPALVELUT NLP CLOUD API:Ä HYÖDYNTÄEN

Opinnäytetyö

Liiketalouden ammattikorkeakoulututkinto

Tietojenkäsittely

2022



**Kaakkois-Suomen
ammattikorkeakoulu**

Tutkintonimike	Tradenomi (AMK)
Tekijä/Tekijät	lina Hyytiäinen
Työn nimi	Tekoäly ja luonnollisen kielen käsittelyn mikropalvelut NLP Cloud API:a hyödyntäen
Toimeksiantaja	Mindhive Oy
Vuosi	2022
Sivut	48 sivua
Työn ohjaaja(t)	Arto Väätäinen

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli tutkia ja testata luonnollisen kielen käsittelyä NLP Cloud API:a hyödyntäen, sekä aloittaa API:n kanssa kommunikoivan komponenttikirjaston luominen toimeksiantajalle luonnollisen kielen käsittelyyn liittyviin tehtäviin asiakasprojektiin valittujen mikropalveluiden pohjalta. Näiden tavoitteiden tarkoituksena oli, että toimeksiantaja voi tarjota opinnäytetyön aikana toteutettuja komponentteja mikropalveluina osana asiakkailleen rakentamiansa sovellusratkaisuja.

Opinnäytetyön teoriaosuudessa keskitytään luonnollisen kielen käsittelyyn tekoälyn osa-alueena, selittäen molempiin liittyviä käsitteitä ja perusteita. Teoriaosuudessa esitellään myös luonnollisen kielen käsittelyyn liittyviä tehtäviä ja käytötapauksia, sekä erilaisia lähestymistapoja ja työkaluja siihen liittyen. Teoriaosuuden lopussa sukellaan vielä hieman tarkemmin API-lähestymistapaan ja esitellään NLP Cloud API.

Käytännön osuudessa esitellään toimeksiantaja ja asiakasprojekti, sekä käydään läpi projektiin valitut ja toteutettavat mikropalvelut. Toteutusvaiheen aikana tehtiin sarja kokeiluja mikropalveluihin liittyen käyttämällä NLP Cloud API:n tarjoamia luonnollisen kielen käsittelyn malleja. Toteutettaviin mikropalveluihin kuului automaattisten yhteenvetojen luominen, kommenttien automaattinen tulkinta, sekä kysymyksiin vastaaminen. Opinnäytetyön aikana aloitettiin komponenttikirjaston luominen näiden mikropalvelujen pohjalta.

Työn tulokset ja johtopäätökset esitellään opinnäytetyön lopussa, ja niistä voidaan todeta, että vaikka osa API:n malleista vaikutti toimivan suhteellisen hyvin perustilanteisiin, niin niiden lisätestaaminen, hienosäätäminen ja jatkokehitys on suositeltavaa.

Asiasanat: tekoäly, luonnollisen kielen käsittely, mikropalvelut, API, koneoppiminen, syväoppiminen, neuroverkot, data, malli, opinnäytetyö

Degree	Bachelor of Business Administration
Author (authors)	Ilina Hyytiäinen
Thesis title	Artificial intelligence and microservices of natural language processing by utilizing the NLP Cloud API
Commissioned by	Mindhive Oy
Time	April 2022
Pages	48 pages
Supervisor	Arto Väätäinen

ABSTRACT

The aim of the thesis was to study and test natural language processing using the NLP Cloud API, and to start implementing the component library that would communicate with the API for tasks related to natural language processing, based on the selected microservices for the client project. The main purpose of these aims was that the thesis commissioner could offer the components implemented during the thesis as microservices as part of the application solutions they build for their customers.

The theoretical part of the thesis focused on natural language processing as a subfield of artificial intelligence, explaining the concepts and fundamentals related to both. The theoretical part also presented the tasks and use cases related to the natural language processing, as well as various approaches and tools related to it. The end of the theoretical part presented the API approach and NLP Cloud API a little more closely.

The commissioner and the client project were introduced in the beginning of the practical part, and after that the microservices selected and implemented for the project were reviewed. During the implementation phase, a series of experiments related to the microservices were performed using natural language processing models provided by the NLP Cloud API. The implemented microservices included the generation of automatic summaries, the automatic interpretation of comments, and the question answering. During the thesis, the implementation of the component library based on these microservices was started.

The results and conclusions of the implementation were presented at the end of the thesis. They showed that although some of the API models seemed to work relatively well in basic situations, their further testing, fine-tuning, and further development must be recommended.

Keywords: artificial intelligence, natural language processing, microservices, API, machine learning, deep learning, artificial neural networks, data, model, thesis

SISÄLLYS

1	JOHDANTO	5
2	TEKOÄLY JA LUONNOLLISEN KIELEN KÄSITTELYN PERUSTEET	6
2.1	Tekoäly	6
2.2	Luonnollisen kielen käsittely	13
2.3	Luonnollisen kielen käsittelyyn liittyviä tehtäviä ja käyttötapauksia.....	18
2.4	Lähestymistavat ja työkalut luonnollisen kielen käsittelyyn	25
2.4.1	Lähestymistavat.....	25
2.4.2	Työkalut	27
2.4.3	API-lähestymistapa ja NLP Cloud API	28
3	MIKROPALVELUJEN TOTEUTUS SOVELLUSTEN TEHOSTAMISEEN	30
3.1	Mindhive Oy.....	30
3.2	Asiakasprojekti.....	31
3.3	Toteutettavat mikropalvelut.....	33
3.4	Toteutus.....	35
4	TULOKSET.....	38
5	PÄÄTÄNTÖ	42
	LÄHTEET.....	44

1 JOHDANTO

Tämän opinnäytetyön aiheena on luonnollisen kielen käsittely tekoälyn osa-alueena. Sekä tekoäly, että luonnollisen kielen käsittely, ovat molemmat tehokkaita työkaluja ongelmien ratkaisemiseen ja uovien ratkaisujen sekä ohjelmistojen rakentamiseen. Ne ovat myös hyödyllisiä monissa erilaisissa sovelluksissa. Opinnäytetyössä tehdään sarja pieniä kokeiluita luonnollisen kielen käsittelyn haasteiden ratkaisemiseksi käyttämällä NLP Cloud API:n tarjoamia malleja. API:ssa käytössä olevat luonnollisen kielen käsittelyn mallit pohjautuvat syväoppimiseen.

Opinnäytetyön tavoitteena on tutkia ja testata luonnollisen kielen käsittelyä NLP Cloud API:a hyödyntäen, sekä aloittaa API:n kanssa kommunikoivan komponenttikirjaston luominen toimeksiantajalle luonnollisen kielen käsittelyyn liittyviin tehtäviin asiakasprojektiin valittujen mikropalveluiden pohjalta. Näiden tavoitteiden tarkoituksena on, että toimeksiantaja voi tarjota opinnäytetyön aikana toteutettuja komponentteja mikropalveluina osana asiakkailleen rakentamiansa sovellusratkaisuja. Komponenttikirjaston luomista ja kehittämistä tullaan jatkamaan tämän opinnäytetyön jälkeenkin toimeksiantajalla.

Kehittämisongelmana on toteuttaa toimeksiantajalle geneerisiä ja uudelleenkäytettäviä komponentteja, jotka kommunikoivat NLP Cloud API:n kanssa. Opinnäytetyössä käytetään NLP Cloud API -rajapinnan kautta esikoulutettuja malleja, jotka hoitavat tekoälyyn ja luonnollisen kielen käsittelyyn liittyvät tehtävät. Jokaiseen API:n päätepisteeseen, jota tarvitaan mikropalvelujen toteutuksessa, tehdään omat komponenttinsa, jotka kommunikoivat API:n kanssa. Komponenttien ohjelmoinnissa käytetään TypeScript-ohjelmointikieltä. Toimeksiantajana työlle on Mindhive Oy.

Opinnäytetyön teoreettisessa osuudessa käsitellään tekoälyä, siihen liittyviä termejä ja osa-alueita, sekä sen hyötyjä ja haasteita. Teoreettisessa osuudessa tullaan esittelemään myös luonnollisen kielen käsittelyä ja siihen liittyviä tehtäviä ja käyttötapauksia, sekä erilaisia lähestymistapoja ja työkaluja luonnollisen kielen käsittelyyn. Koska tekoäly ja luonnollisen kielen käsittely ovat molemmat todella laajoja käsitteitä, tässä opinnäytetyössä ei tulla käsittelemään kaikkia niihin kuuluvia osa-alueita ja käsitteitä.

2 TEKOÄLY JA LUONNOLLISEN KIELEN KÄSITTELYN PERUSTEET

Tässä luvussa käsitellään opinnäytetyön teoreettista osuutta, jossa käydään läpi tekoälyn ja luonnollisen kielen käsittelyn perusteita. Luvussa esitellään myös luonnollisen kielen käsittelyyn liittyviä tehtäviä, käyttötapauksia, lähestymistapoja sekä työkaluja.

2.1 Tekoäly

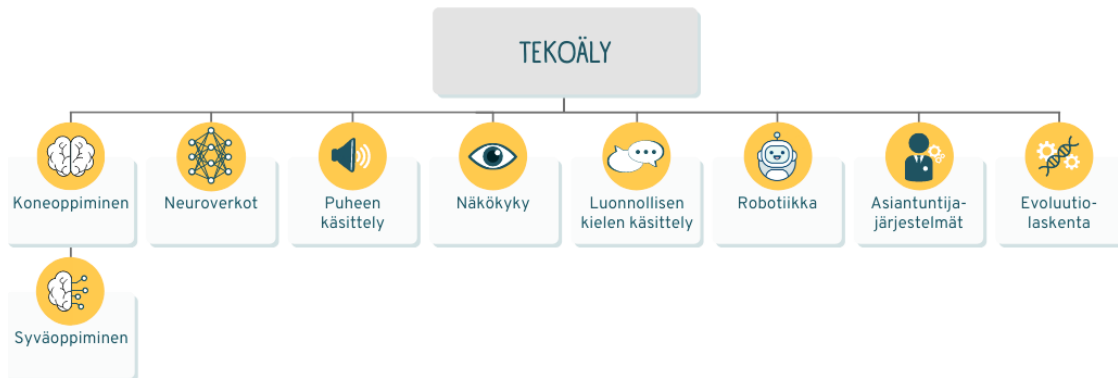
Tekoäly eli AI (engl. Artificial Intelligence) termillä viitataan yleisesti koneen kykyyn suorittaa tehtäviä ja tehdä päätöksiä, jotka mukailevat ihmisen älykkyyttä ja käytöstä (Akerkar 2019, 75). Tekoäly termi on saanut alkunsa Dartmouthin konferenssissa vuonna 1956, joten se ei ole mikään uusi keksintö, vaikkakin se on ollut kovassa nosteessa viime vuosina (Akerkar 2019, 4). Tekoälyn nykyinen suosio johtuu pääosin teknologisesta kehityksestä koneiden laskentatehossa, algoritmien kehittymisestä, datamäärien lisääntymisestä sekä tarjolla olevan tallennustilan kasvamisesta (SAS Institute 2021).

Liiketoiminnallisesta kulmasta katsottuna tekoälypohjaisten järjestelmien avulla voidaan yksinkertaistaa prosesseja, laskea yrityksen kustannuksia, kasvattaa liiketoiminnan voittoa sekä vapauttaa työntekijöiden aikaa ja energiaa työtehtäviin, joissa tarvitaan ihmisen luovuutta, työpanosta tai harkintaa. Automatisoimalla niin kutsutut ”tylsät työtehtävät” voidaan jopa kohottaa työntekijöiden työmoraalia. On hyvä kuitenkin tiedostaa, että tekoälypohjaiset järjestelmät eivät pysty hoitamaan kaikkia töitä, vaan pikemminkin yksittäisiä tehtäviä tai niiden yhdistelmiä. (Yao ym. 2018, 70–72, 157.) Useilla eri alan sektoreilla panostetaan tällä hetkellä tekoälypohjaisiin järjestelmiin, koska niiden potentiaali on valtava (Akerkar 2019, 13). Tekoälyn avulla yritykset pystyvät mm. mukautumaan nopeasti muutoksiin liiketoiminnassa, tekemään parempia liiketoimintapäätöksiä ja tehostamaan asiakkaiden kanssa kommunikointia. Sen avulla voidaan myös löytää asiakkaiden tuottamista tiedoista, eli datasta, uusia kuvioita ja trendejä erilaisten analyysien avulla. Näitä tietoja voidaan hyödyntää mm. uusissa liiketoimintamalleissa ja niiden avulla voidaan vauhdittaa uusia innovaatioita. (Akerkar 2019, 6.) Näin tekoälyn avulla voidaan myös saavuttaa etulyöntiasema markkinoilla oleviin kilpailijoihin verrattuna (Yao ym. 2018, 70–72).

Usein tekoälyjärjestelmistä voidaan hyötyä paljon ja ne voivat olla tehokkaita työkaluja, mutta on myös hyvä muistaa, että ne eivät sovi kaikkiin tilanteisiin (Yao ym. 2018, 129). Kun yritystoimintaan aloitetaan integroimaan mukaan tekoälyä, kannattaa sen käyttöönotto aloittaa pienin askelin. Näin mahdollisiin haasteisiin on helpompi reagoida, eivätkä kustannukset nouse liian suuriksi niitä kohdatessa. (Yao ym. 2018, 70–72.) Tekoälypohjaisten järjestelmien testaaminen ja kehittäminen on kuitenkin jatkuvaa työtä. Joskus niiden hienosäätö voi viedä kuukausiakin, ennen kuin ne pystyvät tuottamaan haluttuja ja hyödyllisiä tuloksia. (Yao ym. 2018, 142.) Tekoälypohjaiset järjestelmät eivät myöskään ole välttämättä kovin halpoja (Yao ym. 2018, 156). Yksi vaihtoehto kustannusten laskemiselle on käyttää kolmannen osapuolen palveluntarjoajia, jotka tarjoavat tekoälypohjaisia ratkaisuja ja malleja esimerkiksi hyvin dokumentoitujen API:en kautta, ja joiden käyttäminen ei välttämättä vaadi kehittäjältä erityistä asiantuntemusta tekoälystä (Wolff 2020).

Tekoäly on laaja käsite ja siihen kuuluu monia eri osa-alueita, tekniikoita ja termejä (Yao ym. 2018, 10). Yleensä tekoäly jaetaan heikkoon ja vahvaan tekoälyyn. Kaikki nykyiset tekoälysovellukset kuuluvat heikkoon tekoälyyn, koska ne on suunniteltu vain tiettyihin tehtäviin, eivätkä ne pysty luomaan abstrakteja käsitteitä tai siirtämään opittua tietoa muihin osa-alueisiin vaivattomasti. Vahva tekoäly taas pystyisi tähän, jolloin voitaisiin sanoa, että se omaisi vähintään ihmisen tasoisen tai korkeamman älykkyyden. Mutta mikään nykyinen tekoälysovellus ei tosiaan vielä yllä tälle tasolle. (Yao ym. 2018, 21–30.) Tekoälystä puhuttaessa viitataan siis moniin eri teknologioihin ja työkaluihin, joita on mahdollista yhdistää erilaisilla tavoilla mm. oppimaan, järjeilemään, aistimaan tai osallistumaan (Akerkar 2019, 3). Se koostuu monista eri osa-alueista (kuva 1), joiden avulla voidaan suorittaa erilaisia tekoälyyn liittyviä tehtäviä.

TEKOÄLYN OSA-ALUEET



Kuva 1. Tekoälyn osa-alueet (GSMA Future Networks 2019; RazorSocial 2020)

Kuten kuvasta 1 näkee, tekoälyn kuuluu monia eri osa-alueita (RazorSocial 2020). Kuvassa on kuitenkin esitelty vain merkittävimmät tekoälyn osa-alueet: koneoppiminen, neuroverkot, puheen käsittely, näkökyky, luonnollisen kielen käsittely, robotiikka, asiantuntijajärjestelmät sekä evoluutiolaskenta. Kuvaan on myös lisätty koneoppimisen osa-alue, syväoppiminen. Muitakin tekoälyn osa-alueita on siis olemassa. (VijiPriya ym. 2016, 543.) Luonnollisen kielen käsittelyä avataan luvussa 2.2. Asiantuntijajärjestelmillä viitataan järjestelmiin, jotka mukailevat päätöksien teossa ihmisasiantuntijoiden päättelyä, ja evoluutiolaskennalla viitataan tietokoneohjelmilla tehtävään laskentaan, joka mukaillee biologista evoluutiota (Eiben & Smith 2015, 13; JavaTpoint s.a.). Seuraavissa osioissa käsitellään muutama tekoälyn ja tähän opinnäytetyöhön liittyvä termi, koska niihin tullaan viittaamaan useasti tässä opinnäytetyössä. Kyseessä olevat termit ovat data ja malli. Näiden jälkeen käsitellään vielä muutama tekoälyn osa-alue, koneoppiminen ja neuroverkot sekä koneoppimisen osa-alue, syväoppiminen.

Data

Data on minkä tahansa tekoälypohjaisen järjestelmän peruspilari (Yao ym. 2018, 130). Data voi olla oikeastaan mitä muotoa tahansa ja mistä tahansa. Se voi olla mm. numeroita, sanoja, lauseita, äänitteitä tai vaikka oletuksia. Se ei kuitenkaan välttämättä ole aina totta, ja sitä voi olla vaikea huomata, kun

dataa käsitellään. (Akerkar 2019, 1.) Ilman tilanteeseen sopivaa ja totuudenmukaista dataa mitkään tekoälypohjaiset mallit eivät pysty tuottamaan oikeanlaisia tuloksia (Yao ym. 2018, 130). Osa tekoälypohjaisista ratkaisuista vaatii oppimiseen ja päätöksen tekoon valtavasti dataa (Akerkar 2019, 13).

Malli

Malli, tai toiselta nimeltään hypoteettinen tila, on koulutettu tunnistamaan datasta tietyn tyyppisiä kuvioita (Jung 2022, luku 2; What is a machine... 2021). Malli koulutetaan datalla, ja se sisältää myös aina jonkin algoritmin, mikä suorittaa laskentaa, ja minkä avulla malli pystyy päättämään ja oppimaan (What is a machine... 2021). Mallia kouluttaessa ja sen hienosäätöjä tehdessä on hyvä muistaa, että tarvittava tarkkuus mallin tuottamassa lopputuloksessa riippuu sen käyttötarkoituksesta. Esimerkiksi lääketieteessä tai itseohjautuvissa autoissa mallien tuottamien lopputuloksien tulee olla tarkkuudeltaan todella korkealla tasolla, koska näissä tapauksissa virheillä voisi olla hyvin vakavia seurauksia. On myös hyvä tiedostaa, etteivät edes nämä edellä mainittuihin tilanteisiin sovellettavat mallit pysty tuottamaan käytännössä 100 % oikein olevia lopputuloksia. (Yao ym. 2018, 134.) Usein vähemmän tarkkuutta vaativissa tehtävissä ns. ”tarpeeksi hyvä malli” on käytännöllisempi kuin täydellisyyden tavoittelu. Nämä asiat on hyvä huomioida tekoälysovelluksia kehitettäessä. (Yao ym. 2018, 81.)

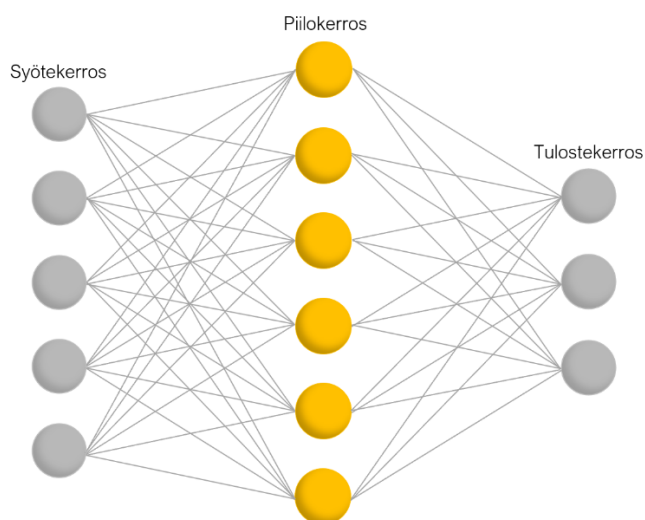
Koneoppiminen

Koneoppiminen on yksi tekoälyn ja tietojenkäsittelytieteen osa-alue (Yao ym. 2018, 13). Kone- ja syväoppimiseen kuuluu kolme peruskomponenttia: data, malli ja häviöfunktio (engl. loss function), jonka avulla mitataan, että kuinka hyvin malli suoriutuu (Jung 2022, luku 2). Koneoppimisen avulla voidaan mm. päätellä mitkä ominaisuudet ja kaavat ovat syötetyssä datassa tärkeitä. Kone oppii, että mitkä algoritmit selittävät parhaiten koulutusdataa ja pystyvät myöhemmin toimimaan hyvin uuden datan kanssa. Tämä kyky kone- ja syväoppimisessä mahdollistaa mm. paremman kuvien, videoiden, äänen, puheen ja tekstin analysoinnin kuin missään ihmisen suunnittelemassa ja toteuttamassa ohjelmointikoodissa. (Yao ym. 2018, 175.)

Koneoppiminen voidaan myös jakaa neljään eri tyyppiin: ohjattuun oppimiseen, ohjaamattomaan oppimiseen, puoliohjattuun oppimiseen sekä vahvistusoppimiseen. Ohjatussa oppimisessa mallia koulutettaessa data jaetaan eri osiin, joita ovat koulutusdata, testidata ja validointidata, jotta mallin suorituskykyä voidaan arvioida (Jung 2022, luku 6.3). Mallille syötetään siis harjoitteluvaiheessa dataa, jossa jokaisen datapisteen ominaisuudet ja niiden oikea lopputulos, eli arvo, tiedetään jo valmiiksi. Näin mallin tarkkuutta on helpompi arvioida lopuksi. Esimerkkinä tästä voi olla se, että mallille syötetään kuva koirasta, ja tälle kuvalle on jo tiedossa sen oikea arvo, tai luokka, eli ”koira”. Ohjaamattomassa oppimisessä näitä arvoja ei tiedetä valmiiksi, vaan mallin pitää löytää itse erilaisia kaavoja syötetystä datasta, joita ihmisten on yleensä vaikea hahmottaa. Puoliohjattu oppiminen on taas näiden kahden välimaastossa. Esimerkiksi Netflix-suoratoistopalvelun suositukset perustuvat tämän tyyppiin oppimiseen. Vahvistusoppimisella tarkoitetaan taas sitä, että mallille määritetään joku tavoite, ja malli koittaa yrityksen ja erehdyksen kautta saavuttaa tämän tavoitteen, oppimalla edellisistä yrityksistä. (Yao ym. 2018, 13–14.)

Neuroverkot

Neuroverkoilla viitataan malleihin, jotka on kehitetty mukailemaan ihmisaivojen toimintaa. Neuroverkot koostuvat yksiköistä, joita kutsutaan neuroneiksi, ja nämä neuronit on liitetty toisiinsa kerroksittain (kuva 2). (Géron 2019, 279–283.)



Kuva 2. Neuroverkko (Akerkar 2019, 35)

Kuvasta 2 voidaan nähdä, että neuroverkoilla on aina vähintään kolme kerrosta: syötekerros, piilokerros ja tulostekerros. Mitä enemmän piilokerroksia on, sitä syvempi neuroverkko on. Neuroneita voi olla eri määriä yhdellä kerroksella ja aikaisemman kerroksen neuronit ovat yhteydessä seuraavan kerroksen neuroneihin. Jokainen piilotetun kerroksen neuroni tekee laskutoimituksia, ja välittää omien laskutoimituksiensa tulokset aina seuraavalle kerrokselle. (Akerkar 2019, 30–35.)

Syväoppiminen

Syväoppiminen on koneoppimisen osa-alue. Syväoppimisen mallit käyttävät syviä neuroverkkoja ongelmien ratkaisemiseen, jolloin piilokerroksia on vähintään kolme. Sen avulla voidaan yltää huikkeisiin tuloksiin joillain osa-alueilla, mutta useat koneoppimiseen liittyvät tehtävät voidaan ratkaista kuitenkin kevyemmällä malleilla. Syväoppimisen mallit vaativat paljon dataa koulutusvaiheessa, ja ne myös vaativat tietokoneilta paljon laskentatehoa. (Yao ym. 2018, 15–16.)

Yao ym. (2018, 9) mukaan, heiltä kysytään usein tekoälyn, kone- ja syväoppimisen sekä datatieteen eroa, ja he määrittelevätkin, että nämä eri tekniikat eroavat toisistaan niiden potentiaalisella vaikutuksella sekä käyttömahdollisuuksilla, mutta ne ovat kaikki esimerkkejä koneälystä. Erilaisten tekniikoiden erottelu siihen, että mikä lasketaan tekoälyksi ja mikä ei, voi olla hankalaa. Monet menestyneimmistä nykyaikaisista tekoälysovelluksista pohjautuvat kuitenkin jollain tapaa koneoppimiseen. (Yao ym. 2018, 10.) Niinpä Yao ym. (2018, 21–30) ovat jakaneet erilaiset järjestelmät ja sovellukset seitsemään eri tasoon niiden toiminnallisuuden ja monimutkaisuuden perusteella (taulukko 1).

Taulukko 1. Järjestelmien ja sovellusten seitsemän eri tasoa (Yao ym. 2018, 21–30)

Toimivat järjestelmät	Alin taso. Järjestelmiä, jotka toimivat ennakkoon määritetyn kaavan ja sääntöjen mukaan, kuten palohälytintä tai auton vakionopeudensäädin. Ei dynaamista päätöksen tekoa. Tämän tason järjestelmiä ei pitäisi väittää tekoälypohjaisiksi järjestelmiksi, mutta valitettavasti jotkin yhtiöt tekevät niin.
Ennustavat järjestelmät	Järjestelmiä, jotka ovat kykeneviä analysoimaan dataa ja tuottamaan todennäköisyysennusteita. Suuri osa näistä järjestelmistä käyttää tilastotiedettä. Mm. voidaan arvioida asiakkaan ostamien tuotteiden perusteella, onko asiakas rasakaana. Ennusteita voidaan käyttää esimerkiksi mainosten kohdentamiseen. Yleensä paljon manuaalista työtä.
Oppivat järjestelmät	Järjestelmiä, jotka tekevät myös ennusteita, mutta voivat oppia suorittamaan tehtäviä ilman nimenomaista ohjelmointia niihin. Monien näiden järjestelmien takana on kone- ja syväoppimista. Yksinkertaisimmissa järjestelmissä ihminen voi kerätä tarvittavan datan ja tehdä toimintapäätöksiä ennusteiden perusteella ja monimutkaisimmissa järjestelmissä, kuten esimerkiksi itseohjautuvat autot tai teolliset robotit, järjestelmät pystyvät hoitamaan tehtäviä aina datan keräämisestä lopulliseen toimintaan tekemiensä analyysien perusteella.
Luovat järjestelmät	Järjestelmiä, jotka pystyvät luomaan esimerkiksi tekstiä, kuvia, musiikkia, teollisia malleja tai ohjelmointikoodia.
Samaistuvat järjestelmät	Järjestelmiä, jotka pyrkivät tunnistamaan tunnetiloja tekstin, äänen, ilmeiden tai kehonkielen perusteella. Tunneanalyysit voivat auttaa esimerkiksi digitaalisia avustajia vastaamaan tilanteeseen sopivalla tavalla ja liiketoiminnassa tunneanalyysien avulla voidaan parantaa mm. myyntiä ja markkinointia.
Hallitsevat järjestelmät	Järjestelmiä, jotka ovat kykeneviä rakentamaan abstrakteja konsepteja ja tekemään päätöksiä niukalla datalla. Lapsi mm. pystyy luomaan abstraktin käsityksen objektista kerran sen nähtyään (esim. koira), ja pystyy sen jälkeen käyttämään tätä tietoa muillakin alueilla, kun taas syväoppimisen mallit joutuvat käsittelemään tuhansia kuvia pystyäkseen tunnistamaan objekteja kuvista ja videoista. Tämän jälkeenkään mallit eivät luotettavasti tunnista objektin abstrakteja muotoja, kuten piirrettyä koira tai koirapukua. Mikään nykyinen tekoäly ei yllä tähän luokkaan.
Kehittyvät järjestelmät	Järjestelmiä, jotka pystyvät dynaamisesti muuttamaan mm. omaa arkkitehtuuriaan. Ihmiset ja muut biologiset organismit kehittyvät geneettisten mutaatioiden avulla sukupolvien aikana, mutta mitkään koneet eivät vielä pysty tähän.

Kuten taulukosta 1 näkee eri tasoisia järjestelmiä ja sovelluksia on paljon, mutta jaottelu näihin eri tasoihin voi auttaa ymmärtämään toiminnallisia eroja erilaisissa tekoälysovelluksissa. On myös hyvä ymmärtää, että vaikka tämä jaottelu on tehty seitsemään eri osaan, niin näissä tasoissa esiintyy monia päällekkäisyyksiä, eikä niiden välille voi tehdä selvää rajausta. Monissa tekoälysovelluksissa lupaavimpia tuloksia antavat oppivat-, luovat- ja samaistuvat järjestelmät, mutta viimeisimpään kahteen tasoon, eli hallitseviin- ja kehittyviin järjestelmiin, erilaiset tekoälysovellukset eivät vielä yllä. (Yao ym. 2018, 21–30.)

Tekoälyn saralla tehdään tutkimusta jatkuvasti, ja uusia löytöjä joka päivä. Myös uusia työkaluja ja menetelmiä kehitetään koko ajan, ja se onkin siis nopeasti kehittyvä teknologia-alue. (Yao ym. 2018, 203.) Yao ym. (2018, 176) mukaan Googlen ja Teslan työntekijät, Pete Warden ja Andrej Karpathy, ovat ennustaneet, että kymmenen vuoden sisään iso osa ohjelmistoalan töistä ei pidä sisällään ohjelmointia, vaan datan keräämistä, sen esikäsittelyä, sekä sen visualisointia ja analysointia, jotta se voidaan syöttää neuroverkoille. Jo nyt kone- ja syväoppimisen sekä luonnollisen kielen käsittelyn avulla voidaan esimerkiksi luoda eri ohjelmointikielille vakuuttavaakin ohjelmointikoodia tekstin avulla, jossa voidaan kuvailla luonnollisella kielellä, että millainen ohjelmointikoodi halutaan muodostaa (Zaremba & Brockman 2021).

2.2 Luonnollisen kielen käsittely

Kuten tekoälykin, luonnollisen kielen käsittely on todella laaja käsite ja aihealue, joten kaikkea siihen liittyvää ei ole mahdollista käsitellä tässä opinnäytetyössä. Sen ydinkonsepti ja joitain luonnollisen kielen käsittelyyn liittyviä keskeisiä käsitteitä on kuitenkin pyritty esittelemään tässä opinnäytetyössä selkeästi ja informatiivisesti.

Luonnollisen kielen käsittely eli NLP (engl. Natural Language Processing) on tietojenkäsittely- ja kielitieteen sekä tekoälyn osa-alue, jossa käsitellään menetelmiä puhutun tai kirjoitetun ihmiskielen analysoimiseen, mallintamiseen ja ymmärtämiseen (IBM Cloud Education 2021; Krupsky 2021). Tässä opinnäytetyössä keskitytään enemmän kirjoitettuun ihmiskieleen, mutta myös joitain

puhutun ihmiskielen käsittelyyn liittyviä sovelluksia esitellään myöhemmin tässä luvussa. Luonnollisen kielen käsittely jaetaan myös joskus sen ala-alueisiin, luonnollisen kielen ymmärtämiseen eli NLU (engl. Natural Language Understanding) ja luonnollisen kielen luomiseen eli NLG (engl. Natural Language Generation) (IBM Technology 2021). Viimeisimpien vuosien aikana varsinkin luonnollisen kielen luominen on ollut merkittävän kehitystyön alla (Yao ym. 2018, 183). Luonnollisen kielen ymmärtämisen tavoitteena on analysoida ihmisten tuottamaa tekstiä tai puhetta mahdollisimman tarkasti, kun taas luonnollisen kielen luomisen tavoitteena on tuottaa samankaltaista tekstiä tai puhetta kuin ihmisetkin tuottavat (IBM Cloud Education 2021).

Luonnollisen kielen käsittelyssä analysoidaan erilaisia kielitieteeseen liittyviä asioita ja näkökulmia, kuten syntaksia eli lauseoppia, semantiikkaa eli merkitysoppia, pragmatiikkaa eli tilanteen ja kontekstin merkitystä sekä morfologiaa eli sanojen muotojen tutkimista. Luonnollisen kielen käsittely yhdistelee kielitieteitä ja tietojenkäsittelyä kielen sääntöjen ja rakenteiden tutkimiseksi. (MonkeyLearn s.a.) Se siis perustuu moniin kieliopillisiin sääntöihin, mutta myös kontekstin ymmärtäminen on tärkeää. Esimerkiksi suomen kielessä sanalla tai lauseella voi olla monta eri merkitystä, kuten lauseella ”Kuusi palaa.”. Se voi tarkoittaa esimerkiksi sitä, että joulukuusi on tulossa tai kuusi palaa jotain, kuten sokeria. (Krupsky 2021.)

Luonnollisen kielen käsittelyn avulla voidaan mm. kääntää tekstiä kielestä toiselle, vastata puhuttuihin komentoihin tai tehdä yhteenvetoja tekstistä (IBM Cloud Education 2021). Todennäköisesti sinäkin olet käyttänyt joitain siihen pohjautuvia sovelluksia, kuten esimerkiksi Google kääntäjää, virtuaalisia avustajia kuten Alexa, Siri tai Google Assistant, nettisivuilla asiakaspalvelun chatti-bottia tai sinulla voi olla sähköpostissasi käytössä sähköpostiviestien automaattinen luokittelu mm. roskapostiksi (IBM Cloud Education 2021; Yao ym. 2018, 27, 132–134.) Näitä erilaisia tehtäviä ja niihin liittyviä käyttötapauksia käsitellään tarkemmin luvussa 2.3.

Ensimmäiset luonnollisen kielen käsittelyn sovellukset olivat sääntöpohjaisia käsin ohjelmoituja järjestelmiä, ja niistä on edelleen hyötyä joissain luonnollisen kielen käsittelyyn liittyvissä tehtävissä (IBM Cloud Education 2021). Kone- ja syväoppimisen sekä luonnollisen kielen käsittelyyn sopivien mallien avulla

koneet analysoivat ja pyrkivät ymmärtämään sekä erottamaan tekstin ja puheen merkityksen. (MonkeyLearn s.a.) Niiden avulla pystytään myös automaattisesti luonnollisen kielen käsittelyyn liittyviä tehtäviä (IBM Cloud Education 2021).

Luonnollisen kielen käsittelyyn perustuvilla järjestelmillä on myös kasvava rooli liiketoiminnassa, koska ne voivat auttaa virtaviivaistamaan liiketoimintaa, lisäämään sen tuottavuutta ja yksinkertaistamaan prosesseja (IBM Cloud Education 2021). Asiakkaiden palautteiden ja käyttäytymisen analysoiminen tarjoaa yrityksille mm. arvokasta tietoa siitä, että mitä asiakkaat odottavat yritykseltä, ja mitä he ovat mieltä yrityksen palveluista tai tuotteista (Wolff 2020). Näiden asioiden ymmärtäminen voi johtaa yrityksessä esimerkiksi uusiin innovaatioihin ja parempaan mainonnan kohdentamiseen (Krupsky 2021). Koska näiden tietojen käsittely on usein toistuvaa ja aikaa vievää puuhaa manuaalisesti tehtynä, luonnollisen kielen käsittely ja prosessien automatisoiminen on usein järkevä ratkaisu. Kuitenkin vielä iso osa yrityksistä ponnistelee löytääkseen parhaimman tavan näiden tietojen analysoimiseen. (Wolff 2020.) Luonnollisen kielen käsittelyn haasteisiin kuuluu yritystoiminnan näkökulmasta mm. siihen perustuvien sovelluksien kehittämiseen ja datan keräämiseen käytetty aika, ja täten myös niiden käyttöönoton lopullinen hinta (Chowdhury 2021; Nityasya ym. 2021). Sen avulla voidaan kuitenkin parantaa yrityksen kilpailukykyä mm. suorittamalla laajamittaisia analyysejä, esimerkiksi yrityksen sosiaalisen median kanavilla olevista keskusteluista ja kommentteista, verkkoarvosteluista tai vaikka asiakastuen tiketeistä. Luonnollisen kielen käsittelyyn ja muihin tekoälyn osa-alueisiin perustuvien työkalujen avulla voidaan lajitella ja reitittää tietoa kellon ympäri, tarkasti, nopeasti ja tehokkaasti, automatisoimalla prosesseja. (MonkeyLearn s.a.) Erilaisia työkaluja, lähestymistapoja ja ratkaisuja luonnollisen kielen käsittelyyn liittyviin haasteisiin käsitellään myöhemmin luvussa 2.4.

Luonnollisen kielen käsittely on koneille vaativa tehtävä, koska niin puhutussa kuin kirjoitetussakin ihmiskielessä on paljon erilaisia variaatioita ja kiemuroita, kuten eri kielet ja niiden lauserakenteiden vaihtelut, kielioppoikkeukset sekä puhutun kielen erilaiset murteet ja painotukset. Ihmiset saattavat käyttää myös esimerkiksi metaforia tai jopa sarkasmia. (IBM Cloud Education 2021.) Kun ih-

miset tuottavat joko puhetta tai kirjoitettua tekstiä, sitä kutsutaan jäsentämättömäksi dataksi (IBM Technology 2021). On arvioitu, että jopa 90 % digitaalisesta datasta on jäsentämätöntä dataa, joka ei ole suoraan koneiden ymmärrettävissä (Krupsky 2021). Jotta koneet pystyisivät käsittelemään puhetta tai tekstiä, jäsentämätön data pitää muuttaa ensin jäsenneyksi dataksi. Tässä luonnollisen kielen käsittely astuu kuvioihin. Luonnollisen kielen ymmärtämistä käytetään, jotta jäsentämätön data saadaan koneille ymmärrettävään muotoon, ja luonnollisen kielen luomista käytetään silloin, kun koneen tuottama jäsenney data halutaan saada ihmisten tuottaman tekstin tai puheen kaltaiseksi. (IBM Technology 2021.) Seuraavissa osioissa käsitellään yleisimpiä vaiheita ja käsitteitä, jotka liittyvät tekstimuotoisen jäsentämättömän datan muuttamiseksi koneille ymmärrettävään muotoon, jotta tämän prosessin perusteiden ymmärtäminen olisi helpompaa. Nämä osiot ovat tekstin esikäsitteily ja tietojen esittäminen.

Tekstin esikäsitteily

Tekstiä on esikäsiteltävä, jotta se saadaan koneelle ymmärrettävään muotoon (Menzli 2021). Tekstin esikäsitteilyn ensimmäisiin vaiheisiin sisältyy yleensä niin kutsuttu tokenisaatio eli saneistus (IBM Technology 2021). Tokenisaatiolla tarkoitetaan tehtävää, jossa teksti pilkotaan pienempiin osiin eri elementeiksi, joita kutsutaan tokeneiksi (Lantz 2019, 112). Ne voivat olla mm. sanoja, lauseita, määrätyn mittaisia merkkijonoja, symboleita, tai vaikka numeroita. Tokenisaatio on tärkeä vaihe jokaisessa luonnollisen kielen käsittelyyn liittyvässä sovelluksessa (Menzli 2021). Tokenisaation jälkeen kone käsittelee tekstiä palloissa, yksi tokeni kerrallaan (IBM Technology 2021).

Usein tekstistä poistetaan myös hukkas sanat. Hukkasanoilla tarkoitetaan joukkoa erittäin yleisiä täytesanoja, kuten 'ja', 'tai', 'sekä', 'joten', 'mutta', 'koska'. Tällaiset sanat poistetaan yleensä siksi, koska ne eivät tarjoa hyödyllistä tietoa paljoakaan, vaikka näitä sanoja esiintyykin tekstissä hyvin tiheästi. Eri kielille on olemassa erilaisia täytesanoja. Hukkasanoja voi määrittää myös itse, jos datasta haluaa karsia muuta tietoa, joka ei ole itselle hyödyllistä. (Lantz 2019, 109–110.)

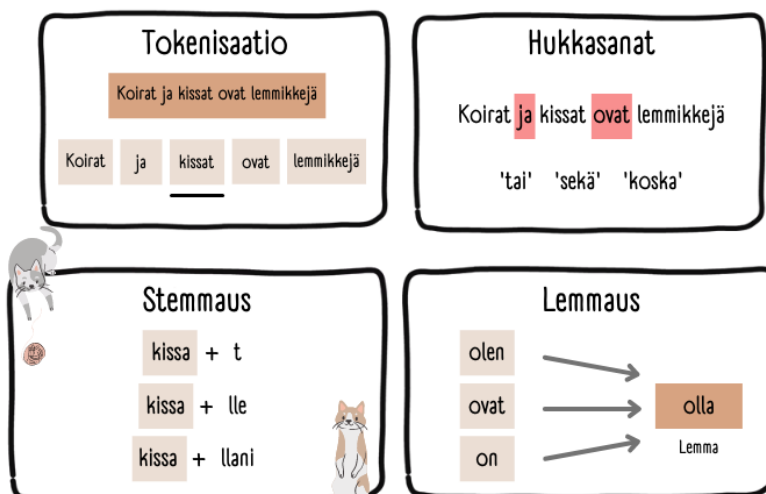
Tekstin esikäsittelyyn kuuluu myös yleensä stemmaus tai lemmaus. Stemmaus on prosessi, jossa taivutetut sanat muutetaan perusmuotoon (Lantz 2019, 111). Stemmaus on nopeampi prosessi kuin lemmaus, mutta ongelmallinen siinä mielessä, että muodostettu sana eli stemma ei ole välttämättä minikään kielen oikea sana. (Beri 2020.) Alla muutama esimerkki stemmauksesta:

1. kissat -> kissa
2. autolla -> auto
3. vakuutuksellani -> vakuutus (ei oikea sana)
4. kuusi -> kuu (merkitys voi muuttua)

Lemmaus taas on prosessi, jossa stemmauksesta poiketen taivutetut sanat muutetaan perusmuotoon kunnollisesti tarkistamalla sana kyseisen kielen sanastosta, varmistaen näin, että kantasana kuuluu kieleen. Tässä tapauksessa muodostettua sanaa kutsutaan lemmaksi. Lemmaus on hitaampi prosessi kuin stemmaus, mutta se on tarkempi näistä kahdesta. (Beri 2020.) Alla muutama esimerkki lemmauksesta:

1. koirat, koiriin, koirien -> koira
2. maat, maihin, maiden -> maa
3. kahdeksannet, kahdeksansiin, kahdeksansien -> kahdeksas

Tokenisaatio, hukkas sanat, stemmaus sekä lemmaus (kuva 3) ovat vain joitain yleisiä esimerkkejä tekstin esikäsittelyn eri vaiheista. Tekstin esikäsittelyyn kuuluu usein myös muita vaiheita, mutta erilaiset vaiheet riippuvat siitä, että mitä luonnollisen kielen käsittelyn avulla halutaan saavuttaa. (Agrawal 2021.)



Kuva 3. Tokenisaatio, hukkas sanat, stemmaus ja lemmaus (Simplilearn 2021)

Kuvasta 3 voidaan nähdä, että tokenisaatiossa teksti pilkotaan pienemmiksi osiksi eli tokeneiksi. Käsiteltävästä tekstistä poistetaan myös hukkas sanat. Stemmauksessa ja lemmauksessa sanoista poistetaan taivutukset ja pyritään löytämään sanan perusmuoto. (Simplilearn 2021.)

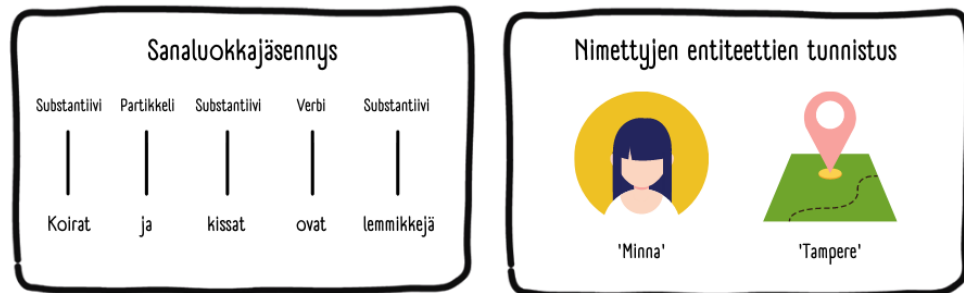
Tietojen esittäminen

Tietojen esittäminen on myös tärkeä vaihe jokaisessa luonnollisen kielen käsittelyyn liittyvässä tehtävässä, jotta teksti saadaan koneille ymmärrettävään muotoon (Singhal 2021). Sääntöpohjaisissa menetelmissä teksti voidaan usein esittää merkkijonoina, mutta kone- ja syväoppimisessa tekstidata tulee esittää numeerisessa muodossa (Hyvärinen 2018). Luonnollisen kielen käsittelyssä tätä prosessia, jossa tekstidata muutetaan numeeriseen muotoon, kutsutaan vektorisaatioksi (Singhal 2021). Vektorisaatioon on olemassa luonnollisen kielen käsittelyssä erilaisia lähestymistapoja, kuten harvat ja tiheät vektorisitykset, sekä niihin liittyviä tekniikoita ja malleja, kuten one-hot-encoding, BoW eli Bag of Words, TF-IDF eli Term Frequency-Inverse Document Frequency, word embedding, Word2Vec, GloVe ja FastText (Hyvärinen 2018; Jha 2021; Singhal 2021). Koska tietojen esittäminen on todella laaja aihealue, sen enempää sitä ei tulla avaamaan tässä opinnäytetyössä. Se on kuitenkin osa luonnollisen kielen käsittelyä ja sen olemassaolo on hyvä tiedostaa.

2.3 Luonnollisen kielen käsittelyyn liittyviä tehtäviä ja käyttötapauksia

Luonnollisen kielen käsittelyyn liittyy todella monia erilaisia tehtäviä, eikä niitä kaikkia ole siten mahdollista käydä läpi tässä opinnäytetyössä. Joitain esimerkkejä erilaisista tehtävistä ovat mm. tokenisaatio, kielioppivirheiden korjaus, tekstin ja asiakirjojen luokittelu, aiheiden tunnistus, avainsanojen erottaminen, tekstin tarkoituksen tunnistaminen, yhteenvetojen tekeminen, chattiboteissa usein käytetty Q&A (engl. Question Answering) eli kysymys ja vastaus, tekstin luominen, tunneanalyysi, semanttisen samankaltaisuuden määrittäminen eli kuinka samankaltaisia tekstit ovat keskenään merkitykseltään, sanaluokkajäsennys eli onko sana esimerkiksi verbi tai prepositio sekä nimettyjen entiteettien eli olioiden tunnistus, kuten esimerkiksi henkilöiden, organisaatioiden ja paikkojen nimet. (IBM Cloud Education 2021; Jayathilaka 2020.)

Esimerkiksi sanaluokkajäsennys (engl. Part-Of-Speech Tagging) ja nimettyjen entiteettien tunnistus eli NER (engl. Named Entity Recognition) ovat hyvin yleisiä tehtäviä luonnollisen kielen käsittelyssä (kuva 4) (MonkeyLearn s.a.).



Kuva 4. Sanaluokkajäsennys ja nimettyjen entiteettien tunnistus (Simplilearn 2021)

Kuten kuvasta 4 näkee, sanaluokkajäsennyksessä tokeneille määritetään sanaluokat (Simplilearn 2021). Tässä prosessissa tutkitaan myös, että missä yhteydessä tokenia käytetään lauseessa, koska jotkut sanat voivat kuulua useaan eri sanaluokkaan, riippuen sen käyttöyhteydestä. Esimerkiksi aikaisemmin tämän opinnäytetyön luvussa 2.2. esitetty lause 'Kuusi palaa.' on tästä hyvä esimerkki. Sana 'kuusi' voi olla joko substantiivi kun puhutaan puusta, tai sitten se voi olla numeraali eli lukusana kun puhutaan numerosta kuusi. (IBM Technology 2021; Voutilainen ym. 2012, 11.) Nimettyjen entiteettien tunnistuksessa tekstistä taas paikallistetaan ja luokitellaan mm. henkilöiden ja paikkojen nimet. Esimerkiksi 'Antti' voitaisiin luokitella luokkaan 'henkilö' ja 'Helsinki' luokkaan 'kaupunki'. (IBM Technology 2021.)

Kone- ja syväoppimisen avulla voidaan ketjuttaa yhteen useita luonnollisen kielen prosessointitehtäviä, kuten tunteiden analysointia, avainsanojen poimimista, aiheiden luokittelua tai tekstin merkityksen havaitsemista. Ketjuttamisen avulla voidaan saavuttaa erittäin hyviä tuloksia. Tiettyjä yleisimpiä prosessointitehtäviä on melkein pakkokin ketjuttaa, jotta edistyneempiä luonnollisen kielen käsittelyä käyttäviä sovelluksia voidaan tehdä. (MonkeyLearn s.a.)

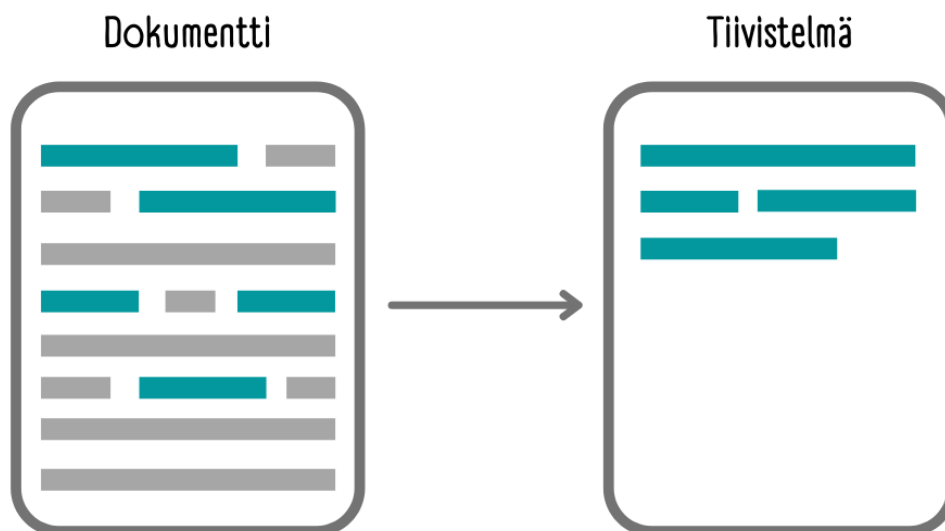
Luonnollisen kielen käsittelyyn liittyviä tehtäviä voidaan hyödyntää eri toimialoilla ja monenlaisiin sovelluksiin. Esimerkiksi Gmail käyttää luonnollisen kielen käsittelyä sähköpostien luokittelussa erilaisiin luokkiin, kuten tarjous-,

sosiaalinen-, ensisijainen- tai roskapostiviesti, käyttämällä hyväksi avainsanojen poimintaa. Käsittelemällä sanoja aiherivillä ja yhdistämällä ne ennakkoon määritettyihin tunnisteisiin, koneet pystyvät luokittelemaan sähköposteja automaattisesti. (MonkeyLearn s.a.) Niitä voidaan hyödyntää myös tiedonhaussa ja osuvimpien tuloksien palauttamiseen, kuten Google-haku, tai ohjelmitoissa, jotka pystyvät korjaamaan kielioppivirheitä, kuten Word (Jayathilaka 2020; MonkeyLearn s.a.). Myös Google kääntäjä käyttää hyödykseen luonnollisen kielen käsittelyä (IBM Cloud Education 2021). Sitä voidaan käyttää hyödyksi myös esimerkiksi sosiaalisen median analyysissä ja seurantatyökaluissa, kielen tunnistuksessa erilaisissa sovelluksissa, dokumenttien automaattisessa prosessoinnissa, kuvatekstien- ja otsikkojen luomisessa, chattiboteissa ja virtuaalisissa avustajissa, kuten Siri ja Alexa, tai vaikka jopa valeuutisten luomisessa ja havaitsemisessa. Luonnollisen kielen käsittelyä voidaan tulevaisuudessa käyttää myös monissa uusissa innovaatioissa, joita ei ole vielä keksittykään, ja joissa vain mielikuvitus on rajana. (AltexSoft 2021; IBM Cloud Education 2021; Jayathilaka 2020; MonkeyLearn s.a.)

Seuraavissa osioissa esitellään hieman tarkemmin kolme luonnollisen kielen käsittelyyn liittyvää tehtävää ja niihin liittyviä käyttötapauksia, koska niitä tullaan käsittelemään myös myöhemmin tämän opinnäytetyön käytännön osuudessa. Osioissa käsiteltävät esimerkkitehtävät ja niihin liittyvät käyttötapaukset ovat tekstin yhteenveto, tunneanalyysi ja kysymyksiin vastaaminen.

Esimerkki 1: Tekstin yhteenveto ja siihen liittyviä käyttötapauksia

Tekstin yhteenvedolla tarkoitetaan prosessia, jossa teksti tiivistetään lyhyemmäksi kuin alkuperäinen teksti (kuva 5). Sen tarkoituksena on siis koostaa tekstin tärkeimmät tiedot tiivistettyyn pakettiin. Yhteenvetojen toteuttamisessa käytetään luonnollisen kielen käsittelyyn liittyviä tekniikoita, joskus valtavienkin tekstimäärien käsittelemiseen, ja tiivistelmien luomiseen, esimerkiksi erilaisille hakemistoille, tutkimustietokannoille tai kiireisille lukijoille, joilla ei ole aikaa tai halua lukea koko tekstiä. (IBM Cloud Education 2021.)



Kuva 5. Tekstin yhteenveto (Thaker 2019)

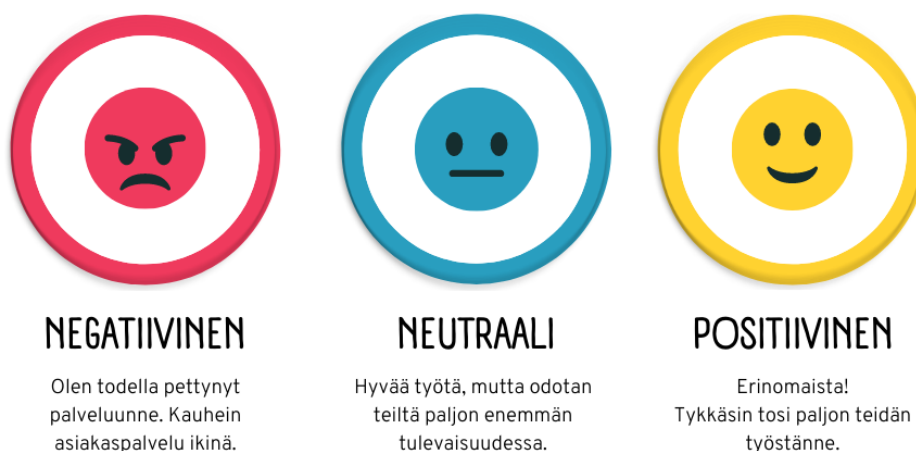
Kuten kuvasta 5 näkee, tekstin yhteenvedossa voidaan poimia tärkeää tietoa alkuperäisestä tekstistä ja tiivistää tärkeimmät asiat lyhyemmäksi ja helpommin sisäistettäväksi tekstiksi (Thaker 2019). Luonnollisen kielen käsittelyssä on olemassa kaksi erityyppistä tapaa tehdä näitä yhteenvetoja, poimimispohjainen ja käsitteellinen tapa. Poimimispohjaisessa yhteenvedossa tekstistä poimitaan tärkeimmät tiedot ja käytetään niitä tiivistelmän tekemiseen. Käsitteellisessä tavassa yhteenvedossa hyödynnetään syväoppimista, joten koneet pystyvät luomaan itse uusia lauseita aiheeseen liittyen ja sellaisia tekstiosioita, joita ei ole alkuperäisessä tekstissä. (MonkeyLearn s.a.) Käsitteelliseen tapaan perustuvat sovellukset käyttävät siis yhteenvetojen tekemiseen luonnollisen kielen luomista ja semanttista päättelyä, eli tekstin merkityksen löytämistä (IBM Cloud Education 2021).

Tekstin yhteenvetoa voidaan käyttää monissa eri tilanteissa. Jotkin työt vaativat työntekijöiltä valtavan määrän aikaa erilaisten tekstien lukemiseen. Yhteenvetosisältöjen koostaminen voi auttaa heitä säästämään paljon aikaa ja energiaa. (Summarization API s.a.) Yhteenvetoa voidaan käyttää myös esimerkiksi oikeudellisten asiakirjojen jäsentämisen apuna, akateemisten julkaisujen tai keskustelujen tiivistämiseen, otsikoiden luomiseen ja vaikka apuna raporttien kirjoittamiseen (MonkeyLearn s.a.; Summarization API s.a.). Ehkäpä tällaista automaattisesti luotua yhteenvetoa voisi käyttää jopa opinnäytetyössä tekstiosion pohjana.

Esimerkki 2: Tunneanalyysi ja siihen liittyviä käyttötapauksia

Jotta luonnollisen kielen käsittelyyn liittyvät tekoälysovellukset menestyisivät kommunikoinnissa ihmisten kanssa, ne tarvitsevat siihen tunneälyä. Tunneanalyysissa voidaan erotella ja määritellä erilaisia tunnetiloja tekstistä, puheesta, ilmeistä sekä kehonkielestä, ja sitä voidaan hyödyntää esimerkiksi digitaalisissa avustajissa, jotta ne pystyvät vastaamaan tilanteeseen sopivalla tavalla. (Yao ym. 2018, 27.) Luonnollisen kielen käsittelyyn soveltuvia malleja opetetaan luokittelemaan tekstiä mielipiteiden tai tunnetilojen mukaan (kuva 6), kuten positiivinen, negatiivinen, neutraali, vihainen, iloinen tai yllättynyt (MonkeyLearn s.a.; Sentiment and Emotion... s.a.).

TUNNEANALYYSI



Kuva 6. Tunneanalyysi (Murali 2021)

Tunneanalyysin avulla voidaan luokitella esimerkiksi asiakkaiden kommentteja asiakaspalveluun liittyen (kuva 6). Kuvassa on käytetty esimerkkinä negatiivista, neutraalia ja positiivista luokittelua, mutta sen avulla voidaan tunnistaa myös muita mielipiteitä ja tunteita. (Murali 2021.)

Tunneanalyysi on yksi suosituimmista luonnollisen kielen käsittelyn tehtävistä (MonkeyLearn s.a.). Sen perusteella voidaan poimia ihmisten asenteita ja tunteita esimerkiksi yritysten tuotearvosteluista, kampanjoihin ja tapahtumiin liittyvistä keskusteluista ja sosiaalisen median viesteistä (IBM Cloud Education 2021). Yritys voi esimerkiksi haluta seurata asiakkaiden reaktioita sosiaalisessa mediassa automaattisesti, jotta varsinkin negatiiviseen palautteeseen pystytään reagoimaan nopeasti (Sentiment and Emotion... s.a.). Tunneanalyysien perusteella saatua tietoa voidaan käyttää hyväksi myös mm. tuotesuunnittelussa ja mainoskampanjoissa (IBM Cloud Education 2021). Tällaisten analyysien hyödyntäminen voi täten parantaa yrityksen myyntiä, markkinointia ja kommunikointia asiakkaiden kanssa (Yao ym. 2018, 27).

Esimerkki 3: Kysymyksiin vastaaminen ja siihen liittyviä käyttötapauksia

Kysymyksiin vastaamista käytetään yleisesti keskustelupohjaisten sovellusten rakentamiseen, joihin kuuluvat mm. chattibotit, virtuaaliset avustajat sekä jotkin sosiaalisen median sovellukset (Microsoft 2021). Virtuaaliset avustajat, kuten Siri ja Alexa, käyttävät äänikomentojen tunnistamiseen puheentunnistusta ja vastauksiinsa luonnollisen kielen luomista. Chattibotit taas käsittelevät puheen sijasta tekstiä (kuva 7). (IBM Cloud Education 2021.) Kysymyksiin vastaamisen käyttötapauksiin voi siis kuulua mikä tahansa keskustelusovellus, joka kommunikoi käyttäjän kanssa luonnollisella kielellä vastatakseen tämän kysymyksiin. Sitä voi käyttää mm. asiakastuen apuna tai vaikka työntekijöiden tiedonhakuun yrityksessä sisäisesti. (Microsoft 2021). Yrityksen asiakkaalla voi olla myös esimerkiksi kysymyksiä hänen sopimukseensa tai johonkin yrityksen tuotteeseen liittyen. Kysymyksiin vastaaville malleille voidaan syöttää mm. yrityksen toimintaan tai tuotteisiin liittyvää tietoa, jotta mallit osaavat vastata esitettyihin kysymyksiin paremmin. (Question Answering API s.a.)



Kuva 7. Kysymyksiin vastaava chattibotti (JivoChat s.a.)

Kuten kuvasta 7 näkee, asiakkaita voi kiinnostaa esimerkiksi tuotteiden koko tai niiden sopivuus, tai heillä voi olla kysymyksiä tilauksen toimitukseen liittyen. Tässä kuvaesimerkissä kysymyksiin vastaava chattibotti koittaa ohjata vastauksiensa perusteella asiakkaita heidän kysymyksiinsä liittyvän tiedon äärelle. (JivoChat s.a.)

On myös hyvä mainita, että kaikki chattibotit eivät käytä luonnollisen kielen käsittelyä tai ole muutenkaan tekoälypohjaisia. Sen sijaan ne saattavat perustua kokonaan sääntöihin. (Diaz 2017.) Sääntöpohjaisista chattiboteista poiketen, tekoälypohjainen chattibotti osaa analysoida sille syötettyä dataa. Mitä enemmän chattibotilla on sitä käytössään, niin käyttäjän kysymyksistä, kuin myös sille tarjotusta kontekstista, sitä tarkempia ja parempia vastauksia se voi antaa. Tämän datan ja analyysiensa perusteella chattibotti koittaa ymmärtää mitä käyttäjä haluaa tietää, antaakseen kysymykseen sopivan ja merkittävän vastauksen. (Akerkar 2019, 13.)

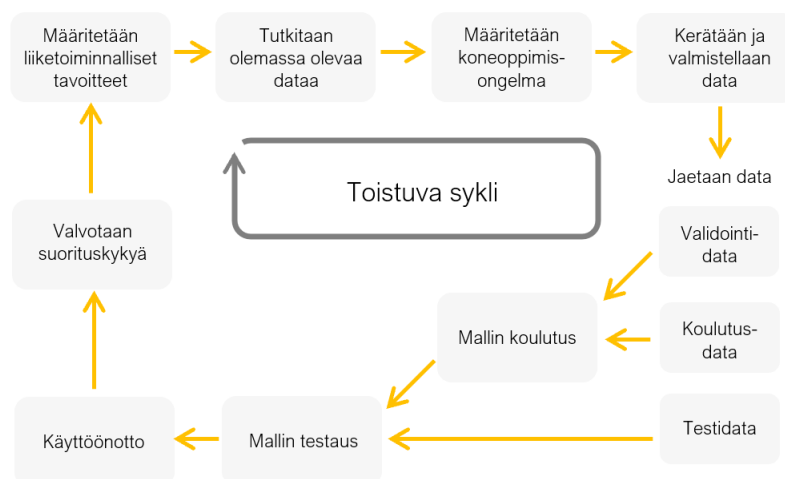
2.4 Lähestymistavat ja työkalut luonnollisen kielen käsittelyyn

Luonnollisen kielen käsittelyyn on olemassa erilaisia lähestymistapoja, mutta tämän luvun ensimmäisessä alaluvussa keskitytään avaamaan sääntö-, koneoppimis- ja syväoppimispohjaisia lähestymistapoja. Toisessa alaluvussa käsitellään erilaisia työkaluja luonnollisen kielen käsittelyyn ja kolmannessa alaluvussa kerrotaan vielä API-lähestymistavasta ja NLP Cloud API:sta, jota käytetään tämän opinnäytetyön asiakasprojektin toteutuksessa.

2.4.1 Lähestymistavat

Varhaisimmat luonnollisen kielen käsittelyyn liittyvät sovellukset olivat kokonaan käsin koodattuja, sääntöihin perustuvia järjestelmiä, jotka pystyivät suorittamaan tiettyjä luonnollisen kielen käsittelyyn liittyviä tehtäviä, mutta ne eivät skaalautuneet helposti data määrien lisääntyessä (IBM Cloud Education 2021). Sääntöpohjaisella lähestymistavalla tarkoitetaan sitä, että ohjelmoijan tulee kirjoittaa manuaalisesti kaikki ohjelman säännöt. Sääntöpohjaisilla lähestymistavoilla on kuitenkin vielä tänäkin päivänä käyttöä joissain luonnollisen kielen käsittelyyn liittyvissä vaiheissa ja tehtävissä. Niitä käytetään edelleen paljon esimerkiksi datan esikäsittelyssä, kuten tokenisaatiossa, ja kielioppiin liittyvissä tehtävissä, kuten sanaluokkajäsennyksessä, jossa tokeneille määritetään sanaluokat. (AltexSoft 2021.)

Kone- ja syväoppimisen malleja on helpompi skaalata suurempiin datamääriin ja niiden avulla pystytään ratkaisemaan monimutkaisiakin ongelmia (IBM Cloud Education 2021). Koneoppiminen on edelleen ehkä yleisin tapa lähestyä luonnollisen kielen käsittelyä (AltexSoft 2021). Kuvassa 8 on nähtävillä tyypillisen koneoppimisprosessin askeleet.



Kuva 8. Tyypillisen koneoppimisprosessin askeleet (Yao ym. 2018, 137)

Kuvasta 8 on nähtävillä, että liiketoiminnalliset tavoitteet määritetään prosessissa ensimmäisenä. Seuraavaksi tutkitaan jo olemassa olevaa yrityksen dataa ja tämän jälkeen ratkaistava ongelma määritetään koneoppimisen näkökulmasta. Kun määritykset on saatu tehtyä, dataa voidaan kerätä lisää tarvittaessa ja se esikäsitellään. Data jaetaan tämän jälkeen kolmeen osaan: koulutusdataan, testidataan ja validointidataan. Koulutusvaiheessa mallille syötetään koulutusdataa sekä validointidataa, jota käytetään, kun mallia aletaan hienosäätämään koulutuksen aikana. Tämän jälkeen mallia testataan vielä testidatalla, jota malli ei ole käyttänyt koulutuksessa, jotta voidaan nähdä, että kuinka hyvin malli suoriutuu tehtävästä uudella datalla. Kun malli on valmis, se otetaan käyttöön ja sen suoriutumista seurataan. Usein tämä prosessi on toistuva sykli, ja mallia kannattaa päivittää ajan kuluessa uudella datalla, jotta sen korkea suorituskyky ei kärsi datassa tai liiketoimintaprosesseissa tapahtuvien muutoksien takia. (Yao ym. 2018, 137–141.) Koneoppimisen haasteena on usein datan esikäsitteilyyn kuluva aika, ja datan tärkeiden ominaisuuksien löytäminen voi olla joskus monimutkaista. Siksi monet luonnollisen kielen käsittelyn tutkimukset koskevat tällä hetkellä koneoppimisen alahaaraa, syväoppimistä. (AltexSoft 2021.)

Syväoppiminen, tai syvät neuroverkot, on koneoppimisen haara. Sitä kutsutaan syväksi, koska se sisältää monia toisiinsa liittyviä kerroksia. Syöttökerrokset vastaanottavat dataa ja lähettävät sen piilotettuihin kerroksiin, jotka suorittavat matemaattisia laskelmia. Syväoppimisprosessissa käytettävät askeleet,

eivät juurikaan eroa koneoppimisprosessin askelista, mutta joitain eroja toki on. Neuroverkot ovat niin tehokkaita, että niille syötetään raakadataa ilman erillisesti valittuja ominaisuuksia ja ne oppivat itsenäisesti mitkä ominaisuudet ovat tärkeitä. (AltexSoft 2021.) Syväoppiminen on vienyt luonnollisen kielen käsittelyn aivan uudelle tasolle (AltexSoft 2021). Tutkimusta tällä saralla tehdään jatkuvasti ja siihen kuuluu monia trendikkäitä menetelmiä (AltexSoft 2021; Mayo 2018). Syväoppimisen haasteena on kuitenkin se, että neuroverkkomallien kouluttaminen vaatii valtavia määriä dataa ja näiden suurien data määrien käsittelyyn ja laskelmien tekemiseen tarvitaan koneilta paljon laskentaresursseja (AltexSoft 2021).

Kaikki nämä erilaiset lähestymistavat luonnollisen kielen käsittelyyn ovat käytössä tänäkin päivänä, ja jokaisesta lähestymistavasta voi olla hyötyä tietyissä luonnollisen kielen käsittelyyn liittyvissä käyttötapauksissa (AltexSoft 2021). Niitä voidaan myös käyttää ja tutkia samanaikaisesti parhaan mahdollisen hyödyn saavuttamiseksi (Mayo 2018). Usein liiketoiminnalliset haasteetkin voidaan ratkaista monilla eri tavoilla (Yao ym. 2018, 21).

2.4.2 Työkalut

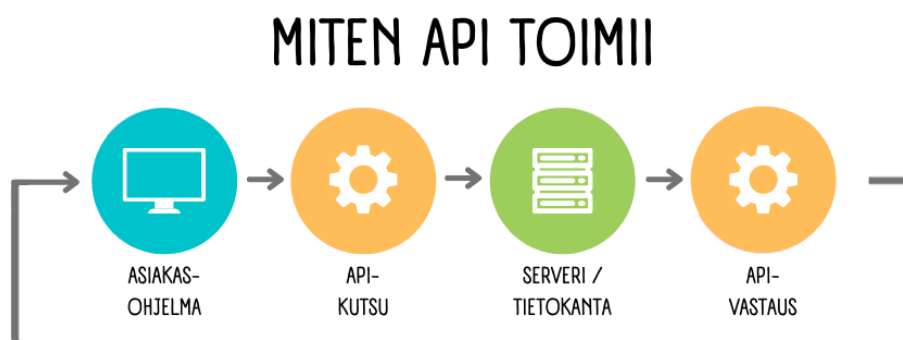
Luonnollisen kielen käsittelyyn liittyvät työkalut ja erilaiset lähestymistavat voivat auttaa yrityksiä löytämään uusia havaintoja ja ideoita liiketoimintaan liittyen, esimerkiksi yrityksen sosiaalisen median viesteistä, tuotearvosteluista tai sähköpostiviesteistä. Luonnollisen kielen käsittelyyn tarkoitettujen työkalujen käytön voi aloittaa esimerkiksi avoimen lähdekoodin kirjastojen tai SaaS-työkalujen (engl. Software as a Service) kautta. (Wolff 2020.)

Ohjelmistokehittäjille suunnatut avoimen lähdekoodin kirjastot luonnollisen kielen käsittelyyn ovat joskus melko monimutkaisia ymmärtää, ja niissä tarvitaan usein kokemusta koneoppimisesta. Kehittäjien työ ei ole kuitenkaan ilmaista ja avoimen lähdekoodin kirjastojen avulla tehty kehitystyö saattaa viedä paljon aikaa. Toisaalta avoimen lähdekoodin kirjastot ovat itsessään ilmaisia. Ne ovat usein myös joustavia ja mahdollistavat hyvin erilaisia räätälöityjä toteutuksia. (Wolff 2020.) Esimerkiksi Python-ohjelmointikielille on tehty monia työkaluja ja kirjastoja, koska se on yksi parhaiten soveltuvista ohjelmointikielistä suurien data määrien, niin kutsutun massadatan (engl. Big Data), käsittelyyn

(Fedak 2018). Python tarjoaa monia työkaluja ja kirjastoja erilaisiin luonnollisen kielen käsittelyyn liittyviin tehtäviin, kuten esimerkiksi NLTK (engl. Natural Language Toolkit), Gensim, spaCy ja TextBlob (IBM Cloud Education 2021; Wolff 2020). SaaS-työkalut luonnollisen kielen käsittelyyn ovat taas yleensä kustannustehokkaita ja nopeita ottaa käyttöön, eivätkä ne välttämättä vaadi ohjelmointikoodia lainkaan, tai ne vaativat sitä vain vähän. Tehokkaat SaaS-työkalut ovat pilvipohjaisia ratkaisuja, ja niiden avulla on mahdollista hyödyntää monimutkaisiakin esikoulutettuja malleja luonnollisen kielen käsittelyyn, joita ei olisi välttämättä mahdollista hyödyntää avoimen lähdekoodin kirjastojen avulla erinäisten rajoitusten, kuten esimerkiksi vaadittavan laskentatehontakia. (Wolff 2020.) SaaS-työkaluihin kuuluu erilaisia alustoja, kuten IBM Watson ja MonkeyLearn, sekä erinäiset API:t, kuten Google Cloud Natural Language API, OpenAI API ja NLP Cloud API, joiden avulla on helppo yksinkertaistaa ja nopeuttaa kehitystyötä, ja jolloin myös erilaisten sovellusten kehitystyön ja käyttöönoton kustannukset voivat pysyä pienempinä (Google Cloud s.a.; IBM 2021; NLP Cloud s.a.; OpenAI 2021; Wolff 2020).

2.4.3 API-lähestymistapa ja NLP Cloud API

Koska tekoälypohjaisten, luonnollisen kielen käsittelyä käyttävien, sovellusten rakentaminen alusta loppuun asti voi olla isoissakin yrityksissä haasteellinen ja aikaa vievä tehtävä, API:t voivat olla oiva ratkaisu nopeuttamaan ja helpottamaan tätä prosessia (Yao ym. 2018, 85). API (engl. Application Programming Interface), eli ohjelmointirajapinta, mahdollistaa tiedonsiirron kahden ohjelman välillä (What is API... 2021). Ohjelmistokehittäjä voi tehdä niin sanotun pyynnön API:in lähettääkseen, hakeakseen, päivittääkseen tai poistaakseen tietoa (kuva 9) (Wyatt 2022).



Kuva 9. API:n toiminta (Calaméo 2021)

Kuten kuvasta 9 näkee, ns. asiakassovellus tekee ensin API pyynnön serverille, joka kommunikoi mahdollisen tietokannan kanssa, ja jonka jälkeen API palauttaa vastauksen asiakassovellukselle (Calaméo 2021). API:t voivat olla erittäin tehokkaita työkaluja, ja siksi tätä lähestymistapaa päätettiin käyttää myös tässä opinnäytetyössä luonnollisen kielen käsittelyyn. Ne ovat myös yleensä hyvin dokumentoituja. Nykyiset luonnollisen kielen käsittelyn liittyvät tehtävät vaativat yleensä syväoppimista ja suuria datamääriä. Mallit ovat myös raskaita, ja erittäin kalliita kouluttaa ja ylläpitää. (AltexSoft 2021.) Yritykset, jotka tarjoavat pääsyn näihin esikoulutettuihin malleihin SaaS-palveluina, mahdollistavat edistyneiden mallien käytön nykyään melkein kenelle tahansa, kun aikaisemmin ne olivat vain suurten toimijoiden saatavilla (Turner 2022).

NLP Cloud on suhteellisen uusi yritys, joka on perustettu elokuussa 2020 (Crunchbase s.a.). Yrityksen tavoitteena on helpottaa luonnollisen kielen käsittelyyn liittyvien projektien testaamista ja siirtämistä tuotantoon, mikä voi olla joskus kehittäjille haasteellinen tehtävä, koska se vaatii vahvoja ohjelmointi-, DevOps- ja tekoälytaitoja sekä kapasiteettia (Crunchbase s.a.; Lawton 2021.) Tarjoamalla helpon ja vakaan API-rajapinnan kautta kehittäjille tekoäly- tai luonnollisen kielen käsittelyn liittyviä malleja, NLP Cloud pyrkii vähentämään näitä kehittäjien haasteita (Lawton 2021).

NLP Cloud API:n avulla kehittäjät pääsevät käsiksi monimutkaisiin esikoulutettuihin avoimen lähdekoodin malleihin, hankkimalla joko ilmaisen tai jonkun maksullisista tilaussuunnitelmista. Ilmaista versiota ei saa käyttää tuotannossa, mutta sen avulla on mahdollista päästä testaamaan osaa malleista ilman maksua. Maksulliset pakettisuunnitelmat vaihtelevat tällä hetkellä 29–499 dollarin välillä kuukaudessa, jos malleja halutaan pyörittää prosessorilla eli CPU:lla, ja 99–699 dollarin välillä, jos malleja pyöritetään grafiikkaprosessorilla eli GPU:lla, mikä on noin 10 kertaa nopeampaa. NLP Cloud tarjoaa myös muita tilaussuunnitelmia, kuten käyttöön perustuvan veloituksen, erilaisia vaihtoehtoja mallien hienosäätöön tai omien mallien käyttöön API:n kautta sekä arkaluonteisille sovelluksille kustomoitavia vaihtoehtoja. Yritys myös mainostaa, että muita tilaussuunnitelmia voidaan luoda ja kustomoida asiakkaiden

tarpeisiin sopiviksi. Kaikkiin suunnitelmiin on myös mahdollista yhdistää monikielinen lisäosa, jonka veloitus perustuu käytön määrään. Kaikki tilaus suunnitelmat voidaan myös peruuttaa milloin vain. (Pricing s.a.)

NLP Cloud API:a on dokumentaation mukaan mahdollista käyttää asiakaskirjastojen kautta tällä hetkellä seuraavilla ohjelmointikielillä: cURL, Python, Ruby, GO, Node.js, PHP ja C#. API-kutsuja voi toki tehdä myös omien funktioiden kautta tai esimerkiksi Postman-sovelluksella. (Introduction s.a.) API-kutsuille on olemassa erilaisia pääteisteitä riippuen siitä, että mitä mallin avulla haluaa tehdä, eli esimerkiksi tunneanalyysille ja tekstin yhteenvedolle on omat pääteisteensä. (Endpoints s.a.) NLP Cloud API:n kautta voi käyttää tällä hetkellä 44 erilaista esikoulutettua mallia, mukaan lukien erilaiset kielimallit ja kieleltä toiselle kääntämiseen tarkoitettua Opus-MT -mallit, joita on kehitetty Helsingin yliopistossa (Language Technology... 2022; Models List s.a.).

NLP Cloud API:n avulla kehittäjän ei siis tarvitse itse toteuttaa mm. joitain tekstin esikäsittelyn vaiheita, kuten tokenisaatiota, vaan se tapahtuu pinnan alla. NLP Cloud API:ssa on kyllä myös olemassa tokenisaatioon pääteiste, missä tällaista käsittelyä voi tehdä niin halutessaan. (Endpoints s.a.)

3 MIKROPALVELUJEN TOTEUTUS SOVELLUSTEN TEHOSTAMISEEN

Tässä luvussa käsitellään opinnäytetyön käytännön osuutta ja kerrotaan lukijalle toimeksiantajasta, asiakasprojektista, siihen toteutettavista mikropalveluista sekä asiakasprojektiin liittyvästä käytännön toteutuksesta.

3.1 Mindhive Oy

Mindhive Oy oli tämän opinnäytetyön toimeksiantaja. Mindhive Oy on vuonna 2016 perustettu mikkeliäinen ohjelmointiyritys, jonka missiona on auttaa yrityksiä menestymään digitalisaation tarjoamien mahdollisuuksien avulla. Mindhive keskittyy auttamaan eri alojen asiantuntijoita ilmaisemaan itseään tavalla, joka voidaan kääntää käyttäjäystävälliseksi ohjelmistotuotteeksi. Yrityksen tavoitteena on löytää ja käyttää tehokkaita työkaluja asiakkaalle räätälöityihin ratkaisuihin ja ratkaista asiakkaan ongelma tehokkaammin panostaen loistavaan käyttökokemukseen. Tämän lisäksi Mindhive pitää tärkeänä ja panostaa

asiakkaiden positiiviseen kokemukseen luoden samalla pitkäaikaisia yhteistyökumppanuuksia ja vahvoja verkostoja. Mindhive pystyy tarjoamaan asiakkailleen asiantuntemustaan niin ohjelmisto- ja käyttäjäkokemussuunnittelun, palvelumuotoilun kuin tekoälyn asiantuntijoiden avulla.

3.2 Asiakasprojekti

Methodix Oy omistaa eDelphi-ohjelmiston, digitaalisen alustan, jossa käyttäjät voivat hyödyntää Delfoi-menetelmää (kuva 10). eDelphillä on käyttäjiä ja lisenssejä kaikilla mantereilla Etelämannerta lukuun ottamatta. Vuonna 2021 käyttäjiä oli 135 eri maasta. eDelphiä on kehitetty 20 vuoden aikana yhdessä suomalaisten tulevaisuuden tutkimuslaitosten, mukaan lukien Turun yliopiston Tulevaisuuden tutkimuskeskuksen ja Tulevaisuuden tutkimuksen seura ry:n, kanssa. Sitä on käytetty useissa suurissa projekteissa ja väitöskirjoissa niin Suomessa kuin ulkomailla, ja sillä on jo vankka käyttäjäpohja akateemisessa maailmassa sekä julkisella sektorilla.

DELFOI-MENETELMÄ

Asiantuntijapaneelien avulla kohti parempaa liiketoimintaa

Delfoi-menetelmä on viestintäteknikka, joka on alun perin kehitetty järjestelmälliseksi ja interaktiiviseksi ennustemenetelmäksi, ja joka perustuu asiantuntijapaneelin antamiin mielipiteisiin.

Tekniikkaa voidaan käyttää kasvokkain tapahtuvissa tapaamisissa, mutta myös etätapaamisissa ja virtuaaliympäristöissä, joiden käyttö on korostunut erityisesti koronapandemian myötä.

KÄYTTÖ JA PERIAATE



Delfoi-menetelmää on käytetty laajasti liiketoiminnan ennustamiseen, ja sillä on tiettyjä etuja muihin ennustamismenetelmiin verrattuna. Menetelmä perustuu periaatteeseen, että valikoidun asiantuntevan yksilöryhmän ennusteet ja mielipiteet ovat tarkempia kuin muiden ryhmien, joita ei ole koottu järjestelmällisesti.

PROSESSI



Asiantuntijat vastaavat kyselyihin yleensä kahdella tai useammalla kierroksella.

Jokaisen kierroksen jälkeen ohjaaja eli fasilitaattori antaa anonymisoidun yhteenvedon asiantuntijoiden edellisen kierroksen ennusteista sekä perusteluista, joita he esittivät arvionsa tueksi. Asiantuntijoita kannustetaan näin ollen arvioimaan uudelleen ja muokkaamaan aiempia vastauksiaan muiden paneelinsa jäsenten vastausten valossa.

Uskotaan, että tämän prosessin aikana vaihtelu vastauksissa vähenee ja ryhmä lähentyy luonnollisesti kohti "oikeaa" vastausta.

Lopuksi prosessi pysäytetään ennalta määrättyjen pysäytyskriteerien täytymisen jälkeen (esim. kierrosten lukumäärä, yhteisymmärryksen saavuttaminen, tulosten vakaus), ja viimeisten kierrosten keskimääräiset tai mediaanipisteet sekä mielipiteet määräävät tulokset.

Kuvasta 10 on nähtävissä, että delfoi-menetelmää on käytetty esimerkiksi laajasti liiketoiminnan ennustamiseen. Menetelmässä asiantuntijapaneelit vastaavat kysymyksiin eri kierroksilla, ja lopullisena tavoitteena on saavuttaa tarkkoja ennusteita kulloiseenkin aihealueeseen liittyen.

Yhteistyössä Mindhive Oy:n kanssa yritys kehittää ohjelmistonsa uutta versiota, nimeltään xDelphi. Tämä ohjelmiston uusi versio ei ole pelkkä päivitys, vaan täydellinen muutos alustan liiketoimintamalliin, visuaalisuuteen ja käyttökokemukseen, jonka tavoitteena on laajentaa ohjelmiston kohdeyleisöä. Sallassapitosopimuksen takia opinnäytetyössäni ei käsitellä uudesta alustasta tarkkoja yksityiskohtia. Kuitenkin ydinkonsepti, jonka parissa työskentelin, on pyritty esittämään opinnäytetyössä niin, että opinnäytetyön laatu ei kärsi sallassapitosopimuksesta huolimatta.

Lyhyesti kerrottuna, uusi kehitteillä oleva ohjelmisto on pilvipalvelupohjainen alusta, joka toimii SaaS (engl. Software as a Service) -liiketoimintamallilla. Tämä edellyttää modernia ohjelmistoarkkitehtuuria ja kehitystekniikkaa. Käyttäjät voivat halutessaan ostaa sovellukseen lisäominaisuuksina asiantuntijatyökaluja tilaamalla edistyneempiä suunnitelmia. Mitä arvokkaampia ominaisuuksia alusta voi tarjota käyttäjilleen, sitä suuremmalla todennäköisyydellä voidaan alustalle saada maksullisia käyttäjiä. Opinnäytetyöni aihe ja toteutus saivat alkunsa tässä yhteydessä. Opinnäytetyön tekemisen aikana toteutettiin sarja kokeiluja tekoälyllä ja luonnollisen kielen käsittelyllä, hyödyntäen NLP Cloud API:a. Kokeilujen tarkoituksena oli tehostaa kehitettävää ohjelmistoa. Kokeilujen toteutus osana xDelphi- ja Aavistus-alustoja muodostaa tämän opinnäytetyön käytännön osuudessa esitellyn tutkimustapauksen.

3.3 Toteutettavat mikropalvelut

Mikropalvelulla tarkoitetaan yleisesti sovelluskehityksen lähestymistapaa, missä sovellus koostuu pienistä itsenäisistä osioista. Asiakasprojektiin valittiin kolme toteutettavaa mikropalvelua, joiden kehittäminen oli tarkoitus aloittaa tämän opinnäytetyön aikana. Valitut mikropalvelut olivat kysymyksiin vastaava chattibotti, automaattisten yhteenvetojen luominen sekä kommenttien automaattinen tulkinta (taulukot 2, 3 ja 4).

Taulukko 2. Ensimmäinen valittu mikropalvelu asiakasprojektiin

Mikropalvelu 1: Chattibotti vastaamaan kysymyksiin sovelluksissa	
Tavoite:	Auttaa käyttäjiä navigoimaan sovelluksessa ja sen mahdollisuuksissa.
Miten:	Sovelluksen jokaisessa näkymässä on kysymyksiin vastaava chattibotti, joka on valmis vastaamaan kysymyksiin.
Askeleet:	<ol style="list-style-type: none"> Syötetään mallille dataa API:n kautta. Testataan Question-, Chatbot- ja Generation-päätepisteistä erilaiset mallit: <ul style="list-style-type: none"> Question: Roberta, fast GPT-J ja fine-tuned GPT-Neox Chatbot: fast GPT-J ja fine-tuned GPT-Neox Generation: GPT-mallit Kustomoidaan Aavistuksen kontekstilla (Delfoi-menetelmän dokumentaatio). Pystyy vastaamaan Delfoi-menetelmää koskeviin kysymyksiin.
Validointi:	Oma arvio eri päätepisteiden ja mallien toiminnasta, suoriutuuko hyvin vai ei.

Taulukko 3. Toinen valittu mikropalvelu asiakasprojektiin

Mikropalvelu 2: Automaattisten yhteenvetojen luominen	
Tavoite:	Auttaa fasilitaattoreita tekemään yhteenvetoja keskusteluista.
Miten:	Yhteenveto-alitoiminto AI-työkalujen alla Facilitate-tilan oikeassa reunassa olevassa osiossa.
Askeleet:	<ol style="list-style-type: none"> Syötetään mallille dataa API:n kautta. Testataan Summarization- ja Generation-päätepisteistä erilaiset mallit: <ul style="list-style-type: none"> Summarization: Bart, Pegasus, fast GPT-J ja fine-tuned GPT-Neox Generation: GPT-mallit Kustomoidaan Generation-päätepisteessä esimerkeillä, jotka sopivat Aavistukseen. Pystyy muodostamaan tiivistelmiä keskusteluista.
Validointi:	Oma arvio eri päätepisteiden ja mallien toiminnasta, suoriutuuko hyvin vai ei.

Taulukko 4. Kolmas valittu mikropalvelu asiakasprojektiin

Mikropalvelu 3: Kommenttien automaattinen tulkinta	
Tavoite:	Luokittelee tekstiä yleisen aiheen ja tunteen mukaan.
Miten:	Tunneanalyysi-alitoiminto AI-työkalujen alla Facilitate-tilan oikeassa reunassa olevassa osiossa.
Askeleet:	<ol style="list-style-type: none"> Syötetään mallille dataa API:n kautta. Testataan Sentiment- ja Generation-päätepisteistä erilaiset mallit: <ul style="list-style-type: none"> Sentiment: Distilbert finetuned, Distilbert emotion Generation: GPT-mallit Kustomoidaan Generation-päätepisteessä esimerkeillä, jotka sopivat Aavistukseen. Pystyy muodostamaan tulkintoja kommentteista.
Validointi:	Oma arvio eri päätepisteiden ja mallien toiminnasta, suoriutuuko hyvin vai ei.

Kuten taulukoista 2, 3 ja 4 pystyy näkemään, kullekin toteutettavalla mikropalvelulle asetettiin tavoite, määritettiin miten ja mihin mikropalvelu toteutetaan, mitkä ovat vaadittavat askeleet ja miten tulokset validoidaan. Menettelytapana oli siis syöttää tekstiä mallille API:n kautta ja tarkoituksena oli arvioida API:n palautusta. Mikropalvelujen avulla voidaan mm. tehostaa sovelluksia, liittämällä esimerkiksi sovellukseen chattibotin, tai muita toiminnallisuuksia, jotka tuovat sovellukseen uutta lisäarvoa. Mikropalvelut ovat yleensä pieniä, generisiä ja uudelleen käytettäviä komponentteja, ja niiden etuna on, että niitä on helpompi käyttää uudelleen erilaisiin tilanteisiin ja sovelluksiin, kuin suurempina kokonaisuuksina toteutettuja komponentteja. Mikropalvelut voivat siis nopeuttaa sovellusten kehitystä, ja niiden avulla sovelluksia on helpompi skaalata tarpeen mukaan.

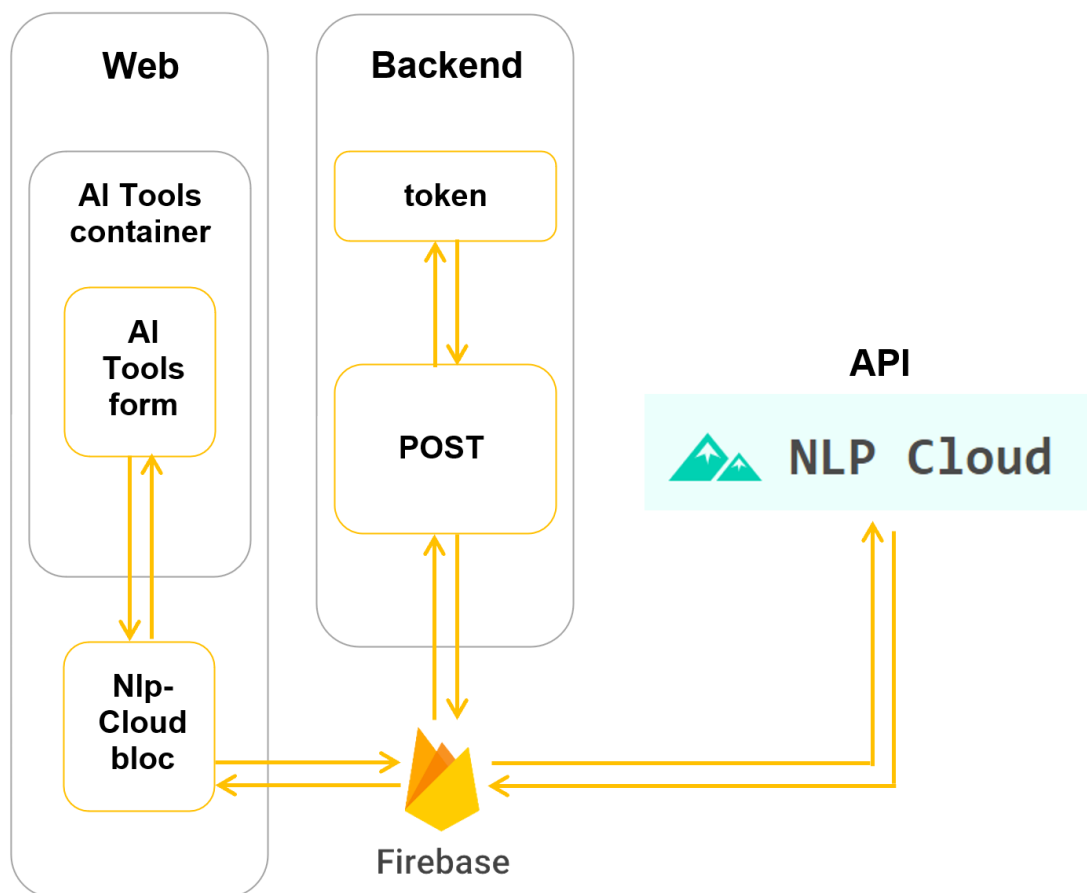
3.4 Toteutus

Mikropalvelujen toteutus aloitettiin suunnitteluvaiheella. Suunnitteluvaiheen aluksi tehtiin vertailua kahden palveluntarjoajan välillä, joihin kuului OpenAI ja NLP Cloud. Vertailussa tarkasteltiin palveluntarjoajien hintoja, tarjolla olevia API-ominaisuuksia, sekä niiden käytännöllisyyttä. Vertailussa huomioitiin

myös OpenAI API:n käyttörajoitteet. OpenAI tarkkailee ja estää mahdollisesti haitallisen sisällön luomisen API:lla, koska se sisältää tehokkaita malleja, jotka pystyvät luomaan tekstiä tai ohjelmointikoodia ihmisen kaltaisesti todella vakuuttavalla tavalla. OpenAI tarkastaa myös jokaisen sovelluksen, joka käyttää OpenAI API:a, ennen julkaisua kriteeriensä mukaisesti varmistaakseen, että julkaistava sovellus noudattaa OpenAI:n käytäntöjä ja turvallisuusvaatimuksia. Vertailussa oli mukana OpenAI API:n tarjoama GPT-3 malliperhe, sekä NLP Cloud API:n tarjoamat erilaiset mallit, kuten esimerkiksi GPT-J ja GPT-NeoX 20B. OpenAI:n tehokkain GPT-3 malli, eli Davinci, on esimerkiksi huomattavasti suurempi parametrimäärältään kuin GPT-J tai GPT-Neox 20B, mikä viittaa siihen, että mallin neuroverkossa on huomattavasti enemmän neuroneita ja niiden välisiä yhteyksiä. Tehokkaimmassa GPT-3 mallissa on noin 175 miljardia parametria, kun taas GPT-J mallissa niitä on noin 6 miljardia ja GPT-Neox 20B mallissa noin 20 miljardia. Tästä voitaisiin päätellä, että tehokkain GPT-3 malli on monimutkaisempi kuin GPT-J ja GPT-NeoX 20B, ja se saattaisi ainakin joissain tapauksissa palauttaa parempia tuloksia. Tästä huolimatta, palveluntarjoajaksi valittiin NLP Cloud, koska sen tarjoaman API:n kautta oli mahdollista testata ja käyttää huomattavasti suurempaa määrää erilaisia malleja, ja koska sen käyttö saattaisi olla edullisempaa erilaisten tilauspakettien ansiosta. NLP Cloud API:n käyttö on myös tuotannollisesta näkökulmasta helpompi vaihtoehto, koska julkaistavia sovelluksia ei tarvitse erikseen hyväksyttää palveluntarjoajalla, toisin kuin OpenAI API:a käytettäessä. Molemmat API:t on kuitenkin rakennettu hyvin samankaltaisesti, ja asiakasprojektiin valitut mikropalvelut voisivat hyödyntää hyvin pienillä muutoksilla kumpaa vaan näistä palveluista.

Seuraavaksi suunnitteluvaiheessa pohdittiin, että minkälaiset luonnollisen kielen käsittelyn tehtävät ja mikropalvelut sopisivat kyseessä olevaan asiakasprojektiin. Tämän pohdinnan perusteella valittiin kolme mikropalvelua, joita aloitettiin toteuttamaan tämän opinnäytetyön aikana. Valitut mikropalvelut olivat kysymyksiin vastaava chattibotti, automaattisten yhteenvetojen luominen ja kommenttien automaattinen tulkinta. Opinnäytetyön yhtenä tavoitteena oli aloittaa NLP Cloud API:n kanssa kommunikoivan komponenttikirjaston luominen joihinkin luonnollisen kielen käsittelyyn liittyviin tehtäviin, joten sen luominen aloitettiin näistä asiakasprojektiin sopivista mikropalveluista. Komponentti-

kirjasto toteutettiin TypeScript-ohjelmointikielellä. Salassapitosopimuksen johdosta, itse ohjelmointikoodia ei tulla esittelemään tässä opinnäytetyössä. Komponenttikirjaston luomista ja kehittämistä tullaan jatkamaan tämän opinnäytetyön jälkeenkin toimeksiantajalla. Kuvasta 11 voidaan nähdä, kuinka toteutetut komponentit, Firebase ja NLP Cloud API välittävät dataa ja toiminnallisuksia toisilleen.



Kuva 11. Tietojen ohjaaminen ja kierto eri osioiden ja ohjelmien välillä

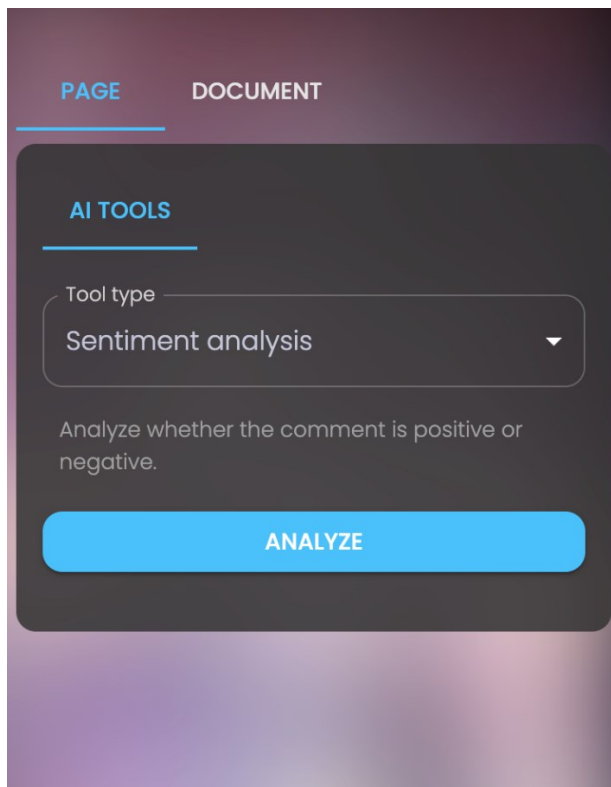
Kuvasta 3 on nähtävillä, että Web-kansioon on toteutettu AI Tools-osio, jonka sisälle tuodaan erillisestä tiedostosta AI Tools-lomake. Kun lomakkeessa valitaan työkalun tyyppi ja painetaan Analyze-painiketta, niin lomakkeella oleva data alkaa siirtymään eteenpäin. Jokaiseen käytettyyn API:n päätepisteeseen tehtiin NlpCloudBloc-luokkatiedostoon omat yksityiset ja julkiset funktiot. Päätepisteitä olivat siis Chatbot, Generation, Question, Sentiment ja Summarization. Jokaisesta NlpCloudBloc-funktiosta tulee Firebase:n pilvifunktio (engl. cloud function). Backend-kansioon toteutettiin myös POST-kutsut jokaiselle

päätepisteelle, ja API kutsuissa tarvittava tokeni, haettiin kutsuun mukaan erillisestä tiedostosta. Nämäkin toimivat pilven kautta. Firebase:n pilvifunktiot tekevät API-kutsun, ja data palaa reittiä pitkin takaisin sovellukseen. Kaikki tarvittavat tyypitykset tehtiin erillisiin tiedostoihin ja koodi kommentoitiin niin, että kommentit sisälsivät ohjeet siitä, mitä kuhunkin API:n päätepisteeseen voi lähettää ja mitä sieltä palautuu, sekä mitkä mallit niissä on käytössä ja mitä eroa niillä on.

Käyttöliittymään toteutettiin AI-tools lomake, jonka valikosta voi valita halutun analysointityökalun. Chattibotin käyttöliittymäelementtiä ei toteutettu tämän opinnäytetyön aikana, vaan sen toteutus jäi vielä seuraaviin kehitysvaiheisiin. Toteutuksen aikana kerättiin ja valittiin myös testidataa mallien testaamista varten.

4 TULOKSET

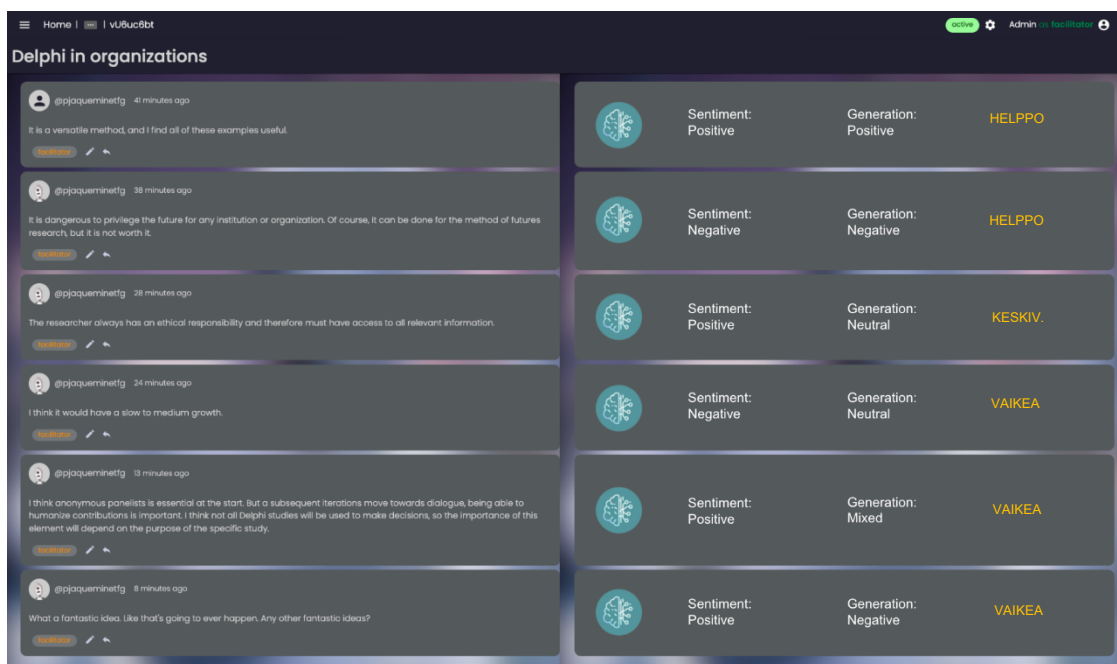
Tässä luvussa esitellään opinnäytetyön toteutuksen tulokset. Käyttöliittymään toteutettiin AI-työkalujen alle lomake, jonka valikossa oli tunneanalyysi- ja yhteenvedo alitoiminto, Facilitate-tilan oikeaan reunaan (kuva 12).



Kuva 12. AI Tools-osio Facilitate-tilassa

Kuten kuvasta 12 näkee, valikosta voi valita analysointityökalun tyyppin ja valikon alla on valintaa kuvaava ohjeistava teksti. Analyze-painiketta painamalla aletaan toteuttamaan API-kutsua. AI-työkaluilla on lomaketta ympäröivässä komponentissa oma välilehtensä, koska komponenttiin tullaan lisäämään toimeksiantajan puolesta muitakin työkaluja sovelluskehityksen jatkovaiheissa.

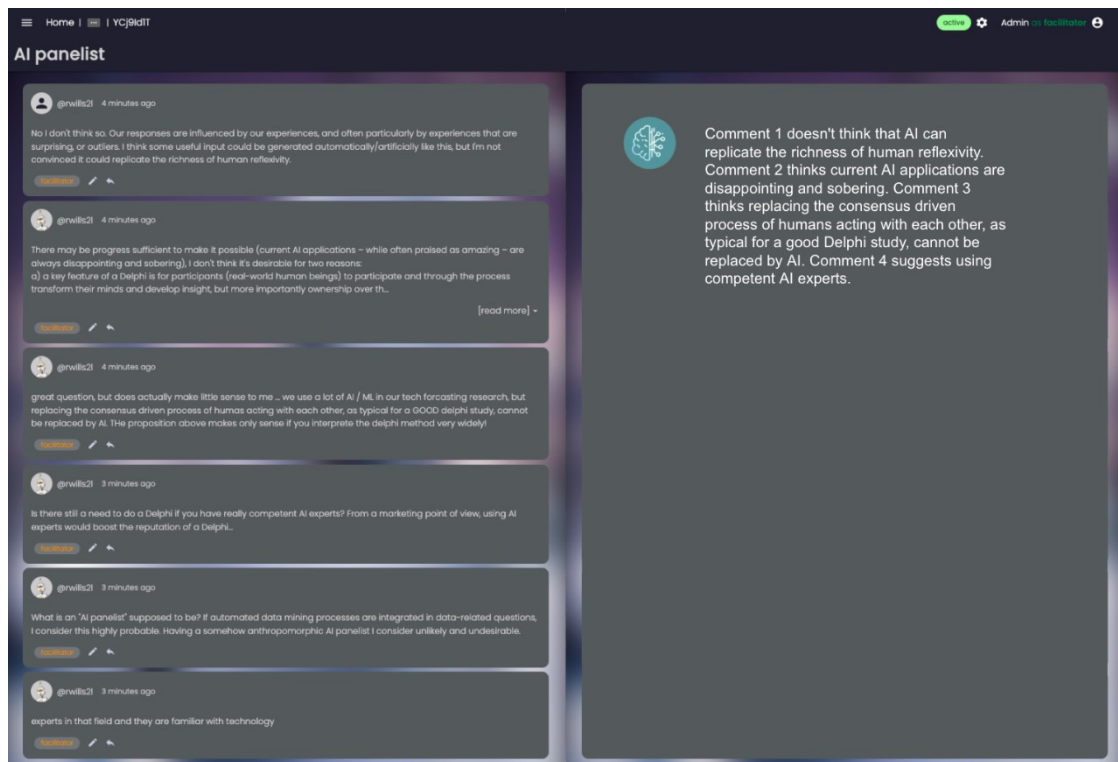
Opinnäytetyössä testattiin erilaisia malleja API:n päätepisteistä, jotka liittyivät asiakasprojektin toteuttamiseen. Yhteensä erilaisia API-kutsuja ja testejä tehtiin 164 kappaletta, johon ei ole laskettu mukaan ihan ensimmäisiä alustavia testailuja, joita tein omalla käyttäjätunnuksellani. Yhteensä lähetettyjä tokeineita oli 41 266 kappaletta. Kaikkia testauksien tuloksia ei ollut mahdollista sisällyttää tähän opinnäytetyöhön, mutta erilaisia testejä ja API:n palauttamia vastauksia on koitettu kuitenkin tuoda monipuolisesti esille tässä opinnäytetyössä. Kuvasta 13 voi nähdä tunneanalyysiin liittyviä palautuksia API:sta.



Kuva 13. Tunneanalyysiin liittyviä palautuksia API:sta

Sentiment päätepisteen malli Distilbert Base Uncased Finetuned STT 2 English vaikutti toimivan helpompiin esimerkkeihin hyvin, mutta ei suoriutunut haasteellisemmista esimerkeistä kovin hyvin (kuva 13). Generation päätepisteessä oleva malli Finetuned GPT-Neox 20B palautti mielestäni parhaita tuloksia, kun sille pystyttiin antamaan muutamia esimerkkejä kutsua tehdessä. Fast GPT-J ja GPT-Neox 20B mallit eivät palauttaneet toivotunlaisia tuloksia oikeastaan ollenkaan testejä tehdessä. Finetuned GPT-Neox 20B malli palautti

kyllä esimerkiksi sarkastiseksi itse ajattelemastani esimerkistä Negative:n, mutta se oli mielestäni ainakin lähempänä oikeaa vastausta, kuin Sentiment-päätepiesteestä saatu tulos. Hyvien esimerkkien määrittäminen on toki oma taiteenlajinsa. Huonona puolena voidaan vielä mainita, että malli myös keksi ihan oman Mixed-luokan, mikä ei ole ehkä kovin suotavaa. Mallin valinta riippuu siis halutusta käyttötarkoituksesta. Mielestäni Sentiment päätepiesteen malli toimii suhteellisen hyvin, jos halutaan tietää vain positiivinen tai negatiivinen luokka, mutta kokonaisuutena Generation päätepiesteen Finetuned GPT-Neox 20B malli palautti mielestäni sopivimpia tuloksia. Tätä kannattaa toki jatko testata. Kuvasta 14 voi nähdä yhteenvetoon liittyvän palautuksen API:sta.



Kuva 14. Yhteenvetoon liittyvä palautus API:sta

Kuten kuvasta 14 näkee, Summarization päätepiesteen bart-large-samsum malli palautti kokonaisista keskusteluista suhteellisen hyviä tiivistelmiä. Toki se jätti joitain kommentteja kokonaan huomioimatta. Finetuned GPT-Neox 20B vaikutti lupaavimmalle yksittäisillä tekstipätkillä, ja teki ihan suhteellisen järkevän näköisiä tiivistelmiä. Testatessa kyllä huomasi myös, että joskus malli generoi itse sellaista tekstiä, mitä alkuperäisessä ei ole ehkä haettu takaa. Keskustelun tiivistäminen toimi sinänsä ihan hyvin, mutta lisätestäminen on tässäkin siis suositeltavaa. Generation päätepiesteen tulokset olivat aika

vaihtelevia, joten käytetyt esimerkit eivät olleet välttämättä kovin hyviä. Kuvassa 15 on näkyvillä kysymyksiin vastaamiseen liittyviä palautuksia API:sta.

The figure displays four separate chat interface panels, each with a dark header labeled 'Chat' and a light background. Each panel shows a sequence of messages between a user (represented by a blue speech bubble) and an AI assistant (represented by a white speech bubble with a robot icon). At the bottom of each panel is a text input field labeled 'Type the message...'. The responses are as follows:

- Panel 1:**
 - User: "Hi there! You can ask me questions related to Delphi method."
 - AI: "Hi there! You can ask me questions related to Delphi method."
 - User: "What is a Delphi method?"
 - AI: "The Delphi Method is a group forecasting technique that relies on the judgement of a panel of experts."
- Panel 2:**
 - User: "Hi there! You can ask me questions related to Delphi method."
 - AI: "Hi there! You can ask me questions related to Delphi method."
 - User: "What is the main advantage of Delphi method?"
 - AI: "The advantage of the Delphi method is that it can produce accurate forecasts from a structured group of individuals."
- Panel 3:**
 - User: "Hi there! You can ask me questions related to Delphi method."
 - AI: "Hi there! You can ask me questions related to Delphi method."
 - User: "What are the steps of the Delphi method?"
 - AI: "The Delphi method has five steps:
 1. Define the problem and target audience.
 2. Recruit a qualified panel of experts.
 3. Distribute a questionnaire to the experts.
 4. Collect and analyze the experts' responses.
 5. Present the findings to the experts.
- Panel 4:**
 - User: "Hi there! You can ask me questions related to Delphi method."
 - AI: "Hi there! You can ask me questions related to Delphi method."
 - User: "What kind of experts does the panel include?"
 - AI: "The panel of experts can include anyone with relevant knowledge or experience in the topic at hand."

The bottom two panels show unrelated questions:

- Panel 5:**
 - User: "Hi there! You can ask me questions related to Delphi method."
 - AI: "Hi there! You can ask me questions related to Delphi method."
 - User: "Who is the president of United States?"
 - AI: "This is not a question that lends itself to a scientific answer."
- Panel 6:**
 - User: "Hi there! You can ask me questions related to Delphi method."
 - AI: "Hi there! You can ask me questions related to Delphi method."
 - User: "What is the meaning of life?"
 - AI: "The meaning of life is knowing if you have reached a stable result for round 4 in Delphi."

Kuva 15. Kysymyksiin vastaamiseen liittyviä palautuksia API:sta

Kuvasta 15 voidaan nähdä, että mallit suoriutuivat suhteellisen hyvin kysymyksiin vastaamisesta. Question päätepiirteen Finetuned GPT-Neox 20B malli vaikutti lupaavimmalle, vaikka GPT-J mallilla oli kyllä joitain ihan hauskojakin ja timanttisia vastauksia, esim. elämäntarkoitus kysymykseen. Mallit osasivat myös mielestäni kiertää suhteellisen hyvin sellaiset kysymykset, mitkä eivät liittyneet millään tapaa niille syötettyyn kontekstiin. Chatbot päätepiirteessä Finetuned GPT-Neox 20B vaikutti myös varteenotettavimmalle vaihtoehdolle. Huono puoli tässä päätepiirteessä on, että kontekstia ei saa erikseen määrittelyä, mutta senkin saa kuitenkin toimimaan suhteellisen hyvin, jos kontekstin lähettää aiemman keskusteluhistorian yhteydessä. Tällöin se vie kyllä toki tilaa muulta keskusteluhistorialta. Tässäkin malli koitti myös soveltaa, jos se ei tiennyt vastausta, mikä oli sinänsä hetkittäin ihan hauskaakin, mutta se ei sovi välttämättä kovin hyvin kaikkiin tilanteisiin. Generation päätepiirteessä malli toimi mielestäni aika samalla tavalla, kuin Question päätepiirteessä. Hyötynä toki oli, että malli vastasi vähemmän luovasti semmoisiin kysymyksiin, mihin ei ollut vastausta kontekstissa. Eli jos halutaan välttää sellaisia tilanteita, missä malli vastaisi ihan omiaan, niin tämän avulla pystyy määrittämään aika hyvin, että mitä mallin kannattaa vastata, jos vastausta ei ole kontekstissa.

5 PÄÄTÄNTÖ

Tekoäly ja luonnollisen kielen käsittely ovat molemmat laajoja osa-alueita, ja varsinkin tässä opinnäytetyössä enemmän esitelty luonnollisen kielen käsittely on todella haastava tehtävä koneille. Ne ovat kuitenkin todella tehokkaita työkaluja, ja niiden avulla voidaan saavuttaa erittäin hyviä tuloksia moniin erilaisiin tehtäviin. Opinnäytetyön aikana käsitykseni luonnollisen kielen käsittelyn haasteellisuudesta vahvistui, mutta NLP Cloud API:n käyttö tuntui helpottavan huomattavasti siihen liittyviä haasteita. API:sta oli myös tarjolla selkeä dokumentaatio.

Mallit suoriutuivat mielestäni tehtävistä suhteellisen hyvin näinkin vähällä testaamisella ja esimerkkien tarjoamisella, mutta niiden käyttöönotto vaatii vielä mielestäni lisätestausta ja kehittämistä. Huono puoli esimerkkien (engl. few-shot learning) kanssa lähetettävissä API-kutsuissa on, että esimerkit pitää lähettää API-kutsun mukana joka kerta, koska ne eivät tallennu API:n puolelle.

Tämä työstää käytettävää tokeni määrää oikealta tekstiltä, koska nämä esimerkit lasketaan tokeni määrään mukaan, niin kuin myös mallin tekemä palautus. Tämä epäkäytännöllisyys toki korjautuisi sillä, kun malleja mahdollisesti hienosäädettäisiin jatkossa omalla datalla, johon voisi sisällyttää oikeanlaisia esimerkkejä. Tällöin jatkokehitetyn mallin kanssa, ei tarvitsi lähettää API-kutsujen mukana esimerkkejä, jolloin varsinaiselle tekstille jäisi enemmän tokeni tilaa. Erinäisten parametrien säätämällä ja mallin hienosäädöllä voisi myös mahdollisesti kohentaa mallien palauttamia tuloksia. On myös hyvä mainita, että laajamittaista mallien testausta ei pystytty suorittamaan tämän opinnäytetyön toteutuksen puitteissa, joten suosittelen kyllä niiden tekemistä jatkossa.

Kokonaisuudessaan Finetuned GPT-Neox 20B malli tuntui suoriutuvan monissa luonnollisen kielen käsittelyyn liittyvissä tehtävissä hyvin, joten suosittelisin jatkamaan kehitystyötä ainakin tämän mallin parissa. Asiakasprojektin tunneanalyysi oli siinä mielessä osittain haasteellinen, että asioista oli keskustellessa asiantuntijapaneeli. Teksti on siis voitu kirjoittaa suhteellisen neutraalisti, ja tunnetta voi olla joskus vaikea poimia siitä. Asiantuntijoiden kommentit eroavat mielestäni siis aika paljon esimerkiksi yrityksen sosiaalisen median tai tuotteiden arvosteluista, joiden käsittely olisi todennäköisesti helpompaa malleille.

Chattibottiin kannattaisi mielestäni olla ehkä myös valittavissa yleiset aihealueet, kun se avataan käyttöliittymässä. Tämä voisi mahdollisesti ratkaista sen haasteen, kun jokaisen pyynnön tokeni määrä on rajoitettu, ja kontekstia pystyy lähettämään vain rajatun määrän. Kovin pitkiä ja kaiken kattavia konteksteja ei siis pysty malleille lähettämään, joten aihealueiden valinnalla voitaisiin ottaa kysymyksiin vastaamiseen aina aihealueeseen sopiva konteksti.

Olisi myös hyvä, jos asiakasprojektin fasilitaattorit pystyisivät jatkossa merkitsemään jotenkin API:n mallien virheellisesti arvioimia tuloksia, ja tallentamaan oikean tai sopivamman arvon. Esimerkiksi vaikka tunneanalyysissä, jos malli on arvioinut tekstin tunnetilan väärin, tai tekstin yhteenvedossa, jos yhteenvedo ei ole fasilitaattorin mielestä onnistunut. Kun nämä virhepalautukset tallentettaisiin, niin niitä voitaisiin mahdollisesti hyödyntää myöhemmin mallin hienosäädössä.

LÄHTEET

Agrawal, R. 2021. Must Known Techniques for text preprocessing in NLP. Analytics Vidhya. Blogi. Päivitetty 14.6.2021. Saatavissa: <https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp/> [viitattu 15.4.2022].

Akerkar, R. 2019. Artificial Intelligence for Business. SpringerBriefs in Business. Cham: Springer.

AltexSoft. 2021. Natural Language Processing: A Guide to NLP Use Cases, Approaches, and Tools. Blogi. Päivitetty 25.8.2021. Saatavissa: <https://www.altexsoft.com/blog/natural-language-processing/> [viitattu 3.4.2022].

Beri, A. 2020. Stemming vs Lemmatization. Towards Data Science. Blogi. Päivitetty 14.5.2020. Saatavissa: <https://towardsdatascience.com/stemming-vs-lemmatization-2daddabcb221> [viitattu 14.4.2022].

Calaméo. 2021. Quick guide to Calaméo's API. Blogi. Päivitetty 25.3.2021. Saatavissa: <https://blog.calameo.com/2744/api-quick-guide/> [viitattu 22.4.2022].

Chowdhury, M. 2021. 10 Major Challenges of Using Natural Language Processing. Analytics Insight. WWW-dokumentti. Päivitetty 8.10.2021. Saatavissa: <https://www.analyticsinsight.net/10-major-challenges-of-using-natural-language-processing/> [viitattu 15.4.2022].

Crunchbase s.a. NLP Cloud. WWW-dokumentti. Saatavissa: <https://www.crunchbase.com/organization/nlp-cloud> [viitattu 5.4.2022].

Diaz, J. 2017. The 3 Different Generations of Chatbot Technology. Blogi. Päivitetty 12.12.2017. Saatavissa: <https://rul.ai/blog/3-different-generations-chatbot-technology/> [viitattu 16.4.2022].

Eiben, A.E. & Smith, J.E. 2015. Evolutionary Computing: The Origins. Teoksessa Eiben, A.E. & Smith, J.E. (toim.) Introduction to Evolutionary Computing. Natural Computing Series. 2. painos. Berlin: Springer, 13–20. PDF-dokumentti. Saatavissa: https://warin.ca/ressources/books/2015_Book_IntroductionToEvolutionaryComp.pdf [viitattu 18.4.2022].

Endpoints s.a. NLP Cloud. WWW-dokumentti. Saatavissa: <https://docs.nlpcloud.io/#endpoints> [viitattu 6.4.2022].

Fedak, V. 2018. 5 Heroic Tools for Natural Language Processing. Towards Data Science. Blogi. Päivitetty 30.1.2018. Saatavissa: <https://towardsdatascience.com/5-heroic-tools-for-natural-language-processing-7f3c1f8fc9f0> [viitattu 3.4.2022].

Géron, A. 2019. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2. painos. Sebastopol: O'Reilly Media, Inc.

Google Cloud s.a. Natural Language AI. WWW-dokumentti. Saatavissa: <https://cloud.google.com/natural-language> [viitattu 17.4.2022].

GSMA Future Networks. 2019. AI & Automation: An Overview. WWW-dokumentti. Päivitetty 25.6.2019. Saatavissa: <https://www.gsma.com/futurenetworks/wiki/ai-automation-an-overview/> [viitattu 22.4.2022].

Hyvärinen, A-M. 2018. Alimerkkijonot suomen sanojen vektoriesitysten tuottamisessa neuroverkoilla. Helsingin yliopisto. Matemaattis-luonnontieteellinen tiedekunta. Pro gradu -tutkielma. PDF-dokumentti. Saatavissa: <http://urn.fi/URN:NBN:fi-fe201902286685> [viitattu 22.4.2022].

IBM. 2021. How to build apps with natural language processing. WWW-dokumentti. Päivitetty 29.9.2021. Saatavissa: <https://www.ibm.com/watson/natural-language-processing> [viitattu 16.4.2022].

IBM Cloud Education. 2021. Natural Language Processing (NLP). WWW-dokumentti. Päivitetty 16.8.2021. Saatavissa: <https://www.ibm.com/cloud/learn/natural-language-processing> [viitattu 29.3.2022].

IBM Technology. 2021. What is NLP (Natural Language Processing)? YouTube. Videoleike. Päivitetty 12.8.2021. Saatavissa: <https://www.youtube.com/watch?v=fLvJ8VdHLA0> [viitattu 12.4.2022].

Introduction s.a. NLP Cloud. WWW-dokumentti. Saatavissa: <https://docs.nlpcloud.io/#introduction> [viitattu 6.4.2022].

JavaTpoint s.a. What is an Expert System? WWW-dokumentti. Saatavissa: <https://www.javatpoint.com/expert-systems-in-artificial-intelligence> [viitattu 18.4.2022].

Jayathilaka, M. 2020. 25 NLP tasks at a glance. Medium. Blogi. Päivitetty 25.6.2020. Saatavissa: <https://medium.com/@miranthaj/25-nlp-tasks-at-a-glance-52e3fdff32e2> [viitattu 29.3.2022].

Jha, A. 2021. Vectorization Techniques in NLP [Guide]. Neptune. Blogi. Päivitetty 31.12.2021. Saatavissa: <https://neptune.ai/blog/vectorization-techniques-in-nlp-guide> [viitattu 22.4.2022].

JivoChat s.a. Chatbot is a useful assistant for your business. WWW-dokumentti. Saatavissa: <https://www.jivochat.com/chatbots/> [viitattu 22.4.2022].

Jung, A. 2022. Machine Learning: The Basics. Singapore: Springer.

Krupsky, S. 2021. Natural Language Processing Introduction: what is Natural Language Processing (NLP)? NLP Cloud. Blogi. Päivitetty 5.7.2021. Saatavissa: <https://nlpcloud.io/introduction-what-is-nlp-natural-language-processing.html> [viitattu 14.4.2022].

Language Technology at the University of Helsinki. 2022. GitHub. WWW-dokumentti. Päivitetty 6.4.2022. Saatavissa: <https://github.com/Helsinki-NLP> [viitattu 6.4.2022].

Lantz, B. 2019. Machine Learning with R: Expert techniques for predictive modeling. 3. painos. Expert Insight. Birmingham: Packt Publishing Ltd.

Lawton, G. 2021. NLP Cloud helps app developers add language processing. VentureBeat. WWW-dokumentti. Päivitetty 12.4.2021. Saatavissa: <https://venturebeat.com/2021/04/12/nlpcloud-io-helps-app-developers-add-language-processing/> [viitattu 5.4.2022].

Mayo, M. 2018. The Main Approaches to Natural Language Processing Tasks. KDnuggets. Blogi. Päivitetty 17.10.2018. Saatavissa: <https://www.kdnuggets.com/2018/10/main-approaches-natural-language-processing-tasks.html> [viitattu 17.4.2022].

Menzli, A. 2021. Tokenization in NLP: Types, Challenges, Examples, Tools. Neptune. Blogi. Päivitetty 13.12.2021. Saatavissa: <https://neptune.ai/blog/tokenization-in-nlp> [viitattu 12.4.2022].

Microsoft. 2021. Transparency note and use cases for question answering. WWW-dokumentti. Päivitetty 5.11.2021. Saatavissa: <https://docs.microsoft.com/en-us/legal/cognitive-services/language-service/transparency-note-question-answering> [viitattu 16.4.2022].

Models List s.a. NLP Cloud. WWW-dokumentti. Saatavissa: <https://docs.nlpcloud.io/#models-list> [viitattu 6.4.2022].

MonkeyLearn s.a. Natural Language Processing (NLP): What Is It & How Does it Work? WWW-dokumentti. Saatavissa: <https://monkeylearn.com/natural-language-processing/> [viitattu 29.3.2022].

Murali, A. 2021. A Guide to Perform 5 Important Steps of NLP Using Python. Analytics Vidhya. Blogi. Päivitetty 17.8.2021. Saatavissa: <https://www.analyticsvidhya.com/blog/2021/08/a-guide-to-perform-5-important-steps-of-nlp-using-python/> [viitattu 22.4.2022].

Nityasya, M.N., Wibowo, H.A., Prasojo, R.E. & Aji, A.F. 2021. Costs to Consider in Adopting NLP for Your Business. 2. versio. ArXiv. PDF-dokumentti. Päivitetty 15.4.2021. Saatavissa: <https://arxiv.org/pdf/2012.08958.pdf> [viitattu 15.4.2022].

NLP Cloud s.a. Advanced Artificial Intelligence API. WWW-dokumentti. Saatavissa: <https://nlpcloud.io/> [viitattu 17.4.2022].

OpenAI. 2021. Build next-gen apps with OpenAI's powerful models. WWW-dokumentti. Päivitetty 12.11.2021. Saatavissa: <https://openai.com/api/> [viitattu 17.4.2022].

Pricing s.a. NLP Cloud. WWW-dokumentti. Saatavissa: <https://nlpcloud.io/#plans> [viitattu 6.4.2022].

Question Answering API s.a. NLP Cloud. WWW-dokumentti. Saatavissa: <https://nlpcloud.io/nlp-question-answering-api.html> [viitattu 16.4.2022].

RazorSocial. 2020. Artificial Intelligence: A Modern Approach to Marketing. Blogi. Päivitetty 4.11.2020. Saatavissa: <https://www.razorsocial.com/artificial-intelligence-a-modern-approach-to-marketing/> [viitattu 22.4.2022].

SAS Institute. 2021. Artificial Intelligence: What it is and why it matters. WWW-dokumentti. Päivitetty 16.7.2021. Saatavissa: https://www.sas.com/en_in/insights/analytics/what-is-artificial-intelligence.html [viitattu 22.4.2022].

Sentiment and Emotion Analysis API s.a. NLP Cloud. WWW-dokumentti. Saatavissa: <https://nlpcloud.io/nlp-sentiment-analysis-api.html> [viitattu 16.4.2022].

Simplilearn. 2021. Natural Language Processing In 5 Minutes | What Is NLP And How Does It Work? | Simplilearn. YouTube. Videoleike. Päivitetty 17.3.2021. Saatavissa: <https://www.youtube.com/watch?v=CMrHM8a3hqw> [viitattu 12.4.2020].

Singhal, S. 2021. Data Representation in NLP. Medium. Blogi. Päivitetty 10.12.2021. Saatavissa: <https://shiiivangii.medium.com/data-representation-in-nlp-7bb6a771599a> [viitattu 22.4.2022].

Summarization API s.a. NLP Cloud. WWW-dokumentti. Saatavissa: <https://nlpcloud.io/nlp-question-answering-api.html> [viitattu 16.4.2022].

Thaker, M. 2019. Comparing Text Summarization Techniques. Medium. Blogi. Päivitetty 25.3.2019. Saatavissa: <https://medium.com/@thakermadhav/comparing-text-summarization-techniques-d1e2e465584e> [viitattu 22.4.2022].

Turner, E. 2022. How to Use Massive AI Models (Like GPT-3) in Your Startup. Future. WWW-dokumentti. Päivitetty 24.2.2022. Saatavissa: <https://future.a16z.com/ai-ml-foundation-models-for-the-rest-of-us/> [viitattu 22.4.2022].

VijiPriya, J., Ashok, J. & Suppiah, S. 2016. A Review on Significance of Sub Fields in Artificial Intelligence. *International Journal of Latest Trends in Engineering and Technology* 6.3, 542–548. PDF-dokumentti. Saatavissa: <https://www.ijltet.org/journal/86.pdf> [viitattu 18.4.2022].

Voutilainen, A., Purtonen, T. & Muhonen, K. 2012. FinnTreeBank2 Manual. Helsingin yliopisto. Nykykielten laitos. PDF-dokumentti. Saatavissa: <http://www.ling.helsinki.fi/kieliteknologia/tutkimus/treebank/sources/FinnTree-BankManual.pdf> [viitattu 16.4.2022].

What is a machine learning model? 2021. Microsoft. WWW-dokumentti. Päivitetty 30.12.2021. Saatavissa: <https://docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model> [viitattu 18.4.2022].

What is API: Definition, Types, Specifications, Documentation. 2021. Al-texSoft. Blogi. Päivitetty 28.7.2021. Saatavissa: <https://www.al-texsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/> [viitattu 19.4.2022].

Wolff, R. 2020. Top 10 NLP Tools & Services in 2022. MonkeyLearn. Blogi. Päivitetty 11.3.2020. Saatavissa: <https://monkeylearn.com/blog/natural-language-processing-tools/> [viitattu 3.4.2022].

Wyatt, M. 2022. What is an API? A Digestible Definition with API Examples for Ecommerce Owners. BigCommerce. Blogi. Päivitetty 8.2.2022. Saatavissa: <https://www.bigcommerce.com/blog/what-is-an-api/#so-what-is-json-and-why-is-it-used> [viitattu 19.4.2022].

Yao, M., Jia, M. & Zhou A. 2018. Applied Artificial Intelligence: A Handbook for Business Leaders. New York: TOPBOTS Inc.

Zaremba, W. & Brockman, G. 2021. OpenAI Codex. OpenAI. Blogi. Päivitetty 18.11.2021. Saatavissa: <https://openai.com/blog/openai-codex/> [viitattu 19.4.2022].