



SAVONIA

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

PEPPER ROBOTIN KOMMU- NIKOINTI CHATBOTTIEN KANSSA

TEKIJÄ/T: Petrus Mertanen

Koulutusala Tekniikan ja liikenteen ala			
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan koulutusohjelma			
Työn tekijä(t) Petrus Mertanen			
Työn nimi Pepper-robotin kommunikointi Chatbottien kanssa			
Päiväys	Joulukuu 14, 2020	Sivumäärä/Liitteet	25
Ohjaaja(t)			
Toimeksiantaja/Yhteistyökumppani(t) Savonia-ammattikorkeakoulu			
Tiivistelmä			
<p>Opinnäytetyön tavoitteena oli selvittää pystyykö Savonian Pepper-robotin yhdistämään chatbottiin ilman päivitystä. Mikäli robotin yhdistäminen chatbottiin on mahdollista, parantaisiko se Pepperissä havaittuja kuulo-ongelmia, auttaisiko se myös kehittämään robotin keskustelun laatua ja saako Pepperin ja chatbotin keskustelusta irti raakadataa. Työn toimeksiantajana toimii Savonia-ammattikorkeakoulu, jonka HyvinRobo hankkeessa oltiin huomattu, ettei Pepperille pystytä rakentamaan syvällisiä keskusteluja.</p> <p>Teoriaosuudessa selvitettiin mitä ovat robotit, tekoäly ja chatbotit. Tietoa selvitettäviin asioihin löytyi kirjoista, internetistä löytyvistä artikkeleista ja chatbotteihin liittyvien yritysten sivuilta. Työhön tarvittavia ohjeita löytyi myös sovellusten kehitysyhteistyöalustalta GitHubista.</p> <p>Työosuudessa keskusteltiin muutaman suomenkielisten chatbottien kanssa, joiden avulla saatiin selville miten keskustelut niiden kanssa toimii. Pepperille rakennettiin keskustelu sen omassa Choregraphe-ohjelmointiympäristössä vertailuksi, miten erilainen se on chatbotteihin verrattuna. Robottia yritettiin yhdistää IBM-Watsonin chatbottiin ja Googlen pilvipohjaisen projektin Dialogflowhon rakennettuun chatbottiin.</p> <p>Työssä saatiin tehtyä muutama chatbot- ja dialogi-ohjelma. Chatbot-ohjelmissa ilmeni ongelmia, joita ei saatu korjattua. Dialogi-ohjelmat toimivat hyvin Pepperin kanssa ja Pepperiltä voi etsiä tietoa liittyen kellonaikaan, päivämäärään ja nimipäivään.</p>			
Avainsanat chatbot, humanoidirobotti, Pepper, Python, tekoäly			

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Petrus Mertanen			
Title of Thesis Pepper Robot's Communication with Chatbots			
Date	December 14, 2020	Pages/Appendices	25
Supervisor(s)			
Client Organisation /Partners Savonia University of Applied Sciences			
<p>Abstract</p> <p>The purpose of this thesis was to research whether it is possible to connect Savonia's Pepper robot with chatbots. If it is possible, could it help the robot with its hearing problems, could it also improve its conversation skills and is it possible to import and analyze the raw data from the robot's and the chatbot's conversations. The subjects and the goals came from Savonia UAS which had a project called HyvinRobo. The problems with Pepper were first noticed in this project.</p> <p>First in the theory part, robots, artificial intelligence and chatbots were researched. Information for the research was gathered from multiple sources such as books, articles from the Internet and business oriented websites about chatbots. Some instructions were also found from the GitHub. Secondly in the development part, there were first chats with a couple of Finnish chatbots which helped to understand how the conversation with them works. A dialog program was made for Pepper to compare how different it is from chatbots. Pepper was tried to be connected with the IBM and Watson based chatbot and the Google cloud project's Dialogflow based chatbot.</p> <p>As the results of this thesis, a couple of chatbots and a couple of dialog programs were made for Pepper. The chatbot programs have some issues which could not been solved. The dialog programs work fine with Pepper and a user can ask information about the current date, time and the name day of the date from the robot.</p>			
<p>Keywords artificial intelligence, chatbot, hmanoid robot, Pepper, Python</p>			

SISÄLTÖ

1	JOHDANTO.....	5
1.1	Lyhenteet ja määritelmät.....	5
2	TAUSTAA	6
3	ROBOTIT	7
3.1	Humanoidirobotit	7
3.2	Pepper	7
4	TEKOÄLY.....	8
4.1	Tekoälyn tasot.....	8
5	CHATBOTIT.....	9
5.1	Ongelmat Chatbotin kanssa	9
6	KESKUSTELUT CHATBOTTIEN KANSSA.....	10
6.1	Sulo chatbot.....	10
6.2	Porvoon Kunta-Kati	11
6.3	Kuopion kaupunginkirjaston chat	12
7	KESKUSTELUN RAKENTAMINEN PEPPERILLE	13
7.1	Keskustelun luominen Choregraphessa.....	13
7.2	Ongelmia Pepperin keskustelun luomisessa	14
8	PEPPERIN YHDISTÄMINEN CHATBOTTIIN	15
8.1	Pepperin yhdistäminen IBM ja Watson chatbottiin.....	15
8.1.1	Ongelmat IBM ja Watson chatbot ohjelman teossa.....	17
8.2	Pepperin yhdistäminen Googlen Dialogflowhon	18
8.3	Ongelmia Dialogflow sovelluksen kanssa	19
8.4	Pepper robotista chatbot	20
9	POHDINTA	22
10	JATKO.....	23
	LÄHTEET JA TUOTETUT AINEISTOT	24

1 JOHDANTO

1.1 Lyhenteet ja määritelmät

Python Python on ohjelmointikieli, joka eroaa muista ohjelmointikielistä siten, ettei sillä annetut komennot pääty puolipisteeseen kuten yleensä ohjelmointikielissä.

Robotti Robotti on kohdeyleisöä varten rakennettu ja ohjelmitava tekninen tuote. Se voi olla teollinen työntekijä kuten teollisuushallien robottikädet tai robotti-imuri. Robotti voi olla myös viihdyke kuten lapsille suunnatut eläinaiheiset robotit. Tai se voi olla myös viihdyttäjä ja läsnä kuten humanoidirobotit.

Humanoidi Humanoidi on ihmisiä piirteitä tai ihmistä muistuttava olento.

URL URL tulee sanoista Uniform Resource Locator. Ja sillä yleensä osoitetaan verkkosivun osoite.

Choregraphe Choregraphe on SoftBank Roboticsin robottien ohjelmointiympäristö.

Fif.top-tiedosto Fif.top-tiedosto on Choregraphen ympäristössä käytettävä keskustelutiedostotyyppi, johon kirjoitetaan robotin kanssa käytävä keskustelu valitulla kielellä.

GCP GCP tulee sanoista Google cloud project. Sivusto on Googlen pilvipohjainen projektisivu, jonne saa luoda 12 ilmaista projektia ja niitä varten voi ladata 300 krediitin edestä tavaraa.

Intent Intenttejä käyttävät sekä IBM-Watsonin sivusto kuin myös Googlen Dialogflowkin. Intent kategorisoi käyttäjän aiheet.

String String on ohjelmointikielissä käytetty merkkijonosta koostuva muuttuja.

2 TAUSTAA

Savonian Pepper-robotti, joka tunnetaan myös Tuiskuna, on ollut mukana HyvinRobo-hankkeessa. Hankkeessa tutkittiin miten hyvinvointiteknologia ja robotiikka sopisi hoitotyön tueksi ja toisi esille teknologian mahdollisuuksia ja tietoisuutta niistä hoitotyöntekijöille. Pepper pilotoi eri hoitotyöyksiköissä. Pilotointeihin on toivottu usein keskustelumahdollisuutta. Muutamiiin paikkoihin on tehty Pepperille yksinkertaisia keskusteluja, joissa robotti reagoi kaikkeen minkä kuulee. Keskustelun tulisi olla kuitenkin syvällisempää. Joten yritän opinnäytetyölläni tutkia, auttaako mahdollisesti Pepperin yhdistäminen chatbottiin, kuulemaan sitä paremmin.

3 ROBOTIT

Robotti tulee tsekin orjatyöläistä tarkoittavasta sanasta robota ja sitä käytettiin ensimmäisen kerran 1900-luvun alussa. Robottien historia ulottuu keskiajalle, jolloin automatisoidut humanoidirobotit automatat, tekivät ajoitettua työtä. Sveitsiläisessä Neuchâtelin museossa on nähtävänä ohjelmoitava kirjoittava automata the Scribe. Niitä voidaan pitää teollisuusrobottien esikuvana, vaikka ne eivät aikanaan herättäneet kiinnostusta. (Poole, 2012)

Nykyään teollisuusrobotit kokoavat yhteen kokonaisuuksia, joita ihminen viimeistelee. Ihmisen on seurattava robotin työtä ja tarpeen vaatiessa huollettava niitä. Kasvanut teollisuusrobottien määrä, ei ole vienyt ihmisiltä töitä, vaan se on voinut muuttaa työnkuvaa ja tuoda uusia työmahdollisuuksia. (Robotit hoitavat tylsät työt, 2020) Vielä tänäpäivänä tietoisesti ajattelevat robotit ei ole todellisia, mutta hyvin ohjelmituna tekoälyllä varustettua yksikköä voi pitää tietoisena. Neuroverkot kasvattavat robottien tietoisuutta. Vaikka robottien tietoisuus on kiinnostava tutkimuskohde, sitä myös vastustetaan. Vastaväitteinä niitä kohtaan pidetään sitä, että ihmiset olisivat myös koneita, jos robotitkin saavat tietoisuuden. Tietoisien robotin toiminnot olisivat kuin ihmisen tekemiä valintoja ja tekoja. Jotta robotit saavuttaisivat tietoisuuden, olisi niiden osattava yhdistää näkemiään ja kokemiaan kokonaisuuksia. Tietoisien robotin olisi kyettävä myös ymmärtämään tunnetiloja. (Haikonen, 2017)

3.1 Humanoidirobotit

Humanoidirobotti on ihmisen kaltainen robotti, joka pystyy tekemään automaattisia tehtäviä. Ne ovat uusi ilmiö palvelurobotiikan saralla. Humanoidirobotteja käytetään vanhus- ja palvelutyössä. (Association) Humanoidirobotit eivät aina muistuta täysin ihmistä, vaan voivat imitoida jotain ihmisen osaa. Sensoreiden ja moottoreiden avulla ne osaavat toimia ja liikkua erilaisissa paikoissa. Humanoidirobottien työt riippuvat siitä, mikä on niille määritetty tarkoitus. (Dang, 2019)

Robottien inhimilliset piirteet voivat aiheuttaa ihmisissä pelkoa ja inhoa. Samanlaista inhoa ja pelkoa aiheuttaa elokuvissa tietokoneella animoidut ihmiset kuten Cats -elokuvan ihmiskissat. Pitämällä esillä robotin koneellisia piirteitä, siitä voi saada positiivisemmän käsityksen. (Mankkinen, 2020)

3.2 Pepper

Pepper on SoftBank Roboticsin valmistama humanoidirobotti. Pepperissä on useita sensoreita ja toimintoja kuten puheentunnistus ja tabletti rintakehällä, mitkä auttavat sitä kohtaamaan ihmisiä ja hahmottamaan ympäristöä. (Robotics) Pepperin on tarkoitus olla lähestyttävä, jolta lapsikin saisi tarvitsemaansa apua. Robotti osaa tehdä ihmiskontaktia vaativia tehtäviä muttei kykene vielä toimimaan yksin. Tekoälyn rajoitukset tulevat vastaan Pepperin kanssa keskustellessa. Tiedon robotti ottaa keskusteluista kysymyksien perusteella. Pepperin kanssa keskustelun tarjoava digital flow-moottori, mahdollistaa eri tilanteisiin sopivan keskustelun ohjelmoimisen. (PYMNTS, 2018)

4 TEKOÄLY

Tekoälyä alettiin kehittää 1950-luvulla, jolloin se määriteltiin, että sillä on oltava älylle vaadittavat ominaisuudet ja fysikaalinen tuote. Fysikaalisella tuotteella tarkoitetaan digitaalista tietokonetta. Määritelmä pätee yhä nykypäivänä. Tuolloin tietokoneiden älykkyyttä ei voitu todistaa, tietokoneet olivat hitaita ja niissä oli vähän muistia. Nykypäivänä tietokoneiden muisti ja nopeus on kasvanut. (Haikonen, 2017) Nykyään tekoälyä on kiinni ohjelmistoissa ja ohjelmissa, mutta vielä se ei kykene toimimaan ihmisaivojen tavoin. Oppiakseen uutta, tekoäly tarvitsee ihmisen antaman ärsyksen. Uuden oppiminen voi tapahtua syväoppimisen avulla. Syväoppimisessa tekoäly oppii hahmottamisen ja matkimisen kautta. (Timo Siukonen, 2019)

4.1 Tekoälyn tasot

Tekoälyä on kahdenlaista. Heikkoa tekoälyä, joka pohjautuu koneoppimiseen ja algoritmeihin. Toinen on vahva tekoäly, jossa tietokoneilla olisi ihmisen kaltaiset toiminnot ja tietoisuus. Saavuttaakseen vahvan tekoälyn, tulisi tietokoneen saavuttaa ihmismieli algoritmein. Algoritmit määrittelevät tietokoneohjelman ongelmanratkaisun sisällön, säännöt ja toiminnot. Toimiakseen tarkasti tuntemattomat tiedon kanssa, neuroverkot käsittelevät esimerkkitietoa. Tekoälyn on vaikea saavuttaa ihmismielen ainutlaatuisuutta ja luovaa tietoisuutta. Tiedon ja taidon suhteen tekoäly on yhä puutteellinen. Se ei pärjäisi ekaluokkalaisten kyvyille. (Timo Siukonen, 2019) Tekoälyä voidaan käyttää moderoina keskustelufoorumeilla. Kuitenkin tekoälyn antamia kieltoja voi kiertää. Ihmistä tarvitaan yhä tarkistamaan keskusteluja, koska kone ei vielä osaa erottaa keskusteluista symbolisia piirteitä. (Seuri, 2020)

5 CHATBOTIT

Chatbot on keskusteluohjelmissa toimiva botti, joka keskustelee asiakkaiden kanssa. Salesforce -siivuston mukaan chatbotteja on kolmenlaisia: sääntöihin perustuvia keskustelijoita, kysymyksiin valmiilla vastauksilla reagoivia ja molempien yhdistelmiä. Chatbottien toiminnoissa voi olla mukana myös tekoälyä, joka auttaa bottia ymmärtämään, miten sen pitäisi reagoida asiakkaan antamaan ärsykkeeseen. Niitä käytetään erilaisissa yrityksissä eri tilanteissa, esimerkiksi neuvoa toimimaan oikein. Chatboteissa on kuitenkin omat ongelmansa. Ne eivät aina ymmärrä kaikkea ja niiden toiminnot ovat rajallisia. Ne voivat oppia keskusteluista kerätyn datan avulla. (Hupli, 2018) Ohjeita seuraavat botit eivät opi uutta ja keskustelu ei etene tuntemattomien asioiden kohdalla. Chatbottia rakentaessa on mietittävä tarkasti käyttäjän ja botin roolit. (ChatbotPack, 2019) Suomenkielisiä chatbotteja on tällä hetkellä vähän. Sellaisen voisi luoda kärsivällisyyden kanssa. Tilanteesta riippuen chatbotin pitää pystyä reagoimaan käyttäjän esittämiin kysymyksiin. Botti voi aloittaa keskustelun asiakkaan kanssa, mutta se ei saa olla aggressiivinen. (Laukkonen, 2018)

5.1 Ongelmat Chatbotin kanssa

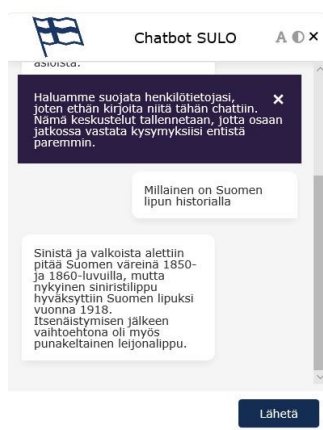
Vuoden 2019 syyskuussa vakuutusyhtiön Turvan chatbot Teppo aiheutti niin kysymyksen esittäjälleen kuin Turvan henkilöstölle naurunhetkiä. Vastasyntynyt äiti esitti Tepolle kysymyksen lapsen syntymästä, johon Teppo vastasi pahoittelemalla tapahtunutta vahinkoa. Chatbotin vastaus kysymykseen lapsen syntymästä, liittyi kysymyksen alkuun, joka on tyypillinen aloitus usealle onnettomuustilanteelle tai kadonneista tavaroista kysyttäessä. (Koskinen, 2019) Chatbottien kanssa ongelmia on saavutettavuus, ihmiset ei välttämättä löydä niiden pariin, kun chatbottien mainostamiseen ei panosteta. Ne eivät tuo kehittäjilleen voittoa. Chatbottien kanssa on hankala keskustella sujuvasti. (BotSide, 2017)

6 KESKUSTELUT CHATBOTTIEN KANSSA

Jotta chatbottien toiminnasta ja ymmärryksestä voisi saada selkeän kuvan, on keskusteltava sellaisten kanssa. Internetistä löytyy suomenkielisiä chatbotteja ainakin Sulo chatbot ja Porvoon Kunta-Kati. Aina ei kuitenkaan voi olla täysin varma onko keskustelija ihminen vai botti. Kuopion kaupunginkirjaston chatissa keskustelijana toimii ihminen. Kirjaston kanssa keskustellessa kieli oli inhimillisempää ja toisin kuin Kunta-Katin ja Sulon kanssa, se teki myös kirjoitusvirheen keskustelun aikana.

6.1 Sulo chatbot

Ensin juteltiin Sulo -chatbotin kanssa Suomen lipusta ja vaakunasta. Ensimmäiseksi Sulolta kysyttiin Suomen lipun historiasta, johon saatiin vastaus (KUVA 1.). Sulo vastasi kertomalla Suomen lipun historiasta.



KUVA 1. Suomen lipun historia (Mertanen, 2020)

Seuraavaksi kysyttiin mitä tehdä jos lippu koskettaa maata, sitä alas tangosta laskiessa. Sulo vastasi kysymykseen aiheellisesti ja oikein (KUVA 2.).



KUVA 2. Lipun lasku (Mertanen, 2020)

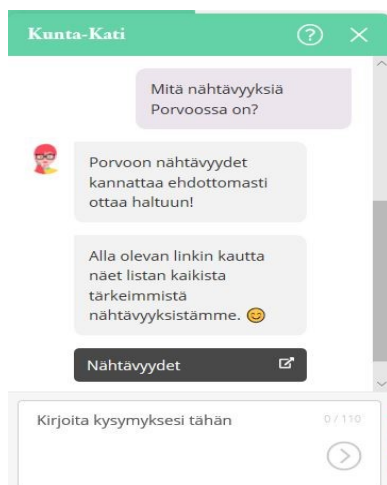
Sulo ei kuitenkaan kertonut Kuopion, vaakunasta kun sitä kysyttiin siltä. Sen sijaan Sulo kertoi Suomen vaakunasta, millainen se on (KUVA 3.). Virhe taisi kuitenkin johtua kysyjästä. Sulo on tehty vastaamaan ainoastaan Suomen lippuun ja vaakunaan koskeviin kysymyksiin. Sulon oletettiin vastaavan kaikkia Suomen vaakunoita koskeviin kysymyksiin, myös eri kuntien vaakunoihin.



KUVA 3. Vastaus ohi kontekstin (Mertanen, 2020)

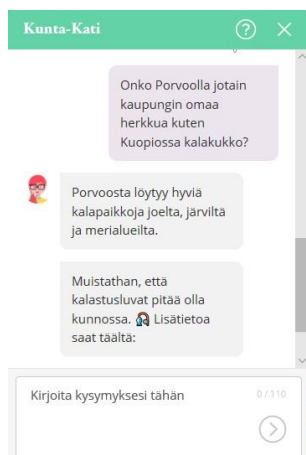
6.2 Porvoon Kunta-Kati

Sulon jälkeen juteltiin Porvoon Kunta-Katin kanssa. Chatbotilta pystyi kysymään Porvoon kaupunkiin ja siihen liittyviä asioita. Aluksi kysyttiin mitä nähtävyyksiä Porvoossa on. Siihen Kati vastasi antamalla linkin Porvoon nähtävyyksiin (KUVA 4.).



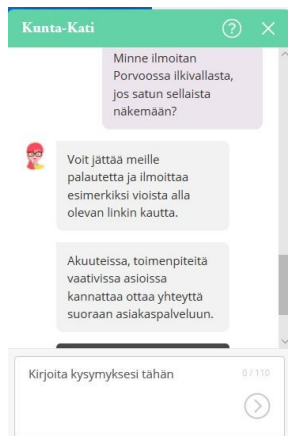
KUVA 4. Porvoon nähtävyydet (Mertanen, 2020)

Seuraavaan kysymykseen Kati ei osannut kunnolla vastata, kun kysyttiin onko Porvoossa Kuopion kalakukon kaltaista herkuä. Kysymykseen saatiin vastaus, mutta se ei kuitenkaan koskenut mitään perinneherkkuä. Saatua vastaus liittyi kalastukseen (KUVA 5.). Asiaa yritettiin useamman kerran kysyä eri tavalla, mutta Kati vastasi joko samalla tavalla kuin aiemmin tai kertoi, ettei ymmärtänyt esitettyä kysymystä.



KUVA 5. Porvoon herkut (Mertanen, 2020)

Katin kanssa keskustellessa huomattiin, miten se on ohjelmoitu vastaamaan tiettyjen sanojen yhteydessä. Kun Katilta kysyttiin miten toimia, jos huomataan ilkivaltaa Porvoossa (KUVA 6.). Kati vastasi siihen, että pitäisi lähettää palautelomake sivuston toiminnasta. Katin oletettiin antavan esimerkiksi ohjeita ottamaan yhteyttä poliisiin. Chatbot ilmeisesti tarttui kysymyksessä olleeseen ilmoittajaan ja vastasi sen myötä, miten se on ohjelmoitu vastaamaan vastaavassa tilanteessa.



KUVA 6. Kysymys ilkivallasta (Mertanen, 2020)

6.3 Kuopion kaupunginkirjaston chat

Kuopion kaupungin kirjaston kanssa keskusteltiin kirjasta. Kirja löytyi kirjastosta ja sen varaamiseen annettiin ohjeita. Chatin puhetyyli herätti epäilyksiä, onko puhujana ihminen vai kone. Keskustelija teki myös kirjoitusvirheen kesken tekstin. Kunta Kati ja Sulo eivät tehneet kirjoitusvirheitä, joten keskustelijalta kysyttiin onko chatissa vastaajana ihminen. Keskustelija paljastui ihmiseksi.

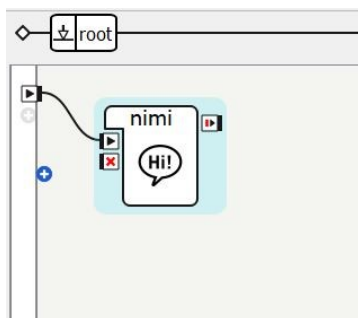
7 KESKUSTELUN RAKENTAMINEN PEPPERILLE

Pepperille rakentaessa keskustelua, siitä tehdään ohjeita seuraava chatbot. Ohjelmoija kirjoittaa keskustelun, jonka robotti käy ihmisen kanssa. Ohjelmoijan on osattava ennakoida vastaukset tai sanat, joihin Pepper reagoi. Keskustelun on parasta antaa Pepperin aloittaa, vaikkapa tervehtimällä edessään seisovaa ihmistä ja kysyä tämän nimeä.

Robotille tehdyistä keskusteluista on huomattu se, ettei sille voi tehdä syvällisiä keskusteluja, kun se ei kuule kunnolla. Sen takia niistä ollaan rakennettu yksinkertaisia. Näissä Pepper reagoi jokaiseen ärsykkeeseen, jonka se kuulee.

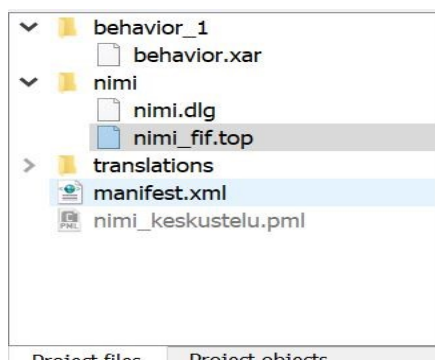
7.1 Keskustelun luominen Choregraphessa

Ohjelmointiympäristössä Choregraphessa luodaan aluksi dialogilaatikko, johon kirjoitetaan keskustelu (KUVA 7.). Choregraphessa on myös määriteltävä millä kielellä keskustelu tehdään. Ensin on



KUVA 7. Dialogilaatikko, jolle on annettu nimi, nimi (Mertanen, 2020)

avattava properties ikkuna, josta valitaan kielivalikosta tarvittava kieli ja tarvittaessa poistetaan englanti. Englannin kieli on automaattisesti valittuna, kun aloitetaan ohjelman rakentaminen Choregraphessa. Kun alkuvalmistelut on tehty, on muokattava keskustelun käytänteitä, eli luotava keskustelu Pepperin ja käyttäjän välille. Kun dialogilaatikko luodaan, se tekee käytänteet-tiedoston ohjelmaan, joka löytyy Choregraphen vasemman puoleisesta listasta fif.top -tiedostona (KUVA 8.).



KUVA 8. Fif.top -tiedosto (Mertanen, 2020)

Sitä kaksoisklikatessa, ilmestyy käytänteet muokattavaksi. Keskustelun luomisessa käyttäjä tekee ensin toiminnon, johon Pepper vastaa jotenkin. Alkutoiminnon ei tarvitse olla puhe, se voi olla myös jokin tekninen –toiminto, kuten kun dialogilaatikko aktivoituu, tapahtuu jotain.

```
u:(e:onStart) Hei, nimeni on Pepper. Savoniolla tunnetaan myös Tuiskuna. Kuka sinä olet?
```

KUVA 9. Aloitus (Mertanen, 2020)

Esimerkiksi dialogilaatikon aktivoituessa, Pepper tervehtii käyttäjää ja kysyy nimeä. (KUVA 9.) Kun Pepper on kysynyt nimeä, on ohjelmoijan mietittävä mahdolliset erilaiset tavat kertoa oma nimi. Pepper reagoi käyttäjän syötteeseen jotenkin, vaikkapa kehumalla nimeä ja kysyen mitä nimi tarkoittaa (KUVA 10.).

```
u1:({"olen" "nimeni on" "minä olen"} * ) Miten mukava nimi. Jutellaanko nimistä, tiedätkö mitä nimesi tarkoittaa?
```

KUVA 10. Nimi ja kehut (Mertanen, 2020)

Eri vastausvaihtoehdot kirjoitetaan samaan kohtaan eri riveillä, ja keskustelun jatkoa voidaan ilmentää keskustelun käytänteissä sisennyksillä (KUVA 11.). Pepper vastaa kyllä vastaukseen satunnaisesti valitulla ilmaisulla, joista jokainen on kirjattu vastausvaihtoehdoksi heittomerkkien sisään.

```
u2:(kyllä) Mitä nimesi tarkoittaa?
u3:(*) ^rand["Ai, sitä nimesi tarkoittaa" "Oho, en olisi tiennyt mitä nimesi tarkoittaa" "Ahaa, no selittää paljon" "Vai niin, onpas jännä alkuperä" "Aijaa, en olisi ikinä arvannut nimesi alkuperää"] Minun Pepper-nimeni tarkoittaa suomeksi pippuria. $onStopped=1
u2:(en) Oli mukava tutustua. $onStopped=1
```

KUVA 11. Sisennys (Mertanen, 2020)

7.2 Ongelmia Pepperin keskustelun luomisessa

Pepperin keskustelun luomisessa on myös ongelmansa. Ensimmäiseksi, kun antaa käytänteissä erilaisia vastausmahdollisuuksia keskusteluun, Pepper ei välttämättä jostain syystä kuuntele niitä. Vastausvaihtoehtoja voi olla liikaa, jolloin se ei osaa reagoida annettuun vastaukseen. Robotti ei myöskään kuule hyvin ja voi kuulla väärin. Huonon kuulon takia Pepperille on helpompi rakentaa niin sanottuja yksinkertaisia keskusteluja. Yksinkertaisissa keskusteluissa erilaisten vastausmahdollisuuksien sijaan vastaukseksi merkitään asteriski, joka ilmaisee kaikkea. Näissä keskusteluissa Pepper reagoi kaikkeen mitä kuulee sen sijaan, että reagoisi tiettyyn sanaan.

8 PEPPERIN YHDISTÄMINEN CHATBOTTIIN

Tutkittiin voiko Savonian Pepperin yhdistää chatbottiin ilman päivitystä. Useassa ohjeessa neuvotaan kuinka Pepperin voi yhdistää chatbottiin, kunhan siihen on ensin tehty kallis päivitys, jonka myötä Pepperiin voi tehdä Android studion avulla ohjelmia. Päivityksen myötä kuitenkin katoaisi nykyiset tavat tehdä sovelluksia ja se tarkoittaisi sitä, että nykyiset ohjelmat lakkaisivat toimimasta.

8.1 Pepperin yhdistäminen IBM ja Watson chatbottiin

Yksi mahdollisista vaihtoehdoista, jotka eivät vaadi Pepperin päivitystä, on yhdistää se IBM:n ja Watsonin chatbottiin, jolloin chatbotin kanssa voisi Pepperin kautta keskustella raha-asioista. Ohjeet tähän Chatbot kokeiluun löytyy GitHubin –sivuilta, josta on myös ladattava paketti. Ensimmäiseksi oli luotava ilmainen tili IBM:n pilvipalveluun, joka ohjeiden mukaan käy nukkumaan 30 aktiivittoman päivän kuluttua.

KUVA 12. IBM –pilvipalvelutili (Mertanen, 2020)

Tilin luomiseen käytetään henkilökohtaista sähköpostia, joka toimii käyttäjätunnuksena. Salasanaksi saa itse tehdä sopivan salasanan (KUVA 12.). Tilin luomisen jälkeen on etsittävä sivuston katalogista Node Red -sovellus ja luotava se. Node Red-sovelluksen luonnissa on tärkeä muistaa mihin maahan sijoitti pilvipalvelun tilin luomisen aikana. Ilmaistilillä voi olla vain yksi tällainen sovellus käytössä kuten myös vain yksi pilvipalvelu, joka on sama kuin käyttäjän luoma tili. Jotta Node Red-sovellusta voisi muokata, on se laitettava käyntiin. Kun sovellusta on laittamassa käyntiin, on sille tehtävä API avain, valittava sama maa, johon sijoitti pilvipalvelut tilin luomisessa, valittava tarpeeksi tilaa in-

stanssien luomiseen ja organisaatio sekä tallennustila. Organisaatio ja tallennustila täyttyvät automaattisesti, kun kaikki menee hyvin. Organisaatioksi valikoituu tässä tapauksessa käyttäjän tili sähköpostitunnuksella (KUVA 13.).

KUVA 13. Node Red sovelluksen luonti (Mertanen, 2020)

Kun sovellus on laitettu käyntiin, voi sitä tarkastella ja muokata. Käynnistettyyn Node Red-sovellukseen pääsee sisälle linkin avulla sitä kaksoisklikaten (KUVA 14.). Ensin on luotava Node Red –tunnukset: käyttäjä ja salasana. Kun tunnukset on luotu sovellukseen, pääsee sitä lopulta käyttämään.

Details	
App URL	https://noderobosano.eu-gb.mybluemix.net
Source	https://eu-gb.git.cloud.ibm.com/peweetu/Noderobosano
Resource group	Default
Deployment target	Node_robosano

KUVA 14. Kaksoisklikkaa tuota (Mertanen, 2020)

Seuraavaksi lisättiin sovelluksen resursseiksi Create services-valikosta tekoälyn alta Watsonin text to speech service. Nyt näkyvänä resurssina painettiin open dashboard-painiketta, jonka avulla pääsee tutkimaan tekstistä puheeksi tunnuksia. Tunnukset on hyvä ottaa ylös esimerkiksi tyhjään tekstitiedostoon, koska niitä tarvitaan jatkossa. Seuraavaksi lisätään Create services-valikosta Watsonin assistant palvelu. Myös Watson assistant palvelun tunnukset on otettava ylös. Watsonin assistantin Manage sivulta, laukaistaan palvelu. Watsonin assistant sovelluksessa lisätään GitHub- (IBM, 2017) (Europe, 2019) (Robotics)paketin workspace json pohjainen tiedosto. Kun sovellus on nyt päivittynyt, on entities listasta mentävä koneen entiteihin ja laitettava päälle sys date ja sys number painikkeet. Node Red –sovellukseen ladattiin kaksi dialogi-tiedostoa GitHubin paketista. Toiseen dialogi –tiedostoon oli muokattava NODE_BASED_URL –noden kohdalle sama internetosoite kuin on Node Redin osoitekin. Seuraavaksi on muokattava assistant nodea, jota kaksoisklikatessa avautuu samanlainen näkymä kuin NODE_BASED_URLin nodessakin muokattava ikkuna. Koska enää ei löydy Watson workspace ID:ä käytetään assistant ID:ä, joka löytyy Watson assistantin Apin details näkymässä (KUVA 15.).

KUVA 15. Node Red assistant node (Mertanen, 2020)

Muutosten jälkeen painettiin Node Redin deploy painiketta. Seuraavaksi etsitään katalogista Watson Studio ja siihen kirjaudutaan sisään IBM-tunnuksilla. Watson Studiossa luotiin uusi projekti. Uuden projektin luotua ladattiin sinne Notebook –tiedosto. Siihen oli ensimmäisenä tehtävä muokkauksia. Ensimmäisen muokattiin tiedoston toista osiota, niin että ladattiin Pandas Framit koodiin ja muutettiin viidennen osion websocket URL vastaamaan projektin omaa (KUVA 16.).

```
def start_websocket_listener():
    websocket.enableTrace(True)
    ws = websocket.WebSocketApp("ws://noderobosano.eu-gb.mybluemix.net/ws/orchestrate",
                                on_message = on_message,
                                on_error = on_error,
                                on_close = on_close)

    ws.on_open = on_open
    ws.run_forever()
```

KUVA 16. Linkki piti muuttaa (Mertanen, 2020)

Seuraavaksi käynnistettiin GitHub paketin Choregraphe –tiedosto. Choregraphessa Watson tts –laatikkoon päivitettiin käyttäjätunnuksen ja salasanan kohdalle watsontexttospeech servicestä apikeyn ja API keyn tunnukset tunnuksiksi, apikey käyttäjätunnukseksi ja salasanaksi API key. Viimeiseen laattikkoon oli päivitettävä Node red sovelluksen internetosoite. Koska käyttäjätunusta ja salasanaa ei enää käytetä, löydettiin ohjeet GitHubin sivustolla olevalta youtube –videon kommenttiosiota. Aivan lopuksi ajettiin ohjelma robotilla.

8.1.1 Ongelmat IBM ja Watson chatbot ohjelman teossa

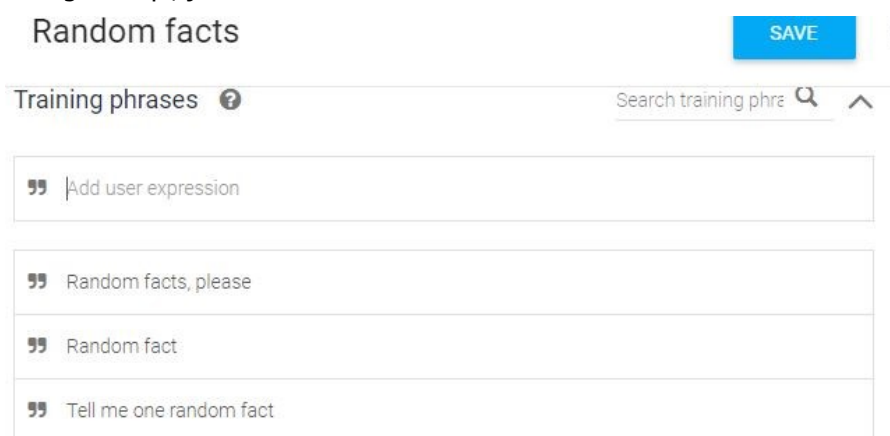
Ensimmäinen ongelma on, että ohjeet on tehty kolme vuotta sitten. Vuosien aikana IBM:n sivusto on muuttunut ja sovellukset myös. Aiemmin Node Red ohjelma oli Node Red starter. Koska ohjelmat on muuttuneet vuosien aikana, ei ohjeita pysty täysin seuraamaan. Aluksi ongelmia oli Node Red sovelluksen käyttöönotossa. IBM:n sivusto kuitenkin auttoi pääsemään alun ongelmista eteenpäin. Toinen ongelma syntyy viimeisessä kohdassa ohjeita, kun ohjelmaa ajaa robotilla. Choregraphe havaitsee virheen, kun tietoa ollaan viemässä JSON-tiedostoon. Choregraphen logissa lukee vaihtelevia virheitä, se ei aina ole sama virhe kun ohjelmaa ajaa.

Samana päivänä kun oltiin rakennettu ohjelma ja ajettu sitä, se oli kaatunut ja muuttunut epävaakaaksi, myöhemmin IBM oli sulkenut ohjelman. Ongelmat Choregraphessa saattoivat johtua tästä. Yrittäessä luoda projektia uudelleen muutamia kertoja, kaikki tyssäsi Node Red sovelluksen luomiseen. Kolmannella kerralla projektin luonti onnistui uudelleen. Aluksi kaikki näytti sujuvan paremmin

kuin ensimmäisellä kerralla, mutta ongelmia syntyi jälleen viimeisessä vaiheessa. Roboteilla testaa- minen ei onnistu. Tarkoituksena olisi että Pepper nauhoittaisi puhetta, jota IBM:n puolella ohjelma analysoisi ja etsisi vastauksen. Ongelmat kuitenkin syntyvät siinä kohtaa ohjelmaa, kun puhe siirtyisi analysoitavaksi. Ohjelma kulkee ohjelmointiympäristössä nauhoituslaatikkoon, josta se vielä siirtyy seuraavaan kohtaan, jossa se analysoisi nauoitettua puhetta IBM:n puolella. Ongelmat saattavat olla IBM:n puolella.

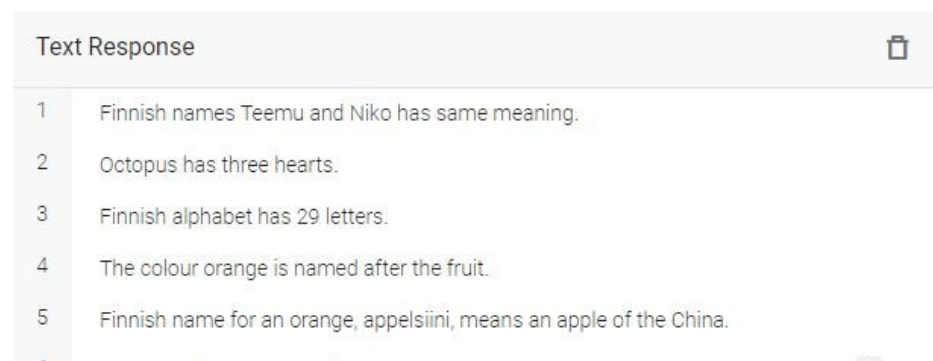
8.2 Pepperin yhdistäminen Googlen Dialogflowhon

Ohjeet Pepperin yhdistämiseen Googlen Dialogflowhon löytyi parista erillisestä lähteestä. Toinen läh- teistä oli Mediumin sivusto ja toinen oli SoftBank Roboticsin kehittäjien sivut. Aluksi seurattiin Soft- Bank Roboticsin ohjeita. Ensimmäiseksi luotiin Google cloud projectiin ilmaisetili, jossa saa tehdä 12 projektia. Tili luotiin google-tunnuksilla. Tunnusten luonnin aikana GCP generoi json-tiedoston tun- nuksia varten. Projektin nimeksi annettiin Jokes. Projektin luotua etsittiin Google cloudin katalogista Dialogflow Api, ja se enablettiin.



KUVA 17. Harjoituslauseita (Mertanen, 2020)

Dialogflowssa luotiin ensin ohjeiden mukaisesti intentti say, joka pyytää robottia kertomaan vitsin. Luotiin myös toinen uusi intentti Random facts. Random facts-intenttiin kirjoitettiin harjoituslauseita (KUVA 17.).



KUVA 18. vastauksia (Mertanen, 2020)

Harjoituslauseisiin robotti vastaa tekstivastauksin, satunnaisella tiedolla (KUVA18.). Kaikki on kirjoj- tettu englanniksi, koska GCP on englanninkielinen. Samalla tavalla myös toimii ohjeiden opastamana

luotu say-intentti. Varsinaisen koodipuoli kirjoitettiin seuraten Mediumin sivua. Medium tarjosi koodin ohjelmaa varten, taustalla toimiva GCP-projekti oli osattava tehdä itsenäisesti tai löytää muualta ohjeita siihen. Koodi tallennetaan python tiedostoksi.

8.3 Ongelmia Dialogflow sovelluksen kanssa

Dialogflown avulla rakennettu sovellus ei lähtenyt käyntiin Pepperin kanssa. Ongelmia ratkottiin kokeilemalla muuttaa koodia, koska virhe on python-tiedostossa. Kun yksi virhe saatiin korjattua, ilmaantui toinen virhe. Virheiden korjaamisketjulle ei näytä tulevan loppua, aina vain ilmaantuu uusi virhe toisen jälkeen. Sovellus ei halua yhdistyä Pepperiin, vaikka osoitteet ja portit robottiin liittyen on kirjattu tiedostoon oikein.

8.4 Pepper robotista chatbot

Kun IBM ja Dialogflow yritykset muodostaa yhteys Pepperillä chatbottiin epäonnistui, ajateltiin vielä kokeilla voisiko Pepperistä tehdä pienimuotoista chatbottia. Robotilta voi kysyä kellon aikaa, päivämäärää ja nimipäivää. Sovelluksen teossa käytettiin Pepperille tehtyä sovellusta, joka hyödyntää robotin tietokantaa. Päivämäärän ja kellonajan Pepper hakee internet-yhteyden avulla. Nimipäivätiedot roboti hakee sqlite3-pohjaisesta tietokannasta, joka on Pepperin tukema tietokantatyyppe.

Aivan ensimmäiseksi Pepper kysyy käyttäjältä mitä tietoa haetaan. Vastaukset kirjataan vaustausvaihtoehdoilla dialogilaatikkoon, josta vastausten perusteella siirrytään python-laatikkoon, johon on kirjoitettu Pepperin antama vastaus. Päivämäärä ja kellonaika tuodaan python-laatikossa datetime muodossa (KUVA 19.).

```
def onInput_onStart(self):
    #self.onStopped() #activate the output of the box
    aika = datetime.datetime.now()
    tunti = aika.hour
    minuutti = aika.minute
    tunt = str(tunti)
    min = str(minuutti)
    nyt = "kello on nyt "+tunt+" "+min+" olkaa hyvä."
    self.tts.setParameter("speed",80)
    self.tts.say(nyt)
    self.onStopped()
```

KUVA 19. Kellonaika (Mertanen, 2020)

On myös keksittävä määreet, jotka ovat string-muuttujia, koska Pepperin antama vastaus tulee tekstistä muodostetusta puheesta. Sen takia kellonaika ja päivämäärä on muutettava stringeiksi. Päivämääräkuvassa (KUVA 20.) Internetistä haetut tiedot muutetaan string muotoon.

```
def onInput_onStart(self):
    #self.onStopped() #activate the output of the box
    tanaan = date.today()
    paiva = tanaan.day
    kuukausi = tanaan.month
    vuosi = tanaan.year
    vuos = str(vuosi)
    kuuk = str(kuukausi)
    paiv = str(paiva)
    if(paiva <10):
        paiv = "0"+str(paiva)
    else:
        paiv=str(paiva)
    if(kuukausi < 10):
        kuuk = "0"+str(kuukausi)
    else:
        kuuk = str(kuukausi)
    koko = " "+paiv+" "+kuuk+" "+vuos
    self.tts.setParameter("speed",85)
    self.tts.say(koko)
    self.onStopped()
```

KUVA 20. Päivämäärä (Mertanen, 2020)

Nimipäivätietoja hakiessa Pepper vertaa kyseistä päivää sqlite3-pohjaisessa tietokannassa olevan tauluun tallennettuja tietoja ja hakee nimipäivän (KUVA 21.). Sqlite3 tietokantaan on päivämäärän

lisäksi tallennettu nimipäivät ja nimipäivään liittyvää nippelitietoa.

```
def haeKannasta(self, conn):
    tanaan = date.today()
    paiva = tanaan.day
    kuukausi = tanaan.month
    vuosi = tanaan.year
    vuosi = str(vuosi)
    kuuk = str(kuukausi)
    paiv = str(paiva)
    if(paiva < 10):
        paiv = "0"+str(paiva)
    else:
        paiv=str(paiva)
    if(kuukausi < 10):
        kuuk = "0"+str(kuukausi)
    else:
        kuuk = str(kuukausi)
    koko = ""+paiv+"."+kuuk+"."
    cur = conn.cursor()
    cur.execute("SELECT teksti FROM nimip WHERE paiva"+"="+koko)
    row = cur.fetchall()
    try:
        return row[0]
    except IndexError:
        return None
```

KUVA 21. Tietokannasta hakeminen (Mertanen, 2020)

9 POHDINTA

Jos Pepperin yhdistäminen IBM-Watsoniin olisi onnistunut, olisi siltä voinut kysyä englanninkielisiä kysymyksiä markkinointiin liittyen. Se ei sellaisenaan olisi palvellut Savonian tarpeita. Olisin kuitenkin mahdollisesti voinut muokata sitä esimerkiksi vastaamaan kysymyksiin liittyen vanhustenhoitoon. Kieli olisi kuitenkin pysynyt englannissa. Englannin kieli on kuitenkin ongelmallinen Pepperin kanssa, kun sen pääkieli on vaihdettu suomeen.

Jos Pepper olisi saatu yhdistettyä Googlen Dialogflown kanssa, se olisi saatu esittämään vitsejä. Samalla tavalla sille olisi voinut rakentaa kysymyksiä ja vastauksia liittyen eri asioihin. Kieli olisi edelleen pysynyt englannissa. Vaikka Pepper olisi saatu yhdistettyä Chatbottiin, olisiko se ollut erilainen kuin sen omalla dialogin moottorilla rakennettu chatbot.

Kuten Choregraphessa rakennetussa dialogissa, myös Googlen Dialogflowssa piti ensin rakentaa kysymykset ja niihin vastaukset. Olen oppinut sen yrittäessäni yhdistää Pepperiä chatbottiin, että Chatbottiin pitää kirjoittaa kysymykset ja vastaukset, ainakin Dialogflown kohdalla. Choregraphessa myös on kirjoitettava dialogi, jota Pepper seuraa ja vastaa kysymyksen tai vastauksen mukaan, jos robotti sattuu kuulemaan kysymyksen tai avainsanan.

10 JATKO

Jatkossa Pepperin yhdistämistä Chatbottiin voisi kokeilla Android-pohjaisena. Toki siihen pitäisi ensin pyytää lupaa ja pohtia onko Pepperin päivittäminen tarpeellista. HyvinRobo-hankkeen aikana se ei ollut tarpeellista, koska sen sovellukset olisivat lakanneet toimimasta mikä olisi ollut takaisku hankkeelle, opinnäytetyö tehtiin hankkeen aikana. Nyt hanke on loppunut ja Pepperiin tehdyt ohjelmat tallennettu toisaalle, joten mahdollisesti Pepperin päivittäminen onnistuisi. Ohjeita android pohjaisen Pepperin yhdistämiseen chatbottiin, löytyy useita verrattuna python pohjaiseen Pepperiin. Jos robotti aiotaan päivittää ajatuksella rakentaa android-pohjainen chatbot-keskustelu Pepperille, on hyvä miettiä mikä on sen tulevaisuus ja missä sitä aiotaan käyttää. Niiden asioiden pohjalta on helpompi rakentaa kysymyksiä ja vastauksia, kun tietää kohteet ja tavoitteet.

LÄHTEET JA TUOTETUT AINEISTOT

- Association, Robotic Industries.** Robotics Online Humanoid Robots. *Robotics Online*. [Online] [Viitattu: 20. Toukokuu 2020.] <https://www.robotics.org/service-robots/humanoid-robots>.
- BotSide. 2017.** 3 Major problems with chatbots and chatbot development Medium. *Medium*. [Online] 14. Elokuu 2017. [Viitattu: 22. Toukokuu 2020.] <https://medium.com/@botsideapp/3-major-problems-with-chatbots-and-chatbot-development-503d84e176aa>.
- ChatbotPack. 2019.** What is chatbot? Introduction to chatbots 2019. *ChatbotPack sivuston blogikirjoitus*. [Online] ChatbotPack, 2019. [Viitattu: 2. Toukokuu 2020.] <https://www.chatbotpack.com/what-is-chatbot-2019>.
- Dang, Sanjit Singh. 2019.** Artificial Intelligence In Humanoid Robots. *Forbes*. [Online] 25. Helmikuu 2019. [Viitattu: 20. Toukokuu 2020.] <https://www.forbes.com/sites/cognitiveworld/2019/02/25/artificial-intelligence-in-humanoid-robots/#190e2b9724c7>.
- Europe, Richard EmTech. 2019.** Medium. [Online] 11. Helmikuu 2019. [Viitattu: 2. Heinäkuu 2020.] <https://medium.com/@blogemtech/pepper-integration-with-dialogflow-1d7f1582da1a>.
- Haikonen, Pentti O. A. 2017.** *Tietoisuus, tekoäly ja robotit*. Helsinki : Art House, 2017. ss. 141-145.
- Hupli, Mikko. 2018.** Chatbot FAQ - kaikki mitä chatboteista on syytä tietää juuri nyt. *Salesforcen blogikirjoitus*. [Online] Salesforce, 2. Heinäkuu 2018. [Viitattu: 2. Toukokuu 2020.] <https://www.salesforce.com/fi/blog/2018/chatbot-usein-kysytyt-kysymykset.html>.
- IBM. 2017.** GitHub. [Online] 3. Lokakuu 2017. [Viitattu: 21. Kesäkuu 2020.] <https://github.com/IBM/watson-nao-robot>.
- Koskinen, Anu L. 2019.** Eeva-Maija Sinkkonen ilmoitti vakuutusyhtiölle vauvastaan – Teppo-robotin vastaus ei mennyt putkeen, siitä tuli somehitti. *YLE*. [Online] 25. Syyskuu 2019. [Viitattu: 9. Heinäkuu 2020.] <https://yle.fi/uutiset/3-10989177>.
- Laukkonen, Jouni. 2018.** Suomenkielinen chatbot. *Salescommunications sivun blogikirjoitus*. [Online] Salescommunications, 8. Tammikuu 2018. [Viitattu: 2. Toukokuu 2020.] <https://www.salescommunications.fi/blog/suomenkielinen-chatbot>.
- Mankkinen, Jussi. 2020.** Kärsitkö robofobiasta? Syynä ovat ihmisaivot, jotka tulkitsevat liian oikealta vaikuttavan robotin sairaaksi tai psykopaatiksi. *Yle*. [Online] 20. Helmikuu 2020. [Viitattu: 12. Toukokuu 2020.] <https://yle.fi/uutiset/3-11214561>.
- Mertanen, Petrus. 2020.** *Aloitus*. Kuopio : 2020.
- . 2020. *Assistant node*. Kuopio : 2020.
 - . 2020. *Dialogilaatikko, jolle on annettu nimi, nimi*. Kuopio : 2020.
 - . 2020. *Faktat*. Kuopio : 2020.
 - . 2020. *Fif.top -tiedosto*. Kuopio : 2020.
 - . 2020. *Harjoituslauseita_Dialogflowhon*. Kuopio : 2020.
 - . 2020. *IBM -pilvipalvelutili*. Kuopio : 2020.
 - . 2020. *Kellonaika*. Kuopio : 2020.
 - . 2020. *Kysymys ilkeivallasta chatbotille*. Kuopio : 2020.
 - . 2020. *Lipun lasku*. Kuopio : 2020.

- , **2020.** *Lisää tuo.* Kuopio : 2020.
- , **2020.** *Muutos.* Kuopio : 2020.
- , **2020.** *Nimi ja kehut.* Kuopio : 2020.
- , **2020.** *Ohi kontekstin.* Kuopio : 2020.
- , **2020.** *Paina tuosta.* Kuopio : 2020.
- , **2020.** *Porvoon herkut.* Kuopio : 2020.
- , **2020.** *Porvoon nähtävyydet.* Kuopio : 2020.
- , **2020.** *Päivämäärä.* Kuopio : 2020.
- , **2020.** *Sisennys.* Kuopio : 2020.
- , **2020.** *Suomen lipun historia.* Kuopio : 2020.
- , **2020.** *Teppo Turvan ongelma.* Kuopio : 2020.
- , **2020.** *Tietokannasta hakeminen.* Kuopio : 2020.

Poole, Harry H. 2012. *Fundamentals of Robotics Engineering.* s.l. : Springer Science & Business Media, 2012. ss. 3-5.

PYMNTS. 2018. Hotel, Banking Customers Embrace SoftBank's Humanoid Robot, Pepper. *PYMNTS.* [Online] PYMNTS, 30. Heinäkuu 2018. [Viitattu: 3. Toukokuu 2020.]
<https://www.pymnts.com/news/artificial-intelligence/2018/pepper-robot-softbank-robotics-technology-innovation>.

Robotics, Developers SoftBank. SoftBank Robotics. [Online] [Viitattu: Heinäkuu. 2 2020.]
<https://developer.softbankrobotics.com/pepper-qisd/lessons/integrating-chatbot-dialogflow/>.

Robotics, SoftBank. Pepper. *Pepper robotin esittelysivu.* [Online] SoftBank Robotics. [Viitattu: 26. Huhtikuu 2020.] <https://www.softbankrobotics.com/emea/en/pepper>.

Robotit hoitavat tylsät työt. **Kuparinen, Jesse. 2020.** s.l. : Iltasanomat, 3. Helmikuu 2020, Iltasanomat sanomalehtiweekin erikoisjulkaisu, s. 16.

Seuri, Ville. 2020. Yle. *Tekoäly sai potkut Yleltä, mikä on konejournalismin tulevaisuus? "Tekoäly ei pysynyt ihmisen luovuuden perässä".* [Online] 25. Tammikuu 2020. [Viitattu: 12. Toukokuu 2020.]
<https://yle.fi/uutiset/3-11171589>.

Timo Siukonen, Pekka Neittaanmäki. 2019. *Mitä tulisi tietää tekoälystä.* Jyväskylä : Docendo, 2019.