

KARELIA-AMMATTIKORKEAKOULU  
Tietojenkäsittelyn koulutusohjelma

Tuomas Kalliola

ASIAKASPALVELUBOTIN TOTEUTUS

Opinnäytetyö  
Marraskuu 2017



**OPINNÄYTETYÖ**  
**Marraskuu 2017**  
**Tietojenkäsittelyn koulutusohjelma**

Tikkarinne 9  
80220 JOENSUU  
(013) 260 600

Tekijä  
Tuomas Kalliola

Nimeke  
Asiakaspalvelubotin toteutus

Toimeksiantaja  
Projektitoimisto Proper Oy

**Tiivistelmä**

Tämä toiminnallinen opinnäytetyö käsittelee prototyyppi-chatbotin toteuttamista, jolla mahdollistetaan tuotehaut Shopify-verkkokauppa-alustalle luonnollista kieltä käyttäen. Toteutuksesta laadittiin kaksi eri versiota, joista toinen mahdollistaa tuotehaut suomeksi ja toinen englanniksi. Lisäksi prototyypin ohessa luotiin testiverkkokauppa Shopify-alustalle.

Opinnäytetyön tarkoituksena oli selvittää, onko prototyyppi mahdollista toteuttaa tehtävään valituilla työkaluilla ja tarkastella, kuinka kattava hakutoiminto opinnäytetyön aikana olisi mahdollista toteuttaa. Prototyypin chatbot rakennettiin Microsoftin Bot Frameworkilla ja luonnollisen kielen ymmärtämisen lisääminen toteutettiin Microsoftin LUIS-palvelulla sekä Wit.ai-palvelulla. Lisäksi opinnäytetyössä selvitettiin eroja luonnollisen kielen ymmärtämisen lisäämisessä suomen ja englannin kielen välillä kyseisten työkalujen osalta.

Teoreettinen viitekehys antaa lukijalle yleiskuvan tekoälystä asiakaspalvelun tukena, chatboteista sekä luonnollisen kielen käsittelystä. Prototyypin toteutus kuvataan aiheisiin tutustuvan opiskelijan näkökulmasta, ja se soveltuu näin kaikille chatbotien kehittämisestä ja niiden luonnollisen kielen ymmärtämisen lisäämisestä kiinnostuneille.

Prototyyppi saatiin opinnäytetyön aikana toteutettua siihen valituilla työkaluilla. Toimeksiantaja hyväksyi prototyypin ja ilmoitti sen täyttävän sille asetetut vaatimukset sekä odotukset. Suomen ja englannin kielen ymmärtämisen lisäämisessä valituilla työkaluilla löydetään useampia pieniä eroja. Erityisesti suomen kielen kohdalla havaitaan tarve työkalulle, jonka avulla sanoista voidaan karsia sijamuotoja.

Kieli  
suomi

Sivuja 77  
Liitteet 0

**Asiasanat**

tekoäly, asiakaspalvelu, chatbot, luonnollisen kielen käsittely, Microsoft Bot Framework, LUIS, Wit.ai, Shopify



**THESIS**  
**November 2017**  
**Degree programme in Business Information Technology**

Tikkarinne 9  
80220 JOENSUU  
(013) 260 600

Author  
Tuomas Kalliola

Title  
Implementation of Customer Service Chatbot

Commissioned by  
Project office Proper Oy

#### Abstract

This functional thesis covers the implementation of prototype chatbot, which enables the user to search products from Shopify e-commerce platform by using natural language. The prototype included two different versions of the implementation. One which allows the user to search products in Finnish and one which allows the searches to be made in English. In addition, a development store was created for testing purposes.

The objective of the thesis was to find out whether the prototype could be implemented by using the selected tools and to find out how a comprehensive search function could be created. The chatbot was built using Microsoft Bot Framework and adding its capabilities for understanding natural language was achieved by using Microsoft's LUIS-service and Wit.ai-service. In addition, the thesis analyzed differences of adding natural language understanding in Finnish and English when the selected tools are used.

The theoretical framework gives the reader an overview about the usage of artificial intelligence in customer service, chatbots and natural language processing. The implementation of the prototype is described from the perspective of a student who is getting familiarized with the topics and it suits all readers who are interested in developing chatbots or adding their natural language understanding.

The prototype was completed during the thesis by using the selected tools. The commissioner of the thesis accepted the prototype and announced that it met the given requirements and expectations. Several minor differences were found between adding natural language understanding in Finnish and English by using the selected tools. Especially in case of Finnish language, a need for a tool which could hyphenate words was found.

Language

Pages 77

Finnish

Appendices 0

Keywords

artificial intelligence, customer service, chatbot, natural language processing, Microsoft Bot Framework, LUIS, Wit.ai, Shopify

# Sisältö

|        |  |    |
|--------|--|----|
| 1      | Johdanto.....  | 7  |
| 2      | Tekoäly asiakaspalvelun tukena.....                              | 10 |
| 2.1    | Yritykset ja asiakkaat valmiina asiakaspalvelun muutoksille..... | 10 |
| 2.2    | Tekoälypohjaisen asiakaspalvelun edut.....                       | 11 |
| 2.3    | Suuret investoinnit – suuret ongelmat.....                       | 12 |
| 3      | Chatbotit.....   | 15 |
| 3.1    | Chatbotin määritelmä.....  | 15 |
| 3.2    | Chatbotien historia.....   | 16 |
| 3.3    | Chatbotit nykypäivänä.....                                       | 18 |
| 4      | Luonnollisen kielen käsittely.....                               | 23 |
| 5      | Työn tausta ja tavoitteet.....                                   | 25 |
| 5.1    | Prototyypin tavoitteet.....                                      | 25 |
| 5.2    | Prototyypin rajaukset.....                                       | 26 |
| 6      | Asiakaspalvelubotin prototyypin toteutus.....                    | 27 |
| 6.1    | Toteutukseen valitut työkalut ja alustat.....                    | 27 |
| 6.1.1  | Microsoft Bot Framework.....                                     | 31 |
| 6.1.2  | LUIS.....  | 32 |
| 6.1.3  | Wit.ai.....  | 33 |
| 6.1.4  | Shopify.....   | 34 |
| 6.2    | Toteutuksen prosessi.....  | 36 |
| 6.2.1  | Toteutuksen toimintaperiaatteet ja arkkitehtuuri.....            | 36 |
| 6.2.2  | Shopify-rajapinnat ja yhdistämismenetelmä.....                   | 38 |
| 6.2.3  | Testiverkkokaupan luominen.....                                  | 40 |
| 6.2.4  | Projektin alustus ja Microsoft Bot Builderin käyttöönotto.....   | 42 |
| 6.2.5  | Testaaminen emulaattorissa.....                                  | 44 |
| 6.2.6  | Perusdialogit.....   | 45 |
| 6.2.7  | NLU-palveluiden käyttöönotto ja yhdistäminen chatbotiin.....     | 47 |
| 6.2.8  | LUISin ja Wit.ai:n peruskonseptit.....                           | 49 |
| 6.2.9  | Mallien harjoittamisen ongelmat.....                             | 52 |
| 6.2.10 | Prototyypille luodut mallit.....                                 | 55 |
| 6.2.11 | Chatbotin hakudialogi.....                                       | 60 |
| 6.2.12 | Shopify-hakufunktiot.....  | 63 |
| 6.2.13 | Lokalisointi.....  | 64 |
| 7      | Tulokset ja johtopäätökset.....                                  | 65 |
| 7.1    | Prototyypin valmistuminen ja arviointi.....                      | 66 |
| 7.2    | Kielten väliset erot prototyypissä.....                          | 67 |
| 7.3    | Hakutoiminnon kattavuus.....                                     | 69 |
| 7.4    | Prototyypin jatkokehitys.....                                    | 70 |
| 8      | Pohdinta.....  | 71 |
|        | Lähteet.....   | 74 |

## Käsitteet

|                         |  |
|-------------------------|--|
| Chatbot                 | Lyhenne sanoista chat robot. Tietokoneohjelma, joka on suunniteltu simuloimaan keskustelua ihmiskäyttäjän kanssa, kommunikation perustuessa teksti- tai ääniviestintään. (Oxford University Press 2017; TechTarget 2016).  |
| JSON                    | Akronyymi sanoista JavaScript Object Notation. Kevyt tiedonsiirtoformaatti, joka on helppolukuista ihmiselle ja koneille helppo jäsenellä ja tuottaa. (JSON.org 2017.)   |
| LUIS                    | Lyhenne sanoista Language Understanding Intelligent Service. Microsoftin palvelu ihmiskieltä ymmärtävien älykkäiden sovellusten luomiseen. (Microsoft Azure 2017a.)  |
| Microsoft Azure         | Microsoftin sarja pilvipalveluita, joiden avulla kehittäjä ja IT-alan ammattilaiset voivat rakentaa, käyttöönottaa sekä hallita sovelluksia (Microsoft Azure 2017b).   |
| Microsoft Bot Framework | Microsoftin vuonna 2016 julkaistu alusta chatbotien rakentamiseen ja julkaisemiseen eri kanavilla (Microsoft 2017a). Ohjelmistokehys mahdollistaa älykkäiden ja tehokkaiden chatbotien rakentamisen, yhdistämisen, testauksen sekä käyttöönoton (Microsoft 2017b). |

|                                  |  |
|----------------------------------|--|
| Microsoft Bot Builder            | Microsoft Bot Frameworkin komponentti, joka on tarkoitettu chatbotien rakentamiseen (Microsoft 2017b).   |
| Luonnollisen kielen käsittely    | Yksi tietojenkäsittelytieteiden, kielitieteiden ja tekoälyn osa-alueista, jonka tarkoituksena on mahdollistaa tietokoneille kyky käsitellä ihmisten käyttämää luonnollista kieltä (Chopra, Prashar & Sain 2013).               |
| Luonnollisen kielen ymmärtäminen | Luonnollisen kielen käsittelyn alakategoria, joka keskittyy tietokoneen kykyyn ymmärtää ihmisten käyttämää luonnollista kieltä (Tutorialspoint 2017).  |
| NLP                              | Kts. Luonnollisen kielen käsittely   |
| NLU                              | Kts. Luonnollisen kielen ymmärtäminen  |
| REST                             | Akronyymi sanoista Representational State Transfer. Arkkitehtuurityyli, joka tarjoaa standardeja järjestelmien väliselle kommunikaatiolle verkossa. (Codeacademy 2017.)  |
| Shopify                          | Kanadasta lähtöisin oleva sähköisen kaupankäynnin ohjelmistoyritys ja yrityksen samannimisen verkkokauppa-alustan nimi. Shopify-alusta on tarkoitettu erityisesti verkkokauppojen suunnitteluun ja julkaisuun. (Silcoff 2013.) |
| Wit.ai                           | Facebookin omistama palvelu puhetta ja tekstiä käsittelevien sovellusten rakentamiseen (TechCrunch 2015; Wit.ai 2017a).  |

## 1 Johdanto

Tekoälyn viime vuosien nopea kehitys ja sen vaikutukset yhteiskuntaamme ovat olleet ajankohtainen ja toistuva aihe viimeaikaisessa uutisoinnissa. Tekoälyyn liittyvät ennusteet ja visiot ovat sisältäneet runsaasti niin mahdollisuuksia kuin mahdollisia uhkakuviakin tulevaisuudellemme. Ensimmäinen teollinen vallankumous käynnistyi 1800-luvun taitteessa höyryvoiman hyödyntämisestä ja koneiden kehityksestä tuotannon tehostamiseksi. Toisessa vallankumouksessa sähkön käyttö mahdollisti massatuotannon. Kolmannessa vaiheessa internetin myötä kehitystä muovasivat digitalisaatio sekä uudet viestintäteknologiat. Nyt neljännen teollisen vallankumouksen ennustetaan käynnistyvän tekoälyn kehityksestä robotiikan ja esineiden internetin ohella. Arvostetun Massachusetts Institute of Technology -yliopiston professori Eric Brynjolfsson esitti Etlan 70-vuotisjuhlissa rohkean toteamuksen, jonka mukaan tekoäly on jopa maailmanhistorian merkittävin teknologinen läpimurto (Liikenne- ja viestintäministeriö, Tekes, Teknologia-teollisuus, Verkkoteollisuus 2017).

Viime aikojen uutisoinnissa tekoälyyn liitettyyn ”*hypeen*” tulisi kuitenkin suhtautua kriittisesti ja varauksella. Vaikka tietokoneidemme laskutehon kasvu ja massiivisten tietomäärien saatavuus ovat mahdollistaneet tekoälyn uuden aallon, ovat sen käytännön sovellutusten löytäminen sekä toteuttaminen osoittautuneet todellisuudessa erittäin haastaviksi ongelmiksi. Jopa suuret teknologiajätit ovat joutuneet korjaamaan ja laskemaan aiempia korkeita tekoälyn liittyviä odotuksiaan.

Tekoälyllä voidaan kuitenkin nähdä olevan suuri hyödyntämispotentiaali yritysten liiketoiminnan tukena. Accenturen Why AI is the future growth -raportissa ennustetaan kehittyvien talouksien olevan transformatiivisten teknologioiden viimeaikaisen lähentymisen myötä siirtymässä kohti uutta aikakautta, jossa tekoälyn avulla voidaan päihittää pääoman ja työvoiman fyysiset rajoitukset sekä avata uusia lähteitä arvolle ja kasvuille. Raportissa tehtyjen analyysien mukaan tekoälyllä on mahdollisuus kaksinkertaistaa kehittyvien talouksien vuosittainen talouskasvu vuoteen 2035 mennessä. (Daugherty & Purdy 2016.) Tekoälyn ajankoh-taisuus ja sen kehityksen tuomat uudet hyödyntämismahdollisuudet siirtävät

aiheen pohdintaa yhä enemmän myös kaikenkokoisten yritysten vastuulle, kun aiemmin tekoälyn tutkimiseen ja kehittämiseen osallistuivat lähes ainoastaan suuret teknologiajätit sekä akateemisen maailman edustajat.

Yksi sovellutusalueista, jonka kautta useat yritykset ovat alkaneet lähestyä tekoälyn hyödyntämistä liiketoiminnassa, on yritysten asiakaspalvelu. Jo vuonna 2011 yhdysvaltalainen tietotekniikan alan tutkimus- ja konsultointiyritys Gartner Inc. esitti ennusteen, jossa vuoteen 2020 mennessä asiakkaiden ja yritysten välisestä vuorovaikutuksesta jopa 85 % pystytään hallitsemaan ilman ihmisen vaikutusta (Gartner Summits 2011).

Tekoälypohjaisen asiakaspalvelun avulla pyritään automatisoimaan yksinkertaisia rutiinitason asiakaspalvelutilanteita ja vapauttamaan ihmisasiakaspalvelijoiden resursseja asiakkaiden monimutkaisempien ongelmien selvittämiseen. Tällä hetkellä yritykset hyödyntävät tekoälyä asiakaspalveluyksiköissään pääasiassa kahdella eri tavalla; käyttämällä suorassa asiakaskontaktissa olevia tekoälypohjaisia chatboteja tai tukemalla ihmisasiakaspalvelijoita tekoälyn avulla (Zhou 2017). Tämän toiminnallisen opinnäytetyön tavoitteena oli tutustua tarkemmin esitellyistä kahdesta tavasta hyödyntää tekoälyä asiakaspalvelussa ensimmäiseen eli tekoälypohjaisiin chatboteihin.

Opinnäytetyön toimeksiantajana oli joensuulainen vuonna 2011 perustettu Projektitoimisto Proper Oy, joka tuottaa laadukkaita asiantuntijapalveluita julkisten yhteisöjen ja yritysten projektien toteuttamiseksi. Properin asiantuntijapalveluiden keskeisiin osa-alueisiin kuuluvat konsultointipalvelut ja projektitoteutukset. (Projektitoimisto Proper Oy 2015.)

Proper Oy:n toimitusjohtaja Teemu Purmonen kertoi yrityksellä olevan vaikeuksia pysyä nopeasti etenevässä kehityksessä mukana itsenäisesti. Tästä syystä opinnäytetyöt ovat heille hyvä tapa päivittää teknologiatasoa ja tehdä selvityksiä eri teknologioiden sovellettavuudesta. Purmonen on vierailut kesäkuussa 2017 Chicagossa järjestetyillä Internet Retail Conference -messuilla, joilla verkkokaupan toimintaa tukevat älykkäät chatbotit olivat esiteltynä korkealla alan tärkeimpien ja innostavimpien suuntausten joukossa. Hän kertoi tämän tukeneen hänen omia



selvityksiään tunnistettaessa asiakaspalveluun suuntautuvat chatbotit nykypäivän megatrendinä. (Purmonen 2017.)

Usein tekoälyn kohdalla esitetty maininta käytännön käyttötapauksien ja sovellustien löytämisen haasteellisuudesta osoittautui paikkansapitäväksi myös opinnäytetyön käytännön toteutusta pohdittaessa. Pitkäkestoisen aiheen pohdinnan tuloksena opinnäytetyössä päädyttiin lopulta toteuttamaan prototyyppi chatbotista, joka mahdollistaa tuotehaun Shopify-verkkokauppa-alustalle. Erityisesti chatbotin taustalle toteutettavan tekoälyn ja sen tason määrittäminen osoittautui hankalaksi, koska osapuolilla ei ollut aiempaa kokemusta tekoälysovellusten teknisistä toteutuksista. Lopulta prototyypissä luotavan chatbotin taustalla oleva tekoäly päädyttiin toteuttamaan hyödyntämällä dialoginhallinnassa luonnollisen kielen käsittelyä. Viimeisimpänä haasteena prototyyppitoteutukseen valittiin luonnollisen kielen käsittelyn toteuttaminen niin suomen kuin englannin kielellä.

Opinnäytetyössä toteutettavan prototyypin kehityksessä pyritään saamaan selville soveltuvatko valitut työkalut Shopify-verkkokaupan tuotehaun toteuttamiseen luonnollisella kielellä chatbot-käyttöliittymän kautta. Lisäksi prototyypin toteutuksen ohessa voidaan saada selvitettyä, mitä eroja luonnollisen kielen käsittelyn toteutuksessa on suomen ja englannin kielen välillä toteutukseen valittujen työkalujen osalta. Prototyypin avulla voidaan myös saada tietoa, kuinka kattava hakutoiminto/luonnollisen kielen käsittely on mahdollista toteuttaa noin kuuksikauden kehitysjakson aikana ilman runsasta aiempaa kokemusta chatbotien toteutuksesta tai luonnollisen kielen käsittelystä ja kun kehitykseen käytetyt resurssit ovat matalat.

Luvut 2–4 käsittelevät opinnäytetyössä toteutettavan chatbotin kannalta oleellisten aiheiden teoreettista viitekehystä. Näitä aiheita ovat tekoälyn hyödyntäminen asiakaspalvelussa, chatbotit sekä luonnollisen kielen käsittely. Luku 5 selventää prototyyppitoteutuksen taustoja esittelemällä prototyypille asetetut tavoitteet sekä rajaukset. Luvussa 6 kuvataan toteutukseen valitut työkalut sekä toteutuksen prosessi ja siinä käytetyt menetelmät. Luku 7 sisältää arvioinnin prototyypin kehityk-

sessä esille tulleista tuloksista. Lisäksi luvussa esitetään ideoita prototyypin jatkokehitystä varten. Kahdeksas ja viimeinen luku käsittää kirjoittajan oman pohdinnan opinnäytetyöstä ja sen sisältämistä aiheista.

## **2 Tekoäly asiakaspalvelun tukena**

### **2.1 Yritykset ja asiakkaat valmiina asiakaspalvelun muutoksille**

Tekoälyn hyödyntäminen yritysten asiakaspalvelussa vaikuttaa olevan muodostumassa yhdeksi nykypäivän ja tulevien vuosien suurimmista trendeistä. Oraclen yhteistyössä Coleman Parkesin kanssa vuonna 2016 tehdyssä tutkimuksessa haastateltiin yhteensä 800 yrityksensä johtoasemissa toimivaa henkilöä Ranskassa, Hollannissa, Etelä-Afrikassa ja Isossa-Britanniassa. Vastajat tulivat kolmelta eri teollisuuden alueelta: tehdasteollisuuden, huipputeknologian- ja verkko-kaupan- sekä tietoliikenteen alalta. Vastanneista yrityksistä 78 % ilmoittivat jo ottaneensa käyttöön tai suunnittelevansa tekoälyn ja virtuaalitodellisuus-tekniologioiden hyödyntämistä palvellakseen asiakkaitaan entistä paremmin. Yrityksistä 80 % oli ottanut käyttöön tai suunnitteli chatbotien käyttöönottoa vuoteen 2020 mennessä. (Oracle 2016.)

Yritysten kiinnostuksen tekoälypohjaiseen asiakaspalveluun vahvistaa myös yhdysvaltalaisen vähittäiskaupan hallinnan konsultointiyrityksen BRB:n vuonna 2017 tehty kysely, jonka tarkoituksena oli selvittää vähittäiskauppioiden näkemyksiä asiakaskokemusten prioriteeteista. Kyselyyn osallistui yli 500 pohjois-amerikkalaista vähittäiskaupan edustajaa. Kyselyyn vastanneista yrityksistä jo 45 % suunnittelee tekoälyn käyttöönottoa chatbotien tai digitaalisten assistenttien muodossa kolmen vuoden sisällä parantaakseen yrityksensä asiakkaiden kokemaa asiakaskokemusta. (BRB 2017.)

Asiakaspalvelun muutoksen taustalla voidaan nähdä osittain samoja taustatekijöitä, jotka ovat viime vuosikymmenen aikana muuttaneet myös kaupankäyntiä.

Digitalisaatio ja mobiililaitteiden lisääntyminen ovat saaneet asiakkaat hankki-  
maan entistä enemmän tietoa omatoimisesti ennen ostospäätöksen tekoa tai et-  
sittäessä ratkaisua ongelmatilanteisiin. Aspect Software Inc.:in vuonna 2016  
amerikkalaisten kuluttajien asiakaspalvelukokemuksia selvittäneen kyselyn mu-  
kaan 18–65 -vuotiaiden ikäluokasta jopa 65 % tunsivat olonsa hyväksi onnistuttu-  
aan saamaan vastauksen asiakaspalveluongelmaansa puhumatta ihmisen  
kanssa. Saman vastauksen kohdalla ero edellisvuoteen oli kasvanut 8 %. Kyse-  
lyyn vastanneista 49 % olisivat valmiita hoitamaan kaikki asiakaspalvelukontak-  
tinsa joko tekstiviestien, chattien tai pikaviestimien välityksellä. (Aspect Software  
Inc 2016.)

## **2.2 Tekoälypohjaisen asiakaspalvelun edut**

Mainittujen tutkimusten mukaan yritykset ja asiakkaat vaikuttavat olevan entistä  
valmiimpia ja avoimempia ilman ihmiskontaktia tapahtuvalle vuorovaikutukselle  
asiakaspalvelutilanteissa. Yksi tekoälypohjaisen asiakaspalvelun tavoitteista on-  
kin vastata entistä paremmin ja tehokkaammin asiakkaiden muuttuneisiin tottu-  
muksiin ja parantaa asiakaskokemuksen laatua. Tekoälyllisen asiakaspalvelun  
eduksi asiakaspalvelukokemuksessa voidaan laskea asiakaspalvelun jatkuva  
saatavuus kellon ympäri sekä nopeat vastausajat, mikä olisi erittäin vaikeasti to-  
teutettavissa ihmisresurssein. Forbesin artikkeli esittääkin esimerkkinä China  
Merchant Bankin WeChat-pikaviestinalustalla toimivan chatbotin, joka käsittelee  
1,5–2 miljoonaa asiakaspalveluun liittyvää keskustelua päivittäin. Määrän käsit-  
telemiseen ilman chatboteja vaadittaisiin tuhansia ylimääräisiä ihmisasiakaspal-  
velijoita. (Zhou 2017.)

Runsaasti asiakasdataa keräävät yritykset voivat myös hyödyntää big dataa, ko-  
neoppimista ja tekoälyä personalisoidakseen asiakaspalvelukokemusta. Perso-  
nalisaaation taso voi vaihdella aina tuotesuosituksista verkkosivujen räätälöintiin  
reaaliajassa asiakkaan luku- ja selailutottumusten mukaisesti. Personalisoinnilla  
voidaan parantaa asiakaspalvelun vuorovaikutusta, edistää kuluttajien tyytyväi-  
syyttä, lisätä ostopäätösten syntymistä ja kannustaa asiakasta jatko-ostoihin.  
(Schneider 2017.)

Yksi mielenkiintoisista tekoälyn sovellusmahdollisuuksista on asiakkaiden ongelmatilanteiden ennakointi ja niihin vastaaminen ennen ongelmien syntymistä. Tämä voi tapahtua esimerkiksi chatbotin kautta proaktiivisten viestien välityksellä, jossa asiakaspalvelukontakti lähtee yrityksen puolelta. IBM:n blogiteksti kertoo asiasta seuraavasti: ”Järjestelmät, joissa on sisäänrakennettua tekoälyä, voivat seurata lähes rajatonta määrää verkkosivuilla ja sovelluksien sisällä tapahtuvaa toimintaa erilaisten häiriöindikaattorien varalta, tunnistuen ongelmia kokevat asiakkaat ja mitkä heidän ongelmansa ovat. Järjestelmä voi vastata reaaliajassa tarjoten tukea usein kysytyjen kysymyksen vastausten tai virtuaalisten palveluagenttien muodossa eri järjestelmillä ja laitteilla. Kyvyllä ratkaista asiakaspalveluongelmia ennen niiden syntymistä on valtava potentiaali. Sillä voitaisiin merkittävästi madaltaa asiakkaiden luopumisastetta ostossyklissä ja vähentää negatiivisen asiakaspalautteen määrää sekä parantaa asiakkaiden tyytyväisyyttä.” (Schneider 2016.)

Tekoälyä hyödyntävän asiakaspalvelun voidaan olettaa myös laskevan asiakaspalveluun kohdistuvia kustannuksia. Tekoälypohjaisen asiakaspalvelun pystyessä käsittelemään yksinkertaiset, monotoniset ja rutiinitason suuren volyymin asiakaskysymykset, voidaan asiakaspalveluun tarvittavien ihmistyöntekijöiden määrää laskea kuten esimerkiksi aiemmin mainitun China Merchant Bankin tapauksessa tai vapauttaa ihmisasiakaspalvelijoiden resursseja entistä enemmän asiakkaiden monimutkaisempien ongelmien käsittelemiseen. Kustannuksissa voidaan odottaa myös säästöjä, kun otetaan huomioon säästöt ihmisasiakaspalvelijoiden rekrytointiin ja koulutukseen vaadittavien kustannusten osalta, verrattaessa kustannuksia tekoälypohjaisen asiakaspalvelijan kouluttamiseen, joka on periaatteessa kertaluontoista (Schneider 2016).

### **2.3 Suuret investoinnit – suuret ongelmat**

Tekoälyyn ja chatboteihin liittyvä ”kultakuume” on tällä hetkellä huipussaan. Eri-tyisesti teknologiajätit ovat ryhtyneet aiheiden osalta varsinaiseen kilpavarusteeluun. McKinsey Global Institutin kesällä 2017 julkaistussa Artificial intelligence the next digital frontier? -raportissa arvioidaan yritysten sijoittaneen vuoden 2016

aikana 26–39 miljardia dollaria tekoälyyn ja sen kehittämiseen. Lukemasta teknologiajättien investointien osuudeksi arvioidaan noin 20–30 miljardia. (McKinsey Global Institute 2017.)

Teknologiajätit ovat suurista panostuksista huolimatta kuitenkin kohdanneet myös suuria vaikeuksia kokeillessaan uusia teknologioita käytännössä ja joutuneet osin laskemaan omia tekoälyn liittyviä odotuksiaan. Monissa tapauksissa tekoälyteknologioihin ja niiden hyödyntämiseen käytännön sovelluksissa on suhtauduttu ylioptimistisesti erityisesti vaadittavan aikajänteen osalta, eikä vastaantulevia ongelmia ei ole kyetty arvioimaan ennalta tarpeeksi tarkasti.

Esimerkiksi Microsoftin Twitterissä vuonna 2016 julkaiseman Tay-botin tarkoituksena oli oppia keskustelemaan käyttäjien sille lähettämien viestien pohjalta. Tay kuitenkin oppi vuorokauden sisällä kiroilemaan ja lähettämään rasisisia sekä seksistisiä viestejä seuraajilleen, minkä jälkeen Microsoft joutui vetämään botin pois käytöstä. (Reese 2016.) Facebookin taas on raportoitu myöntäneen, että sen kehittäjille julkaistun avoimen rajapinnan avulla luodut chatbotit pystyvät vastaamaan ainoastaan noin 30 % asiakkaiden vaatimuksista ilman ihmisagentin vaikutusta (Fuscaldo 2017).

IBM:n Watson-chatbotin menestys Jeopardy-tietovisassa vuonna 2011 herätti suuria toiveita tekoälyn hyödyntämisestä eri käyttötarkoituksissa. IBM pyrki hyödyntämään Watsonin taustalla olevaa teknologiaa jopa lääketieteen alalla toimimalla lääkäreiden tukena tarjoten diagnooseja ja hoitosuosituksia syöpätapauksissa. Watson ei kuitenkaan ole pystynyt vastaamaan korkeisiin odotuksiin, ja esimerkiksi Houstonissa sijaitsevan M.D. Anderson Cancer Centerin ja IBM:n välinen yhteistyöprojekti keskeytettiin neljän vuoden jälkeen, kun työkalu ei ollut valmis vietäväksi pilottitestejä pidemmälle. Alun perin projektille asetetut 2,4 miljooan dollarin kustannukset olivat projektin aikana ehtineet kohota lopulta 39 miljoonaan. (Freedman 2017.)

Microsoftin tutkimusyksikön varapuheenjohtaja Peter Lee on myös kommentoinut teknologiajättien sisäisten huippuluokan koneoppimista hyödyntävien tekoälyjär-

jestelmien olevan edelleen liian joustamattomia ja kalliita, jotta niitä voitaisiin tarjota yritysten käyttöön, sillä teknologioita jouduttaisiin aina työläästi muokkamaan ja yksilöimään kullekin yksittäiselle yritykselle. Leen mielestä ohjelmistot, jotka kykenisivät oppimaan ja mukautumaan yksittäisen yrityksen tarpeisiin ja ehtoihin, eivät ole vielä ratkaisu ongelmiin. Hänen mukaansa Microsoftin omat koekielut saada ohjelmistot oppimaan reaali maailmasta ovat osoittaneet, että teknologiat tätä varten eivät ole vielä tarpeeksi kypsiä. Koneoppimista hyödyntävät järjestelmät ovatkin tulleet Leen mukaan esille ennemminkin epäonnistumisien kuin onnistumisien kautta. Hänen havaintojensa mukaan oppivien ohjelmistojen päästäminen vapaaksi tosielämän tilanteissa vaatii edelleen ihmisasiantuntijoiden tarkkaa seuranta. (Simonite 2017.)

Tekoälypohjaisten asiakaspalvelijoiden vastaukset perustuvat aina ainoastaan todennäköisyyksiin. Tästä syystä monet yritykset arastelevat asiakaspalvelun luottamista täysin tekoälyn vastuulle riskien välttämiseksi. Forbesin artikkeli esittääkin useiden yritysten ottaneen käyttöön menetelmän, jossa puhutaan tekoälyavusteisista ihmisasiakaspalvelijoista. Näissä tekoäly voi esimerkiksi analysoida asiakkaan esittämän keskustelun ja esittää arvion ongelman ratkaisusta ihmisasiakaspalvelijalle, joka tekee lopullisen päätöksen ratkaisun soveltuvuudesta asiakkaan ongelmaan. Tekoäly voi myös avustaa esimerkiksi asiakaspalveluun saapuvien puheluiden analysoinnissa ja reitityksessä oikeille ihmisasiakaspalvelijoille. (Zhou 2017.)

Tekoälyn ja sen hyödyntämisen mahdollisuudet sekä potentiaaliset edut ovat valtavat ja kiistattomat, mutta tekoälyyn liittyvään uutisointiin ja ”hypeen” on pyrittävä suhtautumaan varauksella ja kriittisesti. Tekoälyn kehitys mahdollistaa nykypäivänä jo teknologisesti paljon ja digitalisaation myötä kuluttajat ovat entistä valmiimpia vastaanottamaan ja kokeilemaan asiakaspalvelun uusia muotoja. Ongelmana on kuitenkin asiakaspalvelun osalta reaali maailman asiakaspalvelutilanteiden ja asiakkaiden ongelmien monimuotoisuus, joiden matemaattinen mallintaminen ja ratkaiseminen sovellusten avulla, on käytännössä osoittautunut haastavaksi jopa valtavilla resursseilla toimivien teknologiajättien

huippututkimukselle. Asiakaspalvelun kohdalla tekoälyn avulla kyetään realistisesti ratkomaan vasta rutiinitason toistuvia ongelmia ja tarve ihmisasiakaspalvelulle tulee säilymään vielä pitkään.

Liikenne- ja viestintäministeriön, Tekesin, Teknologiateollisuuden ja Verkkoteollisuuden julkaisema Digibarometri 2017, joka käsittelee erityisesti tekoälyä, pääsee pohdinnoissaan tekoälyyn liitetystä ”hypestä” kirjoittajan kanssa samoihin johtopäätöksiin ja ovat sovellettavissa myös tekoälyn hyödyntämiseen asiakaspalvelussa. Digibarometrin kirjoittajat eivät kiistä tekoälyn merkittävää potentiaalia, mutta pitävät nyky-”hypeä” ylimitoitettuna erityisesti tekoälyn yleistymisen aikajänteen, mutta mahdollisesti myös sovelluspotentiaalin osalta. Kirjoittajat päättävät pohdinnan seuraavasti: ”Ylipäätään teknologia määrittää vain periaatteellisten mahdollisuuksien rajoja – toteutus ja tulemat ovat ihmisten käsissä.” (Liikenne- ja viestintäministeriö ym. 2017.)

### **3 Chatbotit**

#### **3.1 Chatbotin määritelmä**

Chatbotit ovat yksi suosituimmista ja käytetyimmistä teknologioista, joiden kautta tekoälypohjaista asiakaspalvelua on viime vuosina toteutettu. Oxfordin yliopiston ylläpitämä sanakirja määrittelee chatbotin terminä seuraavasti: ”Chatbot on tietokoneohjelma, joka on suunniteltu simuloimaan keskustelua ihmiskäyttäjän kanssa erityisesti internetin välityksellä” (Oxford University Press 2017).

Webopedia sanakirja määrittelee chatbotit hieman kattavammin: ”Chatbot on lyhenne sanoista chat robot. Chatbotit ovat tietokoneohjelmia, jotka simuloivat ihmismäistä keskustelua tai chattia tekoälyn avulla. Tyypillisesti chatbotit kommunikoiivat ihmisen kanssa, mutta kehitteillä on sovelluksia, joissa kaksi chatbotia keskustelevat keskenään. Chatbotit ovat käytössä sovelluksissa kuten verkkokauppojen asiakaspalvelussa, puhelinkeskuksissa tai verkkopeleissä. Näihin tar-

koituksiin käytettyjen chatbotien keskustelut ovat tyypillisesti rajoitetut tiettyyn yksittäiseen tarkoitukseen, eikä niiden ole tarkoitus kattaa koko ihmisten välisten keskustelun kirjoa.” (Wepobedia 2017.)

TechTargetin määritelmä laajentaa vielä hieman määritelmää kuvaamalla chatboteja tekoälyn puhemiehenä tai edustajana. TechTarget myös kertoo chatbotien kommunikoinnin perustuvan joko teksti- tai ääniviestintään. Määritelmän mukaan chatbotit myös tyypillisesti hyödyntävät koneoppimista parantaakseen luonnollisen kielen käsittelyn ja puheentunnistuksen tarkkuutta. TechTargetin määritelmä jakaa chatbotit kahteen eri luokkaan, tilallisiin ja tilattomiin chatboteihin. Tilaton chatbot lähestyy jokaista vuorovaikutusta kuin se tapahtuisi uuden käyttäjän kanssa, kun taas tilalliset chatbotit ovat hienostuneempia ja pystyvät tarkastelemaan aiempia vuorovaikutuksia ja vastaamaan käyttäjälle kontekstin luoman kehityksen sisällä. (TechTarget 2016.)

### **3.2 Chatbotien historia**

Vaikka chatbotien suosio on viime vuosien aikana saavuttanut uuden huippunsa, eivät chatbotit itsessään ole kuitenkaan uusi teknologia. Niiden historia johtaa konseptina aina 1950-luvulle, jolloin Alan Turing kirjoitti artikkelin Computing machinery and intelligence, luoden samalla Turingin testin (Austin 2017). Turingin testissä koneen voidaan sanoa kykenevän ajatteluun, jos sen käyttäjälle antamat vastaukset on mahdotonta erottaa ihmisen antamista vastauksista. Testiä pidetään yhä edelleen yhtenä tekoälyn mittarina ja vuodesta 1990 asti toiminut tunnettu tekoälykilpailu Loebner Prize on ensimmäinen muodollinen instanssi Turingin testistä. (Loebner 2015.)

Ensimmäisenä chatbotina pidetään Joseph Weizenbaumin vuonna 1966 julkaisemaa ohjelmaa nimeltä ELIZA. ELIZA kykeni saavuttamaan illuusion älykkyydestä tunnistamalla käyttäjän syötteestä avainsanoja ja -fraaseja ja vastaamalla näiden mukaisesti käyttämällä ennalta määritettyjä skriptejä. Yksi sen skripteistä antoi ELIZAN esiintyä psykoterapeutin roolissa. Vuonna 1972 psykiatri Kenneth Colby kehitti chatbotin PARRY, joka tunnettiin myös nimellä ”ELIZA with attitude”.



PARRY kykeni ottamaan roolikseen paranoidisesta skitsofreniasta kärsivän potilaan. Psykiatrit eivät pystyneet erottamaan PARRYn kanssa käytyjä keskusteluja ihmispotilaiden kanssa käydyistä keskusteluista. (Austin 2017.)

Richard Wallacen vuonna 1995 kehitetty ALICE pohjasi pitkälti ELIZAssa käytettyihin syötteestä toistuvia malleja etsiviin tekniikoihin. ALICE kuitenkin pääsi lähemmäksi ihmismäisen keskustelun tasoa käyttämällä hyväkseen Artificial Intelligence Markup Language (AIML) -kieltä, joka mahdollisti laajemman valikoiman hienostuneempia vastauksia. (Austin 2017.)

Ensimmäinen tunnettu eräänlaista oppimista hyödyntävä chatbot on Rollo Carpenterin vuonna 1982 alkuun pantu ja vuonna 1997 internetissä julkaistu Jabberwacky. Jabberwacky ei perustanut vastauksiaan staattiseen tietokantaan aiempien chatbotien tapaan, vaan keräsi fraaseja keskusteluun osallistuneilta käyttäjiltä kasvattaen näin dynaamisesti omaa vastaustietokantaansa. Jabberwackyn uusin iteraatio käynnistettiin vuonna 2008, jolloin chatbot myös nimettiin uudestaan Cleverbotiksi. (Austin 2017.)

2000-luvun ensimmäisen vuosikymmenen aikana chatbotien älykkyys kasvoi entisestään. Aikaisempien chatbotien pohjautuessa mallien tunnistamiseen liittyviin tekniikoihin, aloitettiin 2000-luvun chatboteissa ottamaan käyttöön koneoppimiseen perustuvia ja muita kehittyneitä algoritmeja, mitkä mahdollistivat chatbotien sopeutumisen ja oppimisen. Näissä oppiminen pohjautuu käyttäjien kanssa käytyihin vuorovaikutuksiin. Edistyksestä huolimatta tekoälyn asiantuntijat eivät kuitenkaan vielä ole pystyneet kehittämään chatboteja, jotka mallintaisivat tarkasti yleistä ihmiskeskustelua. Chatbotit liikkuvat edelleen tietyn määritellyn tarkoituksen asettamissa rajoissa esimerkiksi virtuaalisten assistenttien muodossa. Virtuaaliassistenttien potentiaalinen käyttöarvo on kuitenkin herättänyt suurten teknologiajättien kuten IBM:n, Applen, Googlen, Amazonin, Microsoftin ja Facebookin kiinnostuksen ja saaneet nämä investoimaan chatbotien kehitykseen viime aikoina. (Austin 2017.)

Yksi 2000-luvun runsaasti julkisuutta saaneista chatboteista oli vuonna 2006 luotu IBM Watson, joka päihitti ihmiskilpailijat amerikkalaisessa Jeopardy-tietovisassa (Austin 2017). Jeopardy-tietovisassa ohjelman juontaja antaa kisailijoille vastauksen ja kilpailijat pyrkivät kertomaan vastaukseen liittyvän kysymyksen. Watson osallistui kilpailuun vuonna 2011, saaden vastaansa kaksi kisassa parhaiten menestynyttä ihmiskilpailijaa. Watsonin menestys kisassa perustui sen kykyyn ajaa satoja kielellisen analyysin algoritmeja samanaikaisesti ja kykyyn hakea tietoa yli 200 miljoonasta sivusta eri lähteistä koostuvaa informaatiota. Watson ei myöskään ollut kisan aikana yhdistettynä internettiin, vaan kaikki sen käytettävissä ollut tieto oli tallennettu paikalliseen tietokantaan. Watsonin oli myös pystyttävä käsittelemään tietoa sekunneissa, jotta se ehtisi painamaan summeria kisassa saaden ensimmäisen vastausvuoron. (Best 2017.)

### **3.3 Chatbotit nykypäivänä**

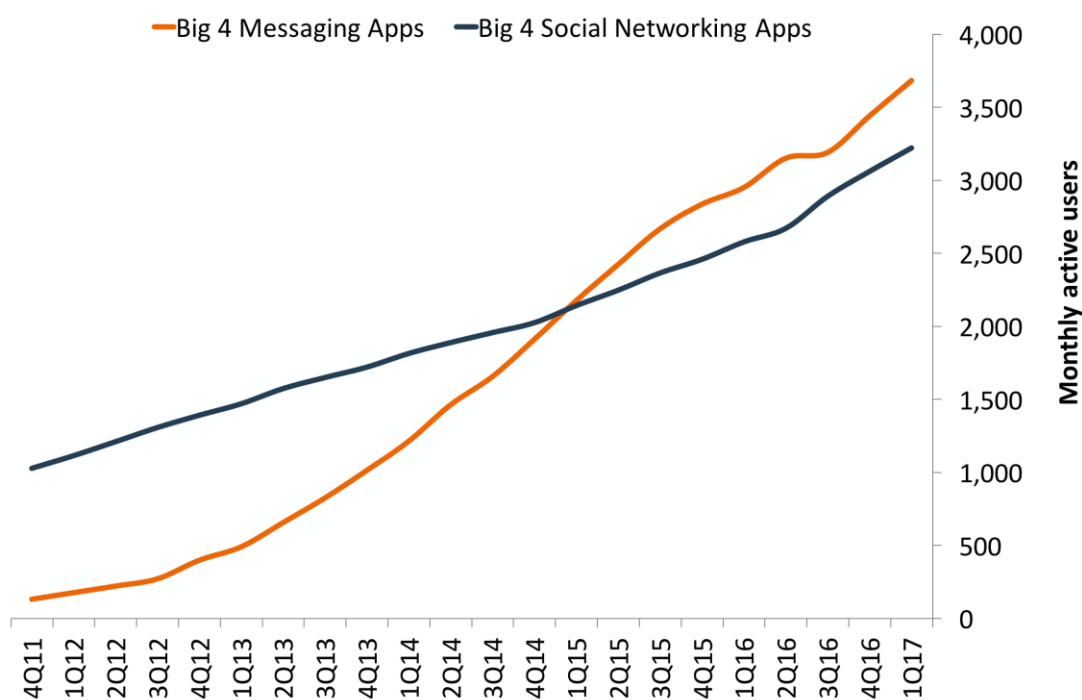
Viime vuosien aikana chatbotit ovatkin tulleet tutuiksi monille viimeistään mobiililaitteisiin sisällytettyjen virtuaaliassistenttien muodossa. Esimerkkejä näistä ovat muun muassa Applen Siri, Google Now, Amazonin Alexa, Microsoftin Cortana sekä Google Assistant. Nämä virtuaaliassistentit käyttävät hyväkseen luonnollisen kielen käsittelyä sekä puheentunnistusta ja yhdistävät web-palveluihin vastataksaan käyttäjien kysymyksiin ja täyttääkseen heidän tarpeensa. Viimeisimpänä markkinoille ovat ilmestyneet myös ubiikkeja käyttöliittymiä koteihin tuovat chatbotit kuten Google Home ja Amazon Echo. (Austin 2017.)

Accenturen raportti Chatbots in customer service vuodelta 2016, näkee chatbotien nykysuosion taustalla tekoälyn kehityksen ohella myös pikaviestintäpalveluiden leviämisen ja suosion kasvun (Accenture Interactive 2016). Pikaviestintäpalveluiden käyttäjämäärien kasvu on ollut räjähdysmäisen nopeaa mobiililaitteiden yleistyessä ihmisten arjessa. Niiden suosio on aktiivisten käyttäjämäärien osalta ohittanut jo aiemmin suositut yhteisöpalvelut.

Kuvio 1 esittää pikaviestimien kasvua suhteessa sosiaalisten verkostojen kasvuun. Kuviosta voidaan havaita neljän suurimman pikaviestintäpalvelun kuukausittaisien aktiivisten käyttäjien määrän lähestyvän 4 miljardin rajaa vuonna 2017. Huomionarvoista on myös, kuinka neljä suurinta pikaviestintäpalvelua ovat vuoden 2014 loppupuolella ohittaneet neljä suurinta yhteisöpalvelua käyttäjämäärissä ja kuinka niiden suosion kasvun arvioidaan jatkuvan aina opinnäytetyön kirjoittamisen hetkelle saakka.

## Messaging Apps Have Surpassed Social Networks

Global monthly active users for the top 4 messaging apps and social networks, In millions



Note: Big 4 messaging apps are WhatsApp, Messenger, WeChat, Viber.  
Big 4 social networks are Facebook, Instagram, Twitter, LinkedIn  
Source: Companies, Apptopia, TechCrunch, BI Intelligence estimates, 2017

BI INTELLIGENCE

Kuvio 1. Pikaviestintäpalveluiden aktiiviset käyttäjämäärät suhteessa yhteisöpalveluihin. (BI Intelligence 2016).

Pikaviestintäpalveluiden suosio on avannut yrityksille uuden kanavan, jonka kautta yritykset voivat olla vuorovaikutuksessa asiakkaiden kanssa. Accenturen raportti pitääkin pikaviestintäpalveluiden kutsumista ”viestintäpalveluiksi” niiden tarjoamien mahdollisuuksien vähättelynä. Nykyiset pikaviestintäpalvelut ovat kehittyneet sisältämään ominaisuuksia kuten maksaminen, tilaaminen ja varausten tekeminen,

mitkä aikaisemmin olisivat vaatineet kanavakseen erillisen mobiiliapplikaation tai verkkosivut. (Accenture Interactive 2016).

Chatbotit ovatkin nousseet yhdeksi ratkaisuksi ja niiden avulla yritykset voivat hyödyntää pikaviestintäpalveluiden suosion kasvua vuorovaikutuskanavana. Pikaviestimet ovatkin muodostuneet suosituimmaksi julkaisualustaksi chatboteille, kun aikaisemmin niiden julkaisualustoina toimivat usein perinteiset kanavat kuten verkkosivut, mobiiliapplikaatiot, tekstiviestipalvelut tai puhelinkeskukset.

Pikaviestimissä toimivien chatbotien suosion juuret johtavat kuitenkin länsimaisten pikaviestintäpalveluiden sijasta Kiinaan. Chatbots Magazinessa julkaistu artikkeli China, WeChat and the origins of chatbots kertoo kiinalaisen WeChat-pikaviestinalustan tarjonnan alustan boteille jo vuodesta 2013 alkaen (Chatbots Magazine 2017). Vuoden 2016 neljännellä kvartaalilla WeChatin päivittäisten käyttäjien määrä kohosi 768 miljoonaan ja arvioiden mukaan puolet käyttäjistä viettävät aikaa palvelun parissa päivittäin vähintään 90 minuutin verran (CIW Team 2016). Chatbotit ovat onnistuneet lyömään itsensä läpi kiinalaisessa kuluttajakulttuurissa ja Chatbots Magazinen artikkelin mukaan monilla kiinalaisilla yrityksillä on olemassa chatbot WeChat-alustalla, vaikka yritykseltä puuttuvat verkkosivut (Chatbots Magazine 2017).

Poikkeuksia lukuun ottamatta WeChat-alustalla julkaistut botit ovat vielä toimintoiltaan melko yksinkertaisia. Osat chatboteista pyrkivät jo hyödyntämään koneoppimisalgoritmeja tullakseen älykkäämmiksi ja oppiakseen nopeammin ymmärtämään käyttäjien syötteitä. Silti monien WeChat-alustalla toimivien chatbotien toiminta pohjaa edelleen avainsanojen tunnistamiseen tekstisyötteistä ja käsinkirjoitettuihin sääntöihin, joiden perusteella eri tilanteisiin valitaan esiasetetut vastausvaihtoehdot. Alustalla julkaistut chatbotit eivät vielä kykene antamaan monipuolisia vastauksia tarkemmin määriteltyihin kysymyksiin. Brändit tarjoavat WeChat-alustalla käyttäjilleen chatbotien kautta pääasiassa ainoastaan perustason asiakaspalvelua ja viihdettä. (Chatbots Magazine 2017.)

WeChat-pikaviestintäpalvelun chatbotien menestystä on pyritty toistamaan viime vuosien aikana myös läntisessä kulttuurissa ja länsimaissa suosittujen pikaviestintäpalveluiden kohdalla. Monet suositut pikaviestintäpalvelut kuten Facebook Messenger, Slack, Telegram, Kik ja Skype ovat avanneet omat rajapintansa chatbot-kehittäjien käyttöön. Suosittu pikaviestintäpalvelu WhatsApp on huhujen mukaan myös avaamassa chatbotit mahdollistavan rajapinnan, mutta opinnäytetyön kirjoittamisen hetkellä aiheelle ei löytynyt virallista ja luotettavaa vahvistusta.

Länsimaisten pikaviestintäpalveluiden chatbot rajapinnat ovat olleet erittäin suosittuja jo ensimmäisten vuosien aikana. Esimerkiksi VentureBeatin julkaiseman artikkelin mukaan Facebookin varapresidentti David Marcus raportoi huhtikuussa 2017 Facebookissa julkaistujen chatbotien määrän ylittäneen 100 000 rajan vain vuoden sisällä rajapinnan julkistamisesta. Marcuksen mukaan yritysten ja asiakkaiden välisten viestien määrä Messenger-pikaviestinalustalla on kuukausittain jo 2 miljoonaa. (Johnson 2017.)

Suomenkielisten ja suomalaisten yritysten chatbotien määrä on pysynyt vielä maltillisena. Facebookin Messenger alustalla chatboteja ovat kuitenkin julkaisseet muun muassa Autotie.fi ja Lassila & Tikanoja. Autotie.fi:n chatbot poistui käytöstä opinnäytetyöprosessin aikana, mutta opinnäytetyöprosessin alkupuolella chatbot mahdollisti myytävien autojen sekä huoltopisteiden haun. Lassila & Tikanojan chatbot Kikka tarjoaa kierrätysohjeita käyttäjän syöttämälle objektille ja avustaa käyttäjää löytämään lähimmän kierrätyspisteen.

Kotipizzan yhteistyössä OP Ryhmän kanssa toteutettu ja Facebookin Messengerissä kesäkuussa 2017 julkaistu KotiBotti on todennäköisesti Suomen ensimmäinen pikaviestintävälitteinen maksusovellus. Sen avulla asiakas voi tilata Kotipizzan tuotteita ja maksaa tilaukset mobiilisti. (Lehtiniitty 2017.) Kirjoittajan testaamisen perusteella Kotipizzan chatbot hyödyntää käyttöliittymänsä osalta nykypäivän chatbotien mahdollistamien valikoiden ja painikkeiden käyttöä, jolloin asiakkaan ei tarvitse kirjoittaa vuorovaikutuksen aikana lainkaan tekstiä.

Myös Finnair liittyi chatboteja pikaviestimessä tarjoavien suomalaisten yritysten joukkoon syyskuussa 2017. Finnairin chatbot tarjoaa asiakkailleen erilaisia lento- matkustamiseen liittyviä tukipalveluja ja mahdollistaa esimerkiksi lentojen varaamisen. Se osaa myös tarvittaessa ohjata keskustelun ihmisasiakaspalvelijalle. Finnairin chatbot toimii alkuvaiheessa ainoastaan englanniksi, mutta tarkoituksena on myös mahdollistaa käyttö lähiaikoina suomeksi ja kiinaksi. Finnair suunnittelee myös julkaisevansa chatbotin alkuvaiheen jälkeen WeChat-alustalla. (Tekniikka ja talous 2017.)

Nykypäivän chatbotien kirjo on opinnäytetyöprosessin aikana esittäytynyt kirjoittajalle erittäin laajana ja monipuolisena. Asiakaspalveluun keskittyvien chatbotien toteutustavoissa havaittiin niihin tutustumisen aikana runsaasti eroja. Osat chatboteista perustavat toiminnallisuutensa esimerkiksi täysin valikoiden ja painikkeiden hyödyntämiselle, jolloin niiden käyttö ei eroa merkittävästi mobiilisovellusten tai verkkosivujen toimintaperiaatteista muutoin kuin julkaisukanavan osalta. Syötteiden käsittelyssä osat chatboteista hyödyntävät erilaisia luonnollisen kielen käsittelyyn ja puheentunnistukseen kehitettyjä tekoälyteknologioita, kun taas osat chatboteista käsittelevät syötteitä esimerkiksi vertailemalla syötteen avainsanoja ennalta määritettyihin malleihin ja sääntöihin, joiden perusteella chatbot valitsee tilanteeseen soveltuvan vastausvaihtoehdon.

Innovatiivisimmat chatbotit pyrkivät hyödyntämään jo muitakin koneoppimiseen tai muihin tekoälyalgoritmeihin perustuvia tekniikoita kuin luonnollisen kielen käsittelyn mahdollistavia algoritmeja esimerkiksi chatbotin vastausta varten tehtävässä tiedonhaussa tai kuvantunnistuksessa. Näiden ominaisuuksien tunnistaminen on kuitenkin käyttäjän näkökulmasta usein hankalaa. Chatbotien kohdalla myös tekoälyyn sekä niiden oppimiseen liittyvien termien käyttö on vaihtelevaa ja niitä vaikutetaan käyttävän tällä hetkellä ilman vakiintuneita käytänteitä ja määritelmiä. Digibarometri 2017 -raportin laatijat ovat myös ottaneet ongelman huomioon tekstissään ja kertovat asiasta seuraavasti: ”Tekoälyn erottaminen mistä tahansa softasta on vaikeaa. Vaikeutta lisää se, että hypen myötä vanhan ja asiaan liittymättömänkin tekemisen kylkeen liimataan mieluusti tekoälytarra seksikkyyttä lisäämään.” (Liikenne- ja viestintäministeriö ym. 2017.)

## 4 Luonnollisen kielen käsittely

Luonnollisen kielen käsittely (engl. Natural language processing, NLP) lasketaan yhdeksi tietojenkäsittelytieteiden, kielitieteiden ja tekoälyn osa-alueista, jonka tarkoituksena on mahdollistaa tietokoneohjelmille kyky ymmärtää ihmisten käyttämää luonnollista kieltä (Chopra, Prashar & Sain 2013). Luonnollisen kielen käsittelyä pidetään yhtenä tietojenkäsittelytieteiden ja tekoälyn haastavimmista ongelmista. Aiheen parissa tehdyn tutkimuksen historia voidaankin johtaa kuten chatbotienkin tapauksessa aina 1950-luvulle asti (Chopra ym. 2013). Kun chatbotit määritellään sovelluksiksi, jotka simuloivat keskustelua käyttäjän kanssa, voidaan niiden kyky ymmärtää ja käsitellä ihmisen tuottamaa luonnollista kieltä nähdä yhtenä olennaisena chatboteihin liitettävänä elementtinä.

Luonnollisen kielen käsittely voidaan jakaa kahteen pääkomponenttiin, jotka ovat luonnollisen kielen ymmärtäminen (engl. Natural language understanding, NLU) ja luonnollisen kielen generointi (engl. Natural language generating, NLG). Luonnollisen kielen ymmärtämisellä pyritään analysoimaan kielen eri muotoja ja muuntamaan luonnollisella kielellä vastaanotetut syötteet esitysmuotoon, jota tietokoneet voivat hyödyntää. Luonnollisen kielen generoinnilla pyritään tuottamaan merkityksellisiä lauseita ja fraaseja luonnollisella kielellä, jonkin tietokoneen sisäisen esitystavan pohjalta. (Tutorialspoint 2017.)

Luonnollisen kielen käsittelyn toteutukseen sisältyy tyypillisesti useampia eri ali-tehtäviä ja vaiheita kuten morfologinen ja leksikaalinen analyysi, syntetaattinen analyysi, semanttinen analyysi, diskurssi-integraatio ja pragmaattinen analyysi (Chopra ym. 2013). Näiden eri vaiheiden ratkaisuksi on kehitetty erilaisia tekniikoita, joita ovat esimerkiksi sanaluokkien analyysi (engl. Part-of-speech analysis) tai sanaluokkien ”taggaus” (engl. Part-of-speech tagging, POS-tagging), nimettyjen entiteettien tunnistaminen (engl. Named entity recognition, NER) sekä syntaattinen ja semanttinen parsiminen (engl. Syntactic and semantic parsing). Tyypillisesti vähintään yksi tai useampi näistä tekniikoista on sisällytettyinä luonnollisen kielen käsittelyä hyödyntäviin dialogijärjestelmiin. (Pascual-Nieto & Perez-Marin 2011.)

Luonnollisen kielen käsittelyssä ongelmia tuottavat yleisesti ihmisten puhuman kielen monitulkintaisuus. Monitulkintaisuutta esiintyy niin leksikaalisella kuin syntaksisella tasolla sekä myös pronominiin yhteyksissä esiintyvien viittausten kohdalla (Tutorialspoint 2017). Ongelmallisia luonnollisen kielen käsittelylle ovat myös erilaiset puhekielen ilmaisut ja alueelliset murteet (Gonsalves & Shah 2017).

Euroopan komission rahoittaman META-NET-huippuosaamisverkoston Suomen kieli digitaalisella aikakaudella -raportissa vuodelta 2012 listataan suomen kielen erityispiirteiksi rikas taivutusjärjestelmä, suhteellisen vapaa sanajärjestys, astevaihtelut, vokaaliharmonia, yhdyssanojen muodostaminen sekä morfologian kompleksisuus. Raportissa suomen kielen kieliteknologisten sovellusten saataavuus ja laatu arvioidaan hyväksyttäväksi ainoastaan perussovellusten ja -resurssien osalta. Raportti myös esittää huolen Suomen jäämisestä jälkeen kieliteknologisten resurssien kehittämisessä, mikäli aihealueen tutkimusta ja siihen käytettyä rahoitusta ei kasvateta. (Arppe, Bartis, Carlson, Hyvärinen, Koskeniemi, Lennes, Lindén, Nuolijärvi, Piehl, Vainio, Westerlund 2012.)

Luonnollisen kielen käsittelyä on pyritty aikojen saatossa toteuttamaan hyödyntämällä erilaisia tilastollisia menetelmiä sekä koneoppimismalleja (Chopra ym. 2013). Viime aikoina suorituskykyä on edelleen kyetty parantamaan erilaisilla syväoppimiseen perustuvilla lähestymistavoilla kuten neuroverkko-algoritmeilla. (The Stanford NLP Group 2017).

Luonnollisen kielen käsittelyn mahdollistavien algoritmien kehittäminen vaatisi kehittäjältä syvällistä tuntemusta kohdekielestä ja sen rakenteista sekä erilaisiin luonnollisen kielen käsittelyyn tarkoitettuihin monimutkaisiin algoritmeihin perehtymistä. Viime aikoina kehittäjien saataville ovat tulleet monet luonnollisen kielen käsittelyn lisäämisen mahdollistavat palvelut, jotka ovat rakennettu monimutkaisten algoritmien päälle. Näiden avulla kehittäjät voivat luoda luonnollisen kielen käsittelyä mahdollistavia malleja, tuntematta tarkalleen kuinka kielen käsittelyn yksityiskohdat ovat taustalla toteutettu. Tällaisia palveluita ovat esimerkiksi Microsoftin LUIS, Facebookin Wit.ai, IBM:n Watson sekä Googlen opinnäytetyöprosessin vielä alkaessa Api.ai nimellä tunnettu Dialogflow.



## 5 Työn tausta ja tavoitteet

### 5.1 Prototyypin tavoitteet

Opinnäytetyön toiminnallisessa osiossa tavoitteena oli toteuttaa tuotehaut Shopify-verkkokauppa-alustalle luonnollisella kielellä mahdollistava chatbot. Ensisijaisesti toteutuksen aikana tutkittiin, onko prototyyppi mahdollista toteuttaa työhön valittujen työkalujen avulla. Chatbotin ja sen dialoginhallinnan toteutukseen valittiin työkaluksi Microsoftin Bot Frameworkin sisältämä Bot Builder SDK for Node.js. Chatbot-sovelluksen ohjelmakoodi kirjoitetaan JavaScript-ohjelmointikielellä. Luonnollisen kielen käsittely toteutetaan englannin kielen osalta Microsoftin LUIS-palvelulla ja suomen kielen kohdalla Wit.ai-palvelua hyödyntäen.

Alustavina toiminnallisuuksina prototyypille määriteltiin tuotehakujen tekeminen 1–2 eri tuotekategorian tuotteille 3–4 hakukriteerin perusteella. Hakujen tekeminen mahdollistettaisiin käyttäjälle luonnollisella kielellä. Englanninkieliset syötteet tarkistettaisiin lisäksi oikeinkirjoituksen varalta käyttäen hyväksi Microsoftin Bing Spell Check Apia.

Suunnittelun alkuvaiheessa esille tuli kysymys luonnollisen kielen käsittelyn toteuttamisesta suomen kielellä. Prototyypin chatbotin ymmärtämälle kielelle ei esitetty aiheen ehdotuksen aikana tiukkaa vaatimusta. Opinnäytetyön kirjoittaja teki lyhyen tutkimuksen mahdollisuuksista lisätä chatbotiin kyky käsitellä suomen kieltä. Wit.ai-palvelu tarjosi suomen kielen luonnollisen kielen käsittelyä beta-version muodossa ja sen toiminta vaikutti kokeilemisen arvoiselta, vaikka ennakkotesteissä havaittiinkin mahdollisia ongelmia. Toimeksiantajan hyväksynnällä suomenkielinen toteutus sisällytettiin käytännön työn sisältöön ja suomen kielen sekä englannin kielen luonnollisen kielen käsittelyn lisäämisen erojen vertailu valittujen työkalujen osalta nostettiin yhdeksi opinnäytetyön tutkimuskysymykseksi.

Prototyypitoteutukselle ei opinnäytetyöprosessin aikana ennätetty hankkia olemassa olevaa Shopify-verkkokauppaa johon tuotehaut kohdistettaisiin. Testiverkkokaupan luominen sisällytettiin käytännön työn sisältöön. Testiverkkokaupan

päätettiin myyvän kameroita sekä mahdollisesti kameroihin ja kuvaamiseen liittyviä oheistarvikkeita toimeksiantajalta saadun esimerkki-idean pohjalta. Testiverkkokaupan tuotteiden määrä pyrittiin pitämään prototyypitoteutusta varten tarkoituksenmukaisena, siten että käyttäjän on mahdollista löytää muutamia eri tuotteita hakukriteerejä muuttamalla.

Opinnäytetyön kirjoittaja toivoi myös sisällyttävänsä käytännön työn oheen prototyypin testauttamisen muuttamalla testihenkilöllä, jonka kautta saataisiin alustavia tuloksia luonnollisen kielen käsittelyn ja hakutoiminnon onnistumisesta. Opinnäytetyön toimeksiantaja ja ohjaaja eivät nähneet testauksen olevan täysin välttämätöntä, mutta se otettiin mukaan suunnitteluvaiheeseen. Lopulta testauksista ei kuitenkaan ehditty toteuttaa tiukan aikataulun vuoksi.

## **5.2 Prototyypin rajaukset**

Prototyypinä toteutettavan chatbotin julkaisukanava oli yksi aiheen valinnan aikana esiintyneistä kysymyksistä. Mahdollisuuksina olivat esimerkiksi pikaviestimet, mobiiliapplikaatiot tai verkkosivut. Lopulta chatbot kuitenkin päädyttiin toteuttamaan, siten että sen toimintaa voi testata Microsoftin Bot Frameworkin emulaattorissa.

Chatbotin toiminnallisuuteen olisi voitu lisätä myös mahdollisuus puheentunnistukselle. Toiminnallisuus päätettiin rajata pois prototyypitoteutuksesta, sillä puheentunnistuksen mahdollistavat ominaisuudet olisi mahdollista lisätä chatbotiin jälkikäteen ja tämän ei nähty olevan oleellista opinnäytetyön aikana toteutettavan prototyypin kohdalla.

Opinnäytetyön kirjoittaja sisällytti alustavaan ehdotukseen prototyypin aiheesta toiminnallisuuden, jossa englanninkielisten syötteiden teksti olisi analysoitu niiden sisältämän positiivisen tai negatiivisen sävyn osalta. Toiminnallisuus olisi hyödyntänyt Microsoft Azuren Text Analytics API-rajapintaa. Tarkoituksena olisi kokeileva varautuminen keskustelun siirtämiseen ihmisasiakaspalvelijalle, mikäli käyttäjän syöttämät lauseet olisivat olleet sävyltään negatiivisia tai osoittaneet

merkkejä turhautumisesta. Vastaavan kaltaista toiminnallisuutta on hyödynnetty jo joissakin asiakaspalveluun suuntautuneissa chatboteissa. Lisäksi teknologiaa käytetään arvioitaessa suuria määriä asiakaspalautetta tai ihmisten mielipiteitä tuotteista, henkilöistä ja ilmiöistä esimerkiksi sosiaalisen median kanavilla. Toiminnon ei kuitenkaan nähty olevan vielä oleellinen prototyyppitoteutuksessa ja se jätettiin pois suunnitelmista.

## **6 Asiakaspalvelubotin prototyypin toteutus**

### **6.1 Toteutukseen valitut työkalut ja alustat**

Opinnäytetyön kirjoittaja havaitsi jo alkuvaiheessa runsaasti erilaisia vaihtoehtoja chatbotien ja niiden sisältämän tekoälyn toteutukselle. Mahdollisuudet vaihtelivat lähes kaiken kehittämisestä alusta alkaen itse aina erilaisten valmiiden työkalujen hyödyntämiseen. Prototyypissä toteutettavan chatbotin ja sen sisältämän tekoälyn kehityksessä käytetyille työkaluille ei toimeksiantajan puolelta esitetty rajoituksia, mutta valmiiden työkalujen hyödyntäminen oli suositeltavaa.

Chabottien kehitystyökalujen ja erilaisten tekoäly-palveluiden määrä on viime vuosien suosion aikana kasvanut nopeasti. Jo pelkästään chatbotien toteutukseen tarkoitettuja työkaluja on saatavilla valtava määrä. Osat työkaluista mahdollistavat chatbotien kehittämisen tiettyä julkaisualustaa ja niiden sisältämiä erityisominaisuuksia varten, kun taas osat työkaluista mahdollistavat yhden toteutuksen julkaisemisen usealla eri alustalla. Toteutustavat vaihtelevat graafisista drag- and drop -käyttöliittymistä täysin tekstimuotoiseen ohjelmointiin pohjaavaan kehitykseen, kun taas osat työkaluista tarjoavat näiden kahden seka- muotoa.

Joissain alustoissa ja työkaluissa on valmiina mahdollisuudet sisällyttää chatbotiin tekoälyominaisuuksia. Yleisimmin kyseessä on luonnollisen kielen käsittelyä mahdollistavaa toiminnallisuutta. Suuret palveluntarjoajat kuten esim. Microsoft,

IBM ja Google tarjoavat älykkäiden chabottien kehittäjille kattavat työkalut ja rajapinnat niin chatbotien kehittämiseen ja erilaisten tekoälypalveluiden lisäämiseen kuin myös mahdollisuudet chatbotien ”hostaamiseen” pilvialustoilla ja niiden julkaisemisen useilla eri julkaisualustoilla.

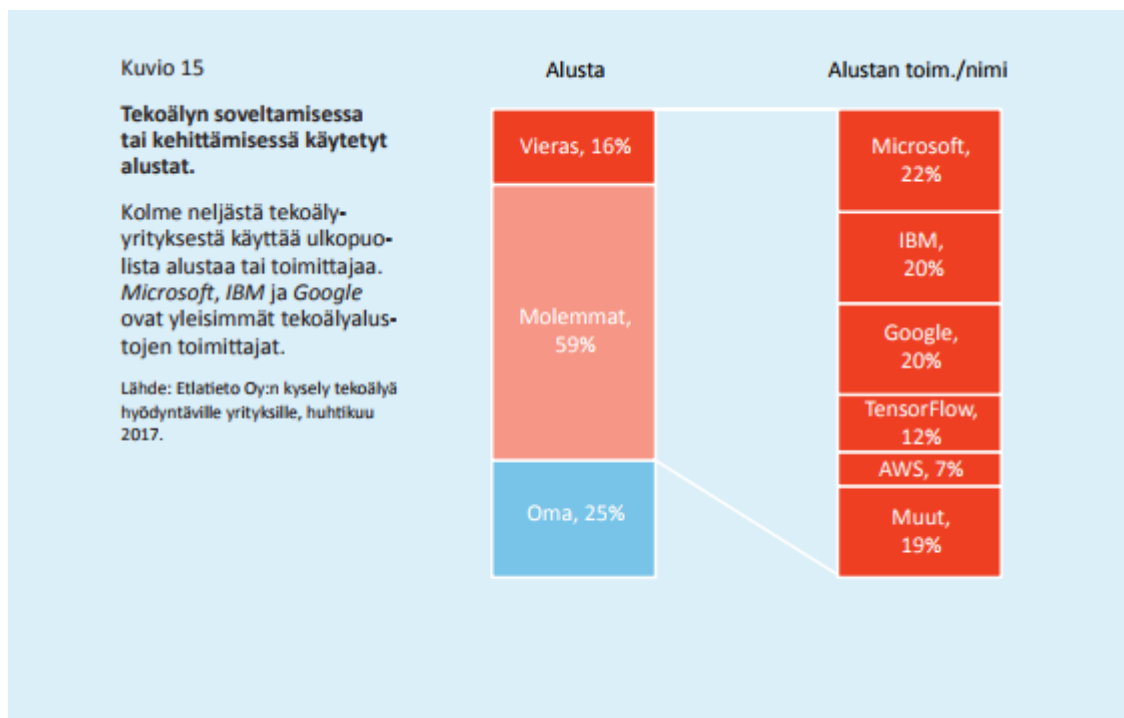
Opinnäytetyön kirjoittaja tutustui ja vertaili useampia eri työkaluja aiheen valinnan aikana. Työkalujen runsaan määrän vuoksi kattava vertailu oli aikataulun puitteissa mahdotonta. Lisäksi ongelmia lisää työkalujen ominaisuuksien osalta tiheään tahtiin viime aikoina tapahtuneet muutokset, joiden vuoksi työkaluja ja niiden eroja käsittelevä materiaali saattaa menettää ajankohtaisuutensa kuukausienkin sisällä. Esimerkiksi opinnäytetyöprosessin aikana Wit.ai jätti työkalustaan pois mahdollisuuden rakentaa chatboteja ja keskittyy nyt ainoastaan luonnollisen kielen ymmärtämisen toteutukseen. Lisäksi esimerkiksi opinnäytetyöprosessin loppupuolella Googlen aiemmin API.ai:na tunnettu palvelu muutti nimensä Dialogflowksi. Monet työkalut myös ilmoittavat olevansa edelleen joko preview- tai beta-tilassa.

Parhaiten tarkoitukseensa sopivien työkalujen valintaa olisi myös helpottanut aiempi käytännön kokemus chatbotien ja luonnollisen kielen ymmärtämisen toteutuksesta eri työkaluilla, jolloin valinnat olisivat voineet pohjata aiempaan tietoon. Työkalujen valintaprosessin haasteet olivat myös aiheena ennen opinnäytetyötä kirjoitetussa harjoitusaineessa.

Mindbrowserin yhteistyössä ChatbotsJournal.com:in kanssa laatimassa Chatbot Survey 2017 -tutkimuksessa selvitetään chatbotien nykyistä tilaa sekä niiden tulevaisuutta. Tutkimus toteutettiin kyselyiden ja haastattelujen avulla ja siihen osallistui yli kolmesataa eri teollisuuden alojen edustajaa. Tutkimus sisältää osion, joka käsittelee kyselyyn osallistuneiden ensisijaista valintaa chatbotien rakennusalueksi. Viiden kärki koostui seuraavista alustoista: IBM Watson, Wit.ai, Microsoft Bot Framework, Chatfuel ja Pandorabots. IBM Watson oli ensisijainen valinta 61 % kyselyyn osallistuneista yrityksistä, Wit.ai keräsi ääniä 45 % ja Microsoftin Bot Framework 41 %. Lisäksi tutkimukseen osallistuneista yrityksistä 76 % uskoo chatbotien rakentamiseen tarkoitettujen resurssien olevan helposti saatavilla. (Mindbrowser 2017.)

Prototyypin toteutukseen valittujen Microsoftin Bot Frameworkin sekä Wit.ai:n kuuluminen Mindbrowserin tuloksissa viiden suosituimman chatbot-rakennusalustan joukkoon tukee kirjoittajan työkaluvalintoja. Tutkimuksen tuloksia tarkasteltaessa on huomiotava, että Wit.ai ei kuitenkaan enää mahdollista itsessään chatbotien rakentamista, vaan ainoastaan näiden taustalle luotavan luonnollisen kielen ymmärtämisen toteutuksen, joten tämän osalta tutkimuksen tulokset ovat jo osin ehtineet vanhentua lyhyessä ajassa.

Digibarometri 2017 -raportissa on julkaistu Etlatieto Oy:n suomalaisille tekoälyyrityksille tehdyn kyselyn tulokset. Kyselyyn vastanneista 94 yrityksestä kolme neljäsosaa kertoi käyttävänsä tekoälyn soveltamisessa tai kehittämisessä ulkopuolista alustaa tai toimittajaa. Eniten käytetyin ulkopuolisen alustan tarjoaja oli Microsoft 22 %:in osuudella. (Liikenne- ja viestintäministeriö ym. 2017.) Kuvio 2 esittää Etlatieto Oy:n kyselyn tulokset tekoälyn soveltamiseen ja kehittämiseen käytetyistä alustoista suomalaisissa tekoäly-yrityksissä.



Kuvio 2. Tekoälyn soveltamisessa tai kehittämisessä käytetyt alustat suomalaisissa tekoäly-yrityksissä. (Liikenne- ja viestintäministeriö ym. 2017).

Etlatieto Oy:n toteuttaman kyselyn tulokset tulivat kirjoittajan tietoon vasta syksyllä 2017, kun työkalut prototyypin kehitykseen olivat jo valitut. Tulokset kuitenkin tukevat kirjoittajan työkalujen valintoja, joissa chatbot ja englannin kielen luonnollisen kielen käsittely toteutetaan Microsoftin tarjoamilla alustoilla ja palveluilla.

Vaihtoehtoisia työkaluja olisi ollut useampia. Kirjoittajaa erityisesti kiinnostaneita työkaluja olisivat olleet usein työkaluja esitelleissä ja vertailevissa materiaaleissa esiintyneet IBM:n Watson Conversation Service, Googlen DialogFlow sekä Recast.ai. Näistä kirjoittaja ehti hyvin kevyesti testata DialogFlow'ta, joka vaikutti lyhyen kokemuksen perusteella mahdollistavan samat ominaisuudet kuin prototyypin toteutukseen valitut työkalut. Watsonin Conversation Serviceen ja Recast.ai:hin ehdittiin tutustua ainoastaan niiden verkkosivujen, dokumentaatioiden ja videotutoriaalien muodossa. Näiden perusteella työkalut olisivat myös voineet soveltua prototyypin toteutukseen. Esimerkiksi Watsonin Conversation Service voisi hyvin korvata Microsoftin Bot Frameworkin sekä LUIS-palvelun. Watson Conversation Servicessä chatbotien toteutuksessa on vahvemmin esillä graafinen käyttöliittymä, kun taas Bot Frameworkissa toteutus on täysin ohjelmakoodiin perustuvaa.

Kirjoittaja ei kuitenkaan löytänyt ennakkoselvitysten aikana materiaalia, joka olisi selkeästi perustellut yhden työkalun paremmuutta suhteessa valittuihin työkaluihin ja yksityiskohtaisempi arviointi vaatisi laajempaa omakohtaista kokemusta työkalujen käytöstä. Kirjoittaja aloitti kuitenkin käytännön kokemuksen hankkimisen Microsoftin Bot Frameworkilla, jonka perusteella se myös osoittautui soveltuvan prototyypin toteutukseen. Wit.ai oli löytyneen materiaalin perusteella suuria ja suosituista työkaluista ainoa, joka mahdollisti suomenkielisen luonnollisen kielen ymmärtämisen. Wit.ai on myös täysin ilmainen, mikä tukee sen valintaa, sillä kirjoittaja pyrki valitsemaan prototyypin toteutukseen työkaluja, joista ei opinäytetyön aikana aiheutuisi kustannuksia.

Työkalujen valinnassa on pyritty myös pitämään mielessä mahdollinen jatkokehitys ottamalla huomioon mm. chatbotin mahdollisten julkaisukanavien määrä, mahdollisuudet lisätä chatbotin ominaisuuksia ja taustalla toimivaa tekoälyä,

chatbotin kehitysmenetelmät ja ohjelmointikielet, tulevat kustannukset sekä dokumentaation kattavuus. Valitut työkalut tulivat hyväksytyiksi myös toimeksiantajan puolelta ennen opinnäytetyön toteutuksen alkamista.

### 6.1.1 Microsoft Bot Framework

Bot Framework on Microsoftin marraskuussa 2016 julkaisema alusta chatbotien rakentamiseen ja julkaisemiseen eri kanavilla (Microsoft 2017a). Microsoft itse ilmoittaa ohjelmistokehyksen mahdollistavan tehokkaiden ja älykkäiden chatbotien rakentamisen, yhdistämisen, testauksen sekä käyttöönoton. Bot Framework koostuu useista komponenteista, joista kolme pääkomponenttia ovat chatbotien rakentamiseen tarkoitettu Bot Builder SDK, chatbotien rekisteröinnistä ja hallinnasta vastaava Bot Framework Portal sekä Channels, joka mahdollistaa chatbotien julkaisemisen useilla eri julkaisukanavilla. (Microsoft 2017b.)

Chatbotien rakentaminen on kirjoittamisen hetkellä mahdollista .Net ja Node.js SDK:iden kautta C#- ja JavaScript-ohjelmointikielillä (Microsoft 2017c). Lisäksi mahdollisuutena on vaihtaa viestejä Bot Framework Portalissa konfiguroitujen julkaisukanavien kanssa HTTPS-protokollan yli käyttämällä REST- ja JSON-standardeja (Microsoft 2017d).

Bot Builder SDK tarjoaa dialoginhallintaan oman vesiputousmaisena toteutustavan, jossa chatbotin kanssa käytävä keskustelu voidaan jakaa sarjoiksi erilaisia tehtäviä. Bot Builder SDK käsittelee tehtävät taulukkona funktioita, joissa ensimmäisen funktion tulokset voidaan viedä syötteenä seuraavalle funktiolle. (Microsoft 2017e.)

Toisena Bot Builder SDK:n hyödyllisenä toimintona dialoginhallintaan on mahdollisuus tallettaa käyttäjään, keskusteluun tai yhteen käyttäjään tietyssä keskustelun kontekstissa liitettävää dataa erilaisiin tietorakenteisiin. Talletetut tiedot on mahdollistaa säilyttää pidempiaikaisesti käyttämällä Azure-pilvipalvelualustan

Azure Table Storagea, Azure DocumentDB:tä tai luomalla oma räätälöity toteutus, jonka avulla tiedot on mahdollista tallettaa omaan tietokantaan. (Microsoft 2017f).

Microsoftin oma pilvipalvelualusta Azure tarjoaa edelleen lisämahdollisuuksia chatbotien kehittäjälle. Azurella on chatboteja varten oma palvelunsa Azure Bot Service, jonka kautta chatbot on mahdollista ottaa käyttöön pilvessä. Pilvipalvelu pitää huolen chatbotin skaalautumisesta käyttöasteen mukaan. Azure Bot Service sisältää myös analytiikkatyökalut chatbotin toiminnan seurantaan. Lisäksi Azuren palveluvalikoimaan kuuluvat esimerkiksi Azure Machine Learning Studio, jonka avulla kehittäjä voi luoda omia koneoppimisalgoritmeja sekä Azure Cognitive Services, joka koostuu kokoelmasta erilaisia tekoälyrajapintoja, jotka on helppo integroida Microsoft Bot Framework:in avulla luotujen chatbotien toiminnallisuuteen. Näistä tekoälyrajapinnoista erityisesti luonnollisen kielen käsittelyn mahdollistavan LUISin (Language Understanding Intelligent Service) integrointi Bot Frameworkilla luotuihin chatboteihin on dokumentoitu kattavasti.

Vaikka Bot Framework ja Azure Bot Service ovat tällä hetkellä ilmaisia, ovat osat Azuren palveluista maksullisia ja kustannukset määräytyvät käyttöasteen perusteella. Bot Frameworkin kohdalla täytyy lisäksi huomioida, että se on julkaisusta lähtien ollut preview-tilassa, mikä hankaloittaa sen tarjoamien työkalujen tulevaisuuden arviointia.

### **6.1.2 LUIS**

LUIS on lyhenne sanoille Language Understanding Intelligent Service. Se on Microsoftin julkaisema palvelu, joka antaa kehittäjien luoda älykkäitä sovelluksia, jotka voivat ymmärtää ihmiskieltä ja toimia käyttäjän pyyntöjen mukaisesti. LUIS käyttää hyväkseen koneoppimista ratkaistakseen vaikeana ongelmana pidettyä käyttäjän aikeen erottelua luonnollisen kielen syötteestä. (Microsoft Azure 2017a.)



LUIS-palvelu perustuu kolmeen pääkonseptiin, jotka ovat lausahdukset (engl. utterances), aikeet (engl. intents) sekä entiteetit (engl. entities) (Microsoft Azure 2017a). Näiden merkitystä tarkastellaan tarkemmin opinnäytetyön luvussa 6.2.8.

LUIS myös tarjoaa kehittäjän työn helpottamiseksi useita erilaisia yleisille aihealueille tarkoitettuja esirakennettuja malleja, jotka taas koostuvat esimallinnetuista aikeista ja entiteeteistä. Aihealueita ovat esimerkiksi kalenteri, viihde ja fitness. (Microsoft Azure 2017c.)

LUIS-palvelu on opinnäytetyön kirjoittamisen hetkellä edelleen preview-tilassa ja ilmainen kuukausittain ensimmäisten 10 000 transaktion osalta. 10 000 transaktion rajan ylittyessä palvelun käytöstä laskutetaan 1000 transaktion joukoissa. (Microsoft Azure 2017d.)

Intento on yhdysvaltalainen yritys, joka keskittyy tekoälyrajapintojen suorituskykyjen mittaamiseen ja niiden käytön yhdistämiseen yhden rajapinnan taakse (Intento 2017a). Heidän elokuussa 2017 tehdyssä luonnollisen kielen ymmärtämistä tarjoavien palvelujen suorituskykyä mittaavassa tutkimuksessa testattavana olivat Api.ai, Wit.ai, IBM Watson Conversation, Microsoft LUIS, Amazon Lex, Recast.ai ja SNIPS. Tutkimuksen mukaan Api.ai, Microsoft LUIS ja IBM Watson nousivat kolmen parhaan joukkoon käyttäjän aikeiden tunnistamisessa, nopeudessa ja mahdollisten kielten määrässä. Näistä palveluista LUIS suoriutui parhaiten erityisesti vastausten nopeudessa. (Intento 2017b.)

### **6.1.3 Wit.ai**

Wit.ai oli alun perin Y Combinator-nimisen startup-yrityksen rajapinta ääniaktivoiden liittymien rakentamiseen. Wit.ai siirtyi Facebookin omistukseen tammi-kuussa 2015. (TechCrunch 2015.) Sen ominaisuuksiin sisältyi noin vuoden ajan chatbotien rakentamisen mahdollistanut Bot Engine-työkalu. Palvelu kuitenkin suljettiin heinäkuussa 2017. (TechCrunch 2017.)

Wit.ai kertoo kotisivujensa mukaan tekevänsä puhetta ja tekstiä käsittelevien sovellusten sekä laitteiden kehittämisen helpoksi ja palvelun soveltuvan niin chatbotien, mobiilisovellusten, kotiautomaatiojärjestelmien, puettavien laitteiden kuin robottienkin teksti- ja äänikäyttöliittymien rakentamiseen. Wit.ai on tällä hetkellä 120 000 kehittäjän käytössä ja sen kerrotaan oppivan ihmiskieltä jokaisesta interaktiosta sekä hyödyntävän oppimisessaan kehittäjien yhteisöä. (Wit.ai 2017a.)

Palvelun lausahduksiin, aikeisiin ja entiteetteihin perustuva toimintaperiaate näyttäytyy kehittäjän silmiin hyvin samankaltaisena kuin LUISissa. Erona LUISiin on beta-versiossa toimiva tuki suomen kielelle. Wit.ai:n tukea suomen kielen luonnollisen kielen käsittelylle on aiemmin hyödyntänyt esimerkiksi RND Works Oy, joka on käyttänyt palvelua Lassila & Tikanojan Kikka-kierrätysbotin toteutuksessa (RND Works 2017).

Wit.ai:n suomen kielelle tarjoamien valmiiden yleisten entiteettien määrä on kuitenkin opinnäytetyön kirjoittamisen suppeampi kuin palvelun englannin kielelle luotujen entiteettien määrä. Esimerkiksi numeroiden käsittelyn mahdollistavien entiteettien puuttuminen suomenkielisestä versiosta on ongelmallista ja numeroiden tunnistaminen joudutaankin prototyypissä opettamaan sovellukselle itse.

Wit.ai:n dokumentaatio ei myöskään yllä kattavuudessaan LUISin tasolle. Erityisesti suomenkielisen version käyttöön ja toimintaperiaatteisiin ei ole vielä olemassa omaa erillistä dokumentaatiota. Dokumentaatio vaikuttaa myös joidenkin ominaisuuksien osalta opinnäytetyön kirjoittamisen hetkellä vanhentuneelta, johon todennäköisesti osin työkalun ominaisuuksien ja palveluiden muutoksista viime aikoina.

#### **6.1.4 Shopify**

Shopify on Kanadan Ottawasta lähtöisin oleva sähköisen kaupankäynnin ohjelmistoyritys ja yrityksen saman nimisen verkkokauppa-alustan nimi. Shopify'n tarina alkaa vuodelta 2004, jolloin yrityksen tarkoituksena oli myydä lumilautoja. Tobias Lütke, joka on yksi Shopify'n perustajista, pettyi kuitenkin markkinnoilla

oleviin verkkokauppa-ohjelmistoihin ja päätti kehittää oman ohjelmiston verkkokauppojen suunnitteluun ja julkaisuun. Shopify-verkkokauppa-alusta julkaistiin vuonna 2006. (Silcoff 2013.) Vuonna 2017 Shopify:n kotisivujen mukaan alustalla toimii 500 000 aktiivista verkkokauppaa ja alustan myynnin volyymi on 40 miljardia dollaria. (Shopify 2017a.)

Shopify:n verkkosivut määrittelevät Shopify:n valmiiksi sähköisen kaupankäynnin ratkaisuksi, jolla myyjät voivat pystyttää verkkokaupan tuotteiden myymiseksi. Sen avulla voidaan organisoida tuotteita, muokata verkkokaupan julkisivua, hyväksyä luottokorttimaksut ja hallita tilauksia ainoastaan muutamilla hiiren painalluksilla (Shopify 2017b).

Yhdysvaltalaisen TechCrunchin vuoteen 2015 asti ylläpitämä start-up-tietokanta Crunchbase kuvaa Shopifyä johtavaksi pilvipohjaiseksi, monikanavaiseksi kauppa-alustaksi, joka on suunniteltu erityisesti pienille ja keskikokoisille yrityksille. Kuvauksen mukaan Shopify-alustaan luottaneita brändejä ovat mm. Tesla, Red Bull ja Nestle. (Crunchbase 2015.)

Shopify listattiin Deloitte Technologyn Fast 500 -listalla vuoden 2014 Pohjois-Amerikan nopeimmin kasvavien yritysten joukossa seitsemänneksi (Columbus 2014). Yritys myös listautui Yhdysvaltojen pörssiin vuonna 2015 (Wilhelm 2015). Vuonna 2017 Shopify:n osakkeiden arvo nousi yrityksen ilmoitettua integraatiosta Amazonin kanssa ja sen toimivan korvaavana sovelluksena Amazonin aiemmalle pienille- ja keskisuurille vähittäismyymijille tarkoitetulle Amazon Webstorelle. Tämä mahdollisti myyjille tuotteiden myynnin Shopify-verkkokaupoista Amazonin suosituilla markkina-alustalla. (Baldwin 2017.)

Shopify julkaisi jo vuonna 2009 kehittäjille rajapinta-alustan sekä sovelluskaupan. Shopify:n rajapintojen avulla kehittäjät voivat luoda sovelluksia Shopify-verkkokaupoille ja myydä luotuja sovelluksia Shopify App Storessa. Shopify:n perustajan Tobias Lütken mukaan rajapinnan tarkoituksena on mahdollistaa käyttäjilleen entistä enemmän vaihtoehtoja kauppajensa personointiin ja uniikkien ostokokemusten luontiin. (Shopify 2009.)

## 6.2 Toteutuksen prosessi

### 6.2.1 Toteutuksen toimintaperiaatteet ja arkkitehtuuri

Prototyypin chatbot ohjelmoitiin vastaamaan kolmeen eri tyyppiseen käyttäjän toimintoon:

1. Chatbotin tunnistessa käyttäjän tervehdyksen, se vastaa tervehdykseen valitsemalla sattumanvaraisesti yhden sille määritellyn tervehdyksen, kertomalla olevansa prototyyppi chatbotista ja kehottamalla käyttäjää tekemään haku tai pyytämään ohjetta. Jos käyttäjä tervehtii keskustelun aikana uudestaan, vastaa chatbot ainoastaan tervehdyksellä.
2. Chatbotin tunnistessa käyttäjän tarpeen ohjeelle tai avulle, se vastaa esittämälle käyttäjälle ohjeita hausta sekä muutamia esimerkkejä hakulauseista.
3. Chatbotin tunnistessa käyttäjän hakuaikeen, se pyrkii poimimaan käyttäjän syötteestä haulle esitetyt kriteerit ja tulostaa tunnistetut hakukriteerit käyttäjälle. Tunnistettujen hakukriteerien perusteella chatbot hakee Shopify:n rajapinnan kautta kriteerejä vastaavat verkkokaupasta löytyneet tuotteet. Löytyneet tuotteet esitetään emulaattorissa karusellimuodossa listana kortteja. Kortit sisältävät tuotteen otsikon, hinnan sekä oletuskuvan. Lisäksi kortissa on kaksi painiketta. Ensimmäisen painike avaa tuotteen verkkosivun selaimessa. Toinen painike avaa näkyville uuden kortin, joka sisältää lisää kuvia tuotteesta sekä tuotteen kuvauksen.

Käyttäjän syötteisiin, joihin LUIS- tai Wit.ai-palvelu ei pysty liittämään esitettyjä kolmea toimintoa, vastataan kertomalla ettei chatbot ymmärrä käyttäjän viestiä ja kehoitetaan käyttäjää yrittämään hakua tai pyytämään ohjetta.

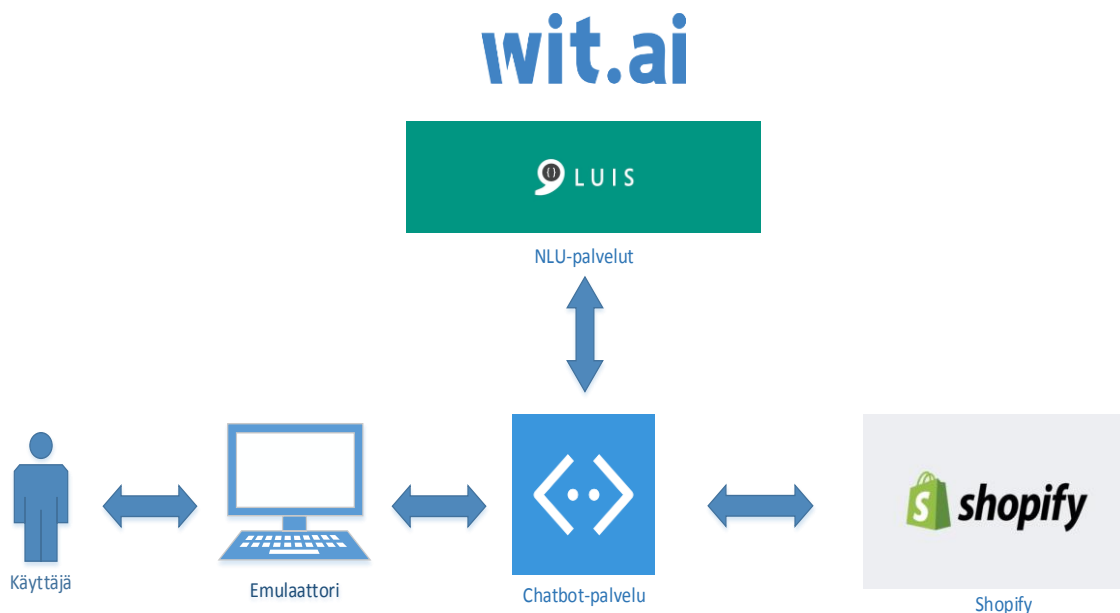
Prototyypissä chatbot vastaanottaa käyttäjän emulaattorin kautta kirjoittamat luonnollisen kielen syötteet joko englanniksi tai suomeksi. Chatbot-sovellus lä-

hettää vastaanotetun syötteen Wit.ai- tai LUIS-sovelluksen rajapinnalle, jotka pyrkivät kehittäjän luoman mallin perusteella määrittelemään syötteen intentin eli käyttäjän aikeen sekä hakuaikeen tapauksessa syötteen sisältämät entiteetit eli hakukriteerit. Aikeet ovat prototyypin kohdalla tervehtiminen, ohjeen pyytäminen ja tuotehaku. Entiteetit ovat haussa esiintyvät kriteerit eli prototyypissä kameran valmistaja, malli ja tyyppi. Lisäksi mallit pyrkivät löytämään entiteetteinä käyttäjän mahdollisesti syöttämän hinnan tai hinnat sekä haluaako käyttäjä etsiä tuotetta alle vai yli annetun hinnan tai kahden hinnan väliltä.

Wit.ai ja LUIS palauttavat tulokset todennäköisimmästä aikeesta ja tunnistetuista entiteetistä JSON-muodossa chatbot-sovellukselle. Sovellus tarkastaa ensin syötteestä tunnistetun aikeen ja siirtyy tämän perusteella toimintoa vastaavaan dialogiin. Microsoft Bot Frameworkin tapauksessa dialogeilla mallinnetaan keskustelua ja sen kulkua. Dialogeja käsitellään tarkemmin luvussa 6.2.6.

Prototyypissä chatbotille on luotu viisi varsinaista dialogia, joista kolme tulostaa käyttäjälle ainoastaan chatbotille valmiiksi määritellyn yksittäisen tai useamman tulosteen. Kaksi dialogeista käyttävät toteutuksessa erilliseen Shopify.js-tiedostoon laadittua moduulia, joka sisältää toiminnallisuuden Shopify:n rajapintaan yhdistämiselle sekä funktiot rajapintaan tehtäville hauille. Shopify:n rajapinta palauttaa tiedot tuotteista JSON-muodossa. Mikäli käyttäjä on esittänyt hakulauseessa rajoituksia hinnalle, suorittaa chatbotin ohjelmakoodi tuotteiden suodattamisen hintojen perusteella. Lopulta hakutulokset tulostetaan käyttäjälle.

Kuva 1 esittää prototyypin arkkitehtuuria yleisellä tasolla. Julkaistaessa chatbot, korvautuu emulaattori valituilla julkaisukanavilla kuten esimerkiksi pikaviestimillä, mobiiliapplikaatiolla, webchatilla tai tekstiviestipalvelulla. Tällöin myös julkaisukanavan ja chatbot-palvelun välissä käytetään Microsoftin Bot Framework Connectoria, jonka tehtävänä on tarjota REST-rajapinta eri julkaisualustoille ja mahdollistaa käyttäjän kommunikointi palvelun kanssa välittämällä viestejä julkaisukanavan ja chatbotin välillä. Prototyypissä chatbotia käytetään ainoastaan emulaattorin välityksellä, jolloin Bot Connectoria ei tarvita.



Kuva 1. Yleiskuva prototyypin arkkitehtuurista.

## 6.2.2 Shopify-rajapinnat ja yhdistämismenetelmä

Yksi ensimmäisistä tehtävistä arkkitehtuuria ja toimintaperiaatetta suunniteltaessa oli määrittellä, kuinka sovellus on yhteydessä Shopify:n rajapintaan ja minkälaiset haut sen välityksellä ovat mahdollisia. Shopify:n kaksi päärajapintaa ovat Admin API sekä Storefront API. Näistä Admin API on tarkoitettu kaupan hallitsijalle suunnattuihin sovelluksiin, kun taas Storefront API on tarkoitettu personalisoitujen ostokokemusten luontiin kaupan asiakkaille.

Rajapintojen dokumentaatiota tutkimalla kirjoittaja löysi lisäksi JavaScript Buy SDK:n. Shopify:n dokumentaatio kuvaa JavaScript Buy SDK:ta kevyeksi kirjastoksi, joka mahdollistaa verkkokaupan rakentamisen mille tahansa verkkosivustolle. Se pohjaa Shopify:n rajapintaan ja tarjoaa mahdollisuudet tuotteiden ja collectioneiden hakemiseen kaupasta, tuotteiden lisäämisen ostoskoriin sekä maksutapahtuman. (Shopify 2017c.) Collectionit ovat Shopify:ssä kokoelmia tuotteista ja ne voidaan rinnastaa tuotekategorioihin.

JavaScript Buy SDK vaikuttaa mahdollistavan lähes samat toiminnot tuotehaku-  
jen osalta kuin kaksi päärajapintaa. Koska chatbot toteutettiin JavaScript-ohjel-  
mointikielellä ja Shopify:n oman kuvauksen tukiessa valintaa, otettiin JavaScript  
Buy SDK hyödynnettäväksi prototyypissä sovelluksen kommunikointiin Shopify-  
rajapinnan kanssa.

Luotaessa sovellusta Shopify:lle, on kehittäjän määriteltävä onko sovellus tarkoi-  
tettu julkiseksi ja toimimaan useiden kauppojen yhteydessä vai onko kyseessä  
yksityinen sovellus, joka toimii ainoastaan tietyn yksittäisen kaupan kohdalla.  
Prototyypin tapauksessa sovellus edustaa ainoastaan yhtä kauppaa ja on yksi-  
tyinen. Shopify-verkkokauppaan yhdistämiseen tarvitaan ainoastaan kaupan  
Storefront access token sekä Shopify-domain eli kaupan url.

JavaScript Buy SDK mahdollistaa kaikkien tuotteiden hakemisen kaupasta, tiet-  
tyyn collectioniin kuuluvien tuotteiden hakemisen ja yksittäisten tuotteiden haun.  
Yksittäisiä tuotteita voidaan hakea tuotteen id:n, handle:n, viimeisen muokkaus-  
päivämäärän ja korkeintaan kymmenen eri tagin perusteella. (Shopify 2017d.)  
Handle:t ovat useimmiten Shopify:n tuotteen otsikosta automaattisesti generoimia  
merkkijonoja, joiden perusteella luodaan esimerkiksi tuotteen url (Shopify 2017e).  
Tagit taas ovat asiakkaalle näkymättömiä avainsanoja, joilla voidaan organisoida  
ja kategorisoida tuotteita sekä helpottaa niiden hakua ja suodatusta. Lisäksi haun  
palauttamien tulosten määrää voidaan rajata ja palautettavat tuotteet voidaan jär-  
jestellä esimerkiksi hinnan ja otsikon mukaan. (Shopify 2017f.)

Alustaviksi hakukriteereiksi suunniteltiin tuotteen valmistajaa, mallia, tyyppiä  
sekä hintaa. Näistä ainoastaan tuotteen tyyppiä pystytään hakemaan suoraan  
collectionin muodossa. Valmistajaan ja malliin kohdistuvat haun on toteutettava  
tuotteen tagien perusteella. Koska tuotteen tyyppin tai kategorian voidaan olettaa  
usein sisältyvän myös tuotteen avainsanoihin, voidaan haku kameran tyyppille to-  
teuttaa myös tagien avulla collectioneiden sijasta. Tuotteiden hakeminen hintojen  
perusteella täytyy toteuttaa hakemalla ensin lista tuotteista ja suodattamalla  
vasta sitten ohjelmakoodin avulla löydetyistä tuotteista ne tuotteet, joiden hinta  
vastaa käyttäjän esittämiä kriteerejä.

### 6.2.3 Testiverkkokaupan luominen

Toimeksiantajalta saadun ehdotuksen perusteella prototyypissä toteutettavan testiverkkokaupan päätettiin mallintavan kamerakauppaa. Opinnäytetyön kirjoittajalla ei ole asiantuntemusta kameroista tai kuvaamisesta, joten mallia kaupalle jouduttiin hakemaan verkkohauilla löytyneistä olemassa olevista kamerakaupoista.

Kameroita myyvät verkkokaupat vaikuttivat tutustumisen perusteella keskittyvän eri tyyppisten kameroiden sekä niihin liittyvien lisävarusteiden myyntiin. Myytävien tuotteiden määrä vaihteli eri kauppojen osalta paljon. Osat kaupoista keskittyivät myymään vain muutamia eri merkkiä kameroita, kun taas suurimmilla kaupoilla erilaisia kameroita ja lisävarusteita on tarjolla satoja. Prototyypitoteutuksessa ei ollut aikataulun vuoksi tarkoituksena luoda täysin kattavaa ja toiminnallista verkkokauppaa, vaan ainoastaan suhteellisen kevyt toteutus, joka olisi riittävä luonnollisen kielen tuotehaun havainnollistamiseksi.

Kirjoittaja otti tuotteina mukaan ainoastaan eri tyyppisiä kameroita. Lisävarusteiden lisääminen olisi toteutettu myöhemmin, mutta tälle ei aikataulun osalta jäänyt aikaa. Kameroita pyrittiin valitsemaan kolmesta eri kategoriasta. Kirjoittaja otti lajittelutavaksi muutamissa muissa kaupoissa kuten esimerkiksi verkkokauppa.com:issa esiintyneen jaon digitaalikameroihin, action-kameroihin sekä järjestelmäkameroihin. Näistä jokaiseen kategoriaan lisättiin kymmenen eri mallista kameraa, joten tuotteiden yhteismääräksi tuli 30 kappaletta. Testiverkkokaupaan lisätyt kamerat ovat yhdeksältä eri valmistajalta ja kameroiden hinnat vaihtelevat noin 100 euron ja 12 000 euron välillä.

Testiverkkokaupan luominen vaati ensin Shopify Partner -tilin luomisen. Shopify Partner -ohjelmaan liittyminen on kirjoittamisen hetkellä ilmaista (Shopify 2017g). Shopify Partnerina kehittäjän on mahdollista luoda rajaton määrä kehitysverkko-kauppoja, joita voidaan hyödyntää demoissa ja portfolioissa (Shopify 2017h). Luotu tili mahdollistaa pääsyn Shopify Partner Dashboardille, jonka kautta kehittäjä pääsee luomaan testiverkkokauppoja.



Kaupan luontia varten vaadittavat tiedot olivat ainoastaan kaupan nimi, käyttäjänimi ja salasana kaupan ylläpitäjälle sekä kaupan osoitetiedot. Luotuaan kaupan, pääsee kehittäjä luomaan kaupalle mm. collectioneita, lisäämään tuotteita, hallitsemaan tilauksia ja asiakkaita sekä kauppaan liitettäviä applikaatioita. Vaikka tuotehaut toteutetaan suomeksi ja englanniksi, toteutettiin verkkokauppaan sisältyneet tekstit ainoastaan englannin kielellä. Tällä tarkoitetaan collectionien nimiä, collectioneiden kuvauksia, tuotteiden otsikoita, tuotteiden kuvauksia, tuotetyyppejä sekä tagien avainsanoja.

Ensimmäisenä testiverkkokaupalle luotiin collectionit. Testiverkkokaupan tapauksessa collectioneja luotiin yhteensä neljä kappaletta. Nämä olivat kaikki kamerat sisältävä *Cameras*, digitaalikamerat sisältävä *Digital Cameras*, action-kamerat sisältävä *Action Cameras* sekä järjestelmäkamerat sisältävä *Dslr Cameras*. Jokaiselle kategorialle asetettiin myös lyhyt kuvaus.

Jälkimmäiseen kolmeen kategoriaan lisättiin kymmenen niihin luokiteltavaa tuotetta. Mallia kameroiden lajitteluun kategorioittain otettiin taas <https://www.verkkokauppa.com> -sivustolta. Jokaiselle tuotteelle asetettiin otsikko, kuvaus, tuotetyppi, jälleenmyyjä, collection, 3–6 kuvaa sekä hinta euroissa. Tuotetyyppien nimet asetettiin vastaamaan collectionien nimiä ja jälleenmyyjäksi asetettiin kameran valmistaja. Tuotteiden otsikot, kuvaukset ja kuvat kopioitiin <https://www.bhphotovideo.com> -sivustolta. Sivustolla kameroiden tiedot ja kuvaukset vaikuttivat hyvin yhtenäisiltä, joten näin esimerkiksi testiverkkokaupan kuvausten tekstit saatiin ulkoasultaan toisiaan vastaaviksi. Kuvauksista testiverkkokauppan kuvauksiin otettiin alkuvaiheessa ainoastaan yleiskuvaus sekä tuotteen ”kohokohdat” (engl. product highlights).

Tuotteiden tageiksi asetettiin aluksi ainoastaan tuotteen valmistajan nimi sekä kameran tyyppi. Digitaalikameroille *Digital Camera*, actionkameroille *Action Camera* ja järjestelmäkameroille *DSLR Camera*. Myöhemmin, kun prototyypin englanninkieliseen toteutukseen lisättiin mahdollisuus hakea tuotteita mallin perusteella, lisättiin tageihin myös mallien nimet. Mallit, joiden nimet koostuvat useammasta sanasta tai merkkijonosta, paloitteltiin yksittäisiksi tageiksi. Tämä paloittelu tehtiin, koska oletettiin ettei käyttäjä aina muista mallin koko nimeä tai

kirjoita sitä täysin vastaavassa kirjoitusasussa. Esimerkiksi kameralle, jonka malli on *EOS 5D Mark IV* asetettiin tagit *EOS*, *5D*, *Mark* ja *IV*.

Tuotteiden varianttien hyödyntämistä pohdittiin suunnitteluvaiheessa. Varianteilla tarkoitetaan eri versioita samasta tuotteesta esimerkiksi eri värillä tai koolla. Testiverkkokauppa sisälsi joitain tuotteita, joista esiintyi eri versioita. Esimerkiksi yhtä digitaalikameraa myytiin eri värisenä ja yksi järjestelmäkamera on saatavilla pelkänä runkona tai linssin kanssa. Toiminto, joka ehdottaa tai näyttää hakutuloksessa saadulle tuotteelle eri versioita, olisi täysin mahdollista toteuttaa, mutta prototyypistä se jouduttiin jättämään pois aikataulullisista syistä. Lopulta prototyypissä eri versioita kohdellaan erillisinä tuotteina. Tuotteen eri versioiden ehdottaminen olisi ehdottomasti yksi asiakaskokemukseen vaikuttava toiminnallisuus, joka olisi hyödyllistä lisätä chatbotiin, jos toteutusta jatkokehittäisiin ja se julkaistaisiin aidoille asiakkaille.

#### **6.2.4 Projektin alustus ja Microsoft Bot Builderin käyttöönotto**

Prototyypin kehittäminen aloitettiin luomalla Node.js-projekti ja ottamalla käyttöön Microsoft Bot Frameworkin Bot Builder Node.js SDK. Kirjoittajalla oli asennettuna koneella kirjoittamisen hetkellä Node.js:n uusin versio 6.11.4 sekä npm-paketin hallinnan versio 5.5.1. Node.js-projekti alustettiin luomalla koneelle hakemisto ja navigoimalla tähän komentorivin kautta. Komentorivillä annettiin komento *npm init* ja täytettiin prototyypin osalta vaadittavat tiedot. Kirjoittaja muutti oletustiedoista paketin nimen, kirjoitti lyhyen kuvauksen projektille, asetti url:in git-repositorylle ja lisäsi tekijäksi opinnäytetyön kirjoittajan nimen. Projektin hakemistoon luodaan komennon jälkeen automaattisesti package.json-tiedosto, johon talletetaan tiedot ja versiot eri paketeista, joista projekti on riippuvainen.

Kirjoittaja loi projektille myös repositoryn Bitbucket-palveluun, jota käytettiin projektin versionhallintaan. Repositoryyn talletettiin ohjelmakoodi sekä LUIS- ja Wit.ai-palveluihin luotujen mallien tiedot, jotka saatiin JSON-muodossa kyseisten palveluiden export-toimintojen avulla.

Chatbotille luotiin prototyypissä kaksi erillistä App.js-tiedostoa. App.js-tiedostot sisältävät ohjelmakoodin, joka tarvitaan sovelluksen käynnistämiseen ja chatbot-palvelun luomiseen. App.js-tiedostoissa määritellään myös mitä palvelua chatbot käyttää luonnollisen kielen ymmärtämiseen. Prototyypissä käytettiin kahta eri palvelua, joten tämän vuoksi tiedostosta oli luotava kaksi erillistä toteutusta.

App.js-tiedostoissa määritellään myös chatbotin dialoginhallinta. Dialogit voidaan käynnistää LUIS tai Wit.ai-palvelujen tunnistaman todennäköisimmän aikeen perusteella, käynnistämällä dialogi ohjelmakoodin komennon kautta tai tunnistamalla käyttäjän syötteestä dialogin käynnistämisen ehdoksi määritetyjä avainsanoja tai malleja. Pitkien dialogien sisältö voidaan sijoittaa erilliseen tiedostoon kuten prototyypin tapauksessa hakutoiminnon dialogi sekä tuotteista lisätiedot näyttävä dialogi. Niiden käynnistämisen ehdot ja minkä polun päästä niiden sisältö on sovelluksen löydettävissä määritellään kuitenkin App.js-tiedostoissa. Prototyypin lyhyemmät dialogit kuten oletusdialogi, tervehtiminen sekä ohjeen esittäminen ovat sijoitettuna App.js-tiedostoon.

Microsoftin Bot Frameworkiin sisältyvän chatbotien rakentamiseen tarkoitetun Bot Builder Node.js SDK saatiin käyttöön komennolla *npm install -save botbuilder*. Paketti otetaan käyttöön App.js-tiedostoissa komennolla *const builder = require('botbuilder');*

Chatbotin tulee myös toimia tietyn rajapinnan päätepisteen päässä. Tätä varten sovelluksessa käytetään Express-websovelluskehystä. Express asennettiin komennolla *npm install -save express* ja se otettiin käyttöön koodissa komennolla *const app = require(express);*

Chatbotin luominen vaatii ensin ChatConnector-luokan toteutuksen ja määrittelemisen. ChatConnectorille määriteltäisiin chatbotin julkaisemisen yhteydessä Microsoftin AppId sekä AppPassword. Prototyypin toimiessa ainoastaan emulaattorissa, jätetään nämä tiedot tyhjäksi. ChatConnector annetaan parametrina luotaessa uusi toteutus UniversalBot-luokasta. UniversalBotille voidaan sen konstruktorissa määritellä asetuksia kuten esim. lokalisoinnin kieli ja lokalisointitiedostojen polku. Lisäksi sille voidaan konstruktorissa määritellä oletusdialogi, joka

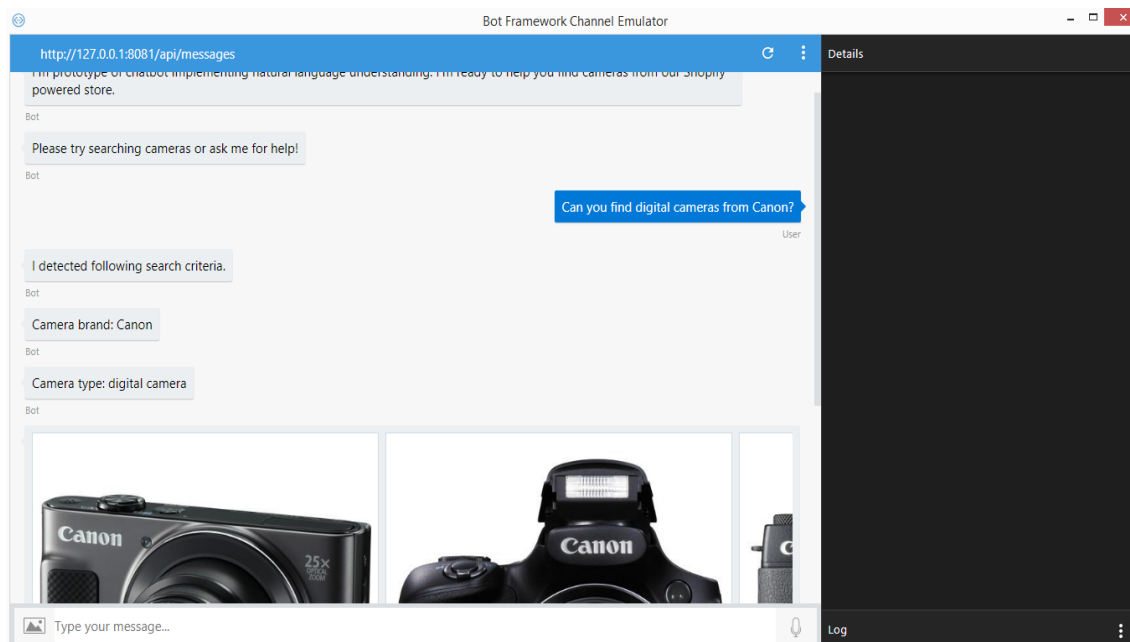
käynnistetään, jos chatbot ei pysty syötteen perusteella käynnistämään mitään muuta dialogia. Prototyypissä oletusdialogi määriteltiin konstruktorin ulkopuolella luomalla dialogi, joka saa osoitteekseen ainoastaan `/-`-merkin.

Lopulta express-aplikaatio asetetaan kuuntelemaan sille määriteltyä porttia ja ChatConnector kuuntelemaan päätepisteeseen `/api/messages` saapuvia viestejä. Sovellus käynnistetään navigoimalla komentorivillä halutun kielen `App.js`-tiedoston sijaintiin ja käynnistämällä sovellus käskyllä `node app.js`.

### 6.2.5 Testaaminen emulaattorissa

Microsoft Bot Framework tarjoaa chatbotien kehittäjille emulaattorin, jossa chatbotin toimintaa voidaan testata. Emulaattori pitää ladata ja asentaa erikseen. Kirjoittamisen hetkellä emulaattorin asennustiedostot voi ladata osoitteesta <https://github.com/Microsoft/BotFramework-Emulator/releases/tag/v3.5.32>.

Emulaattorin käynnistyessä sen ylälaitaan kirjoitetaan prototyypin tapauksessa osoite `http://127.0.0.1:8081/api/messages`. Emulaattori pyytää Microsoftin App ID:tä ja App Passwordia sekä Localea, mutta nämä voidaan prototyypin tapauksessa jättää täyttämättä. Node.js-sovelluksen ollessa käynnissä voidaan sovellukseen yhdistää painamalla *Connect*-painiketta. Tämän jälkeen emulaattorin käyttäjä voi lähettää ja vastaanottaa viestejä chatbot-sovellukselta. Kuva 2 esittää esimerkin emulaattorin näkymästä.



Kuva 2. Näkymä emulaattorista kehityksen aikana.

Kirjoittaja pitää emulaattoria erinomaisena työkaluna chatbotin testaamiseen, mutta joidenkin toimintojen testaaminen oli emulaattorissa ajoittain hidasta. Näitä toimintoja olivat esimerkiksi liitetiedostojen lisääminen viesteihin sekä karusellinäkömön vierittäminen sivusuuntaan.

### 6.2.6 Perusdialogit

Chatbotin perusdialogeilla kirjoittaja tarkoittaa prototyypissä kaikkia muita kuin hakutoimintoihin liittyviä dialogeja. Näitä ovat prototyypissä oletusdialogi, tervehdintä ja ohjeen näyttäminen. Prototyypissä nämä dialogit ovat melko lyhyitä ja tulostavat käyttäjälle ainoastaan yhden tai muutaman tulosteen. Näiden dialogien sisällöt kirjoitettiin eri kielten App.js-tiedostoihin ja niiden sisältämät tulosteet kirjoitettiin suoraan ohjelmakoodiin kyseisellä kielellä.

Dialogeille annetaan parametreinä niihin viittavaa osoite sekä joko yksittäinen funktio tai joukko funktioita. Nämä funktiot ottavat vastaan seuraavat parametrit: *session* (instanssi keskustelusta), *args* (tietoja keskustelusta) sekä *next* (seuraava funktio, jos dialogi koostuu joukosta funktioita). *Args* ja *next* ovat prototyyp-

pin perusdialogeissa osittain turhia, sillä dialogit eivät käsittele keskusteluun talletettuja tietoja eikä niille ole määritelty funktiosarjoja. Bot Frameworkin vesiputousmaisella toimintaperiaatteella toimivat funktiosarjat ovat useimmiten käytössä, kun dialogi koostuu esimerkiksi vuoropuhelusta käyttäjän ja chatbotin välillä.

Dialogeille voidaan myös määritellä millä perusteella ne käynnistetään, jos niitä ei käynnistetä erikseen ohjelmakoodissa. Muussa tapauksessa dialogeille voidaan määritellä *triggerAction*, joka käynnistää dialogin esimerkiksi LUISin tai Wit.ai:n palauttaman todennäköisimmän aikeen perusteella tai tunnistamalla käyttäjän syötteestä määriteltyjä avainsanoja tai säännöllisiä lausekkeita. Prototyypissä dialogien *triggerActionit* tunnistavat käyttäjän syötteistä tunnistettujen aikeiden nimiä.

Kuva 3 esittää esimerkin tervehtimisen toteuttavasta dialogista. Metodi *session.send* lähettää käyttäjälle sille parameterina annetun viestin. Esimerkin tapauksessa sille annetaan parametrina taulukko, joka sisältää erilaisia viestejä. Kun komento saa parametrina taulukon, valitaan taulukon alkioista aina sattumanvaraisesti yksi lähetettävä viesti. Komento *session.beginDialog* käynnistää sille parametrina annetun dialogin osoitteen. Tässä tapauksessa käynnistettävän dialogin osoite on */-*merkki, joka on oletusdialogin osoite.

```
bot.dialog('/Greeting',function(session, args, next) {
    session.send(["Hei!", "Moi!", "Terve!", "Morjes!", "Tervehdys!"]);
    session.beginDialog('/');
}).triggerAction({
    matches: 'Greeting'
});
```

Kuva 3. Esimerkki tervehdys-dialogista.

Aikeiden nimet määriteltiin molempien kielten toteutuksissa englanniksi. Tervehtimisen toteuttava dialogi käynnistetään tunnistettaessa aie *Greeting* ja ohjeen näyttävä dialogi tunnistettaessa aie *Help*. Tunnistettaessa aie *None* siirrytään keskustelussa oletusdialogiin.

### 6.2.7 NLU-palveluiden käyttöönotto ja yhdistäminen chatbotiin

LUIS-palveluun rekisteröityminen vaatii Microsoft-tiliä. Palvelu toimii graafisella käyttöliittymällä verkkoselaimessa osoitteessa <https://www.luis.ai>.

Uudelle LUIS-sovellukselle määritellään nimi ja kulttuuri, joka vastaa mallissa käytettävää kieltä. Lisäksi sovellukselle voidaan antaa vapaaehtoinen kuvaus. Prototyypin toteutuksessa kulttuuriksi valittiin Englanti. Ennen sovelluksen ensimmäistä julkaisemista sille täytyy luoda alustava malli ja harjoittaa sovellusta vähintään kerran. Tämän jälkeen käyttöliittymän osiosta *Publish App* voidaan määrittellä julkaisua varten tarvittavat tiedot. Sovellukselle pitää vähintään kertoa onko julkaistava malli tarkoitettu tuotantoon vai kehitykseen. Prototyypin kohdalla valittiin *staging*-vaihtoehto, joka tarkoittaa, että sovellus on tarkoitettu kehitykseen. Sovellukselle voidaan myös määrittää aikavyöhyke ja valita otetaanko käyttöön Bing Spell Checker. Bing Spell Checker tarkistaa ja pyrkii korjaamaan virheelliset syötteet oikeinkirjoituksen osalta. Prototyypissä Bing Spell Checker otettiin käyttöön. Samassa ikkunassa saadaan myös sovelluksen rajapinnan päätepisteen osoite.

Bot Builder sisältää valmiin tuen LUIS-palvelulle. Yhdistämistä varten luodaan uusi instanssi LuisRecognizer-luokasta ja annetaan sille parametrina LUIS-sovelluksen rajapinnan päätepisteen osoite, jonka hankkiminen kuvattiin edellisessä kappaleessa. Tämä asetetaan chatbotin recognizeriksi, joka on tarkoitettu tarkastelemaan viestejä ja määrittelemään niiden aikeen. Tämän jälkeen chatbot pystyy käynnistämään eri dialogeja LUISin palauttaman todennäköisimmän aikeen nimen perusteella. Kuva 4 kuvaa ohjelmakoodirivit, joilla chatbot asetetaan tunnistamaan LUIS-sovelluksen palauttamia aikeita.

```

var model = process.env.model || 'https://westus.api.cognitive.microsoft.com/luis/v2.0/apps/7a0df9e2-a338-44ec-86f0-6864e411d81d?subscription-key=d95e55628bac41e7a8e2313252eb2cf4&staging=true&timezoneOffset=0&verbose=true&spellCheck=true&q=';
bot.recognizer(new builder.LuisRecognizer(model));

```

Kuva 4. LUIS-sovelluksen yhdistäminen chatbotiin

Wit.ai-palveluun kirjautuminen on kirjoittamisen hetkellä mahdollista ainoastaan GitHub- tai Facebook-tunnuksilla. Palvelu ei tarjoa muuta erillistä tapaa rekisteröitymiseen. Sovellus toimii graafisella käyttöliittymällä verkkoselaimessa osoitteessa <https://wit.ai/>.

Uudelle Wit.ai-sovellukselle annetaan nimi ja kuvaus. Lisäksi määritellään sovelluksessa käytettävä kieli sekä halutaanko sovellus pitää yksityisenä vai julkisena. Jos sovellus pidetään julkisena, on sen data avointa koko Wit.ai-yhteisölle. Prototyypin tapauksessa kieleksi valittiin suomi ja sovellus asetettiin yksityiseksi. Uusi sovellus on myös mahdollista luoda olemassa olevasta mallista, jos sellainen on aiemmin exportattu aiemmasta Wit.ai-sovelluksesta. Kun sovellus on luotu, saadaan käyttöliittymän *Settings*-osion takaa tarvittavat tiedot rajapintaan yhdistämiselle. Prototyypin tapauksessa toteutusta varten tarvittiin sovelluksen App Id.

Bot Builderissä ei ole valmista tukea Wit.ai-sovelluksen käyttämiselle chatbotin recognizerina. Prototyypissä käytetään tämän mahdollistamiseksi kolmannen osapuolen laatimaa moduulia. Moduuli löytyy Githubista osoitteesta <https://github.com/sebsylvester/botbuilder-wit>. Moduulin laatija esiintyy Githubissa nimerkillä *sebsylvester*. Samalta laatijalta löytyy julkaisuja myös Microsoftin Githubista löytyvissä Bot Builderille luoduissa esimerkeissä, joten laatijan toteutus vaikuttaa kirjoittajan silmissä tarpeeksi luotettavalta.

Moduuli saatiin käyttöön asentamalla se npm-paketinhallinnan kautta komennolla `npm install -save botbuilder-wit`. Ohjelmakoodissa moduuli otettiin käyttöön dokumentaation esimerkin mukaisesti komennolla `const {WitRecognizer} = require('botbuilder-wit');`. Tämän jälkeen prototyypissä luodaan uusi instanssi



WitRecognizerista ja annetaan sille parameterina edellisen kappaleen kuvauksen mukaisesti hankittu App Id. Chatbotin recognizer asetettiin komennolla *bot.recognizer(new Witrecognizer('App Id'))*; Näin chatbot saadaan käyttämään aikeiden tunnistamiseen ja dialogien käynnistämiseen Wit.ai-palvelua samalla tapaa kuin LUISia.

### 6.2.8 LUISin ja Wit.ai:n peruskonseptit

LUIS- ja Wit.ai-palveluilla on kirjoittajan havaintojen mukaan muutamia yhteisiä piirteitä. Molemmissa esiintyy kolme tärkeää pääkonseptia, jotka kehittäjän on välttämätöntä ymmärtää. Kirjoittaja kuvaa näitä opinnäytetyössä suomennetuin termein lausahdukset (engl. utterances), aikeet (engl. intents) ja entiteetit (engl. entities).

Mallien rakentaminen aloitetaan määrittelemällä aikeet. Aie voi olla esimerkiksi ohjeen pyytäminen tai tuotteen hakeminen ja se voidaan rinnastaa lauseen verbiin. Aikeelle annetaan sovelluksissa aietta kuvaava nimi. Kun aie on luotu, sille annetaan esimerkkejä lausahduksista. Lausahdukset vastaavat käyttäjältä saatavia tekstisyötteitä. Esimerkiksi aikeelle *tuotehaku* pyritään antamaan mahdollisimman paljon erilaisia tuotteen hakemista vastaavia käyttäjän syötteitä kuten esim. *Etsi kameroita* tai *Hae kirjoja*. Mitä enemmän esimerkkilausehduksia sovellukselle annetaan, sitä paremmin sen pitäisi pystyä tunnistamaan käyttäjän lausahduksen aie, vaikka lausahdus ei esiintyisikään täysin vastaavassa muodossa kuin sovellukselle annetuissa esimerkeissä.

Entiteetit ovat lausahduksista poimittavia aikeeseen liittyviä ja sitä tarkentavia tietoja. Jos aikeet voitiin rinnastaa lauseen verbeihin, voidaan entiteetit nähdä lauseen substantiiveina. Entiteetti voi olla esimerkiksi paikka, päivämäärä, nimi tai tuote. Aiemman kappaleen tuotehaku-aikeen esimerkeille voitaisiin määrittää entiteetti *tuote*. Edeltävässä kappaleessa esiintyvistä esimerkeistä voitaisiin silloin pyrkiä tunnistamaan entiteettinä *kameroita* tai *kirjoja*. Myös entiteettien tunnistamisen tarkkuuden pitäisi kasvaa, mitä useampia esimerkkejä sovellukselle annetaan lausahduksista ja niiden sisältämistä entiteeteistä.

Palveluiden käyttöliittymät eroavat aikeiden ja lausahdusten lisäämisessä sekä entiteettien merkitsemisessä jonkin verran. Pieniä eroja käyttöliittymän yksityiskohdissa ei opinnäytetyössä lähdetä selvittämään tarkemmin. Pääasiassa palvelujen peruseriaate on kuitenkin hyvin samankaltainen ja käyttäjän opittua käyttämään toista palvelua, on toisen palvelun peruseriaatteet opittavissa melko nopeasti. Molemmissa palveluissa käyttäjän on mahdollista luoda ja nimetä aiheet. Näihin liitetyt lausahdukset on mahdollista lisätä olemassa olevasta datasta tai lisäämällä niitä manuaalisesti kirjoittamalla ne yksitellen käyttöliittymien tekstikenttään. Lisätyistä lausahduksista voidaan merkitä tiettyjä lausahduksen osia entiteeteiksi hiirtä käyttämällä. Entiteetti voi olla käyttäjän itse määrittämä tai sovelluksen esirakentama.

LUIS tarjoaa kehittäjälle useita esirakennettuja domaineja eli aihealueita, jotka sisältävät esirakennettuja aikeita sekä entiteettejä. Opinnäytetyön aikana kirjoittaja kuitenkin havaitsi, että esirakennetuista aikeista esimerkiksi avun pyytäminen sisältää ainoastaan kymmenen valmista esimerkkilausahdusta ja määrä ei kirjoittajan kokemuksen mukaan riittänyt tunnistamaan kovin monipuolisesti käyttäjän eri syötteitä avun tai ohjeen pyytämiseksi. Esirakennetut aiheet ovat hyvä lähtökohta monelle toteutukselle, mutta kehittäjän on luultavasti tarpeellista lisätä lausahdusten määrää itse.

LUIS tarjoaa englannin kielelle kirjoittamisen hetkellä 13 erilaista esirakennettua entiteettiä. Näitä ovat mm. paikka, päivämäärä, numero, email ja raha. Näiden entiteettien taso ja tunnistamisen kattavuus oli kuitenkin kirjoittajan kokemuksen mukaan vaihtelevaa. Esimerkiksi LUISin esirakennettu entiteetti rahan arvolle, tunnisti arvon hyvin, kun se on kirjoitettu numeromuodossa ja sen perässä on joko symboli € tai sana *euro(s)*. Jos käyttäjä kuitenkin käyttää pikaviestinnässä usein käytettyä e-kirjainta merkitsemään rahan arvoa, ei LUIS pysty tätä tunnistamaan.

Wit.ai ei tarjoa lainkaan esirakennettuja aikeita suomen kielelle. Esirakennettuja entiteettejä on 14 kappaletta. Näitä ovat esim. ikä, paikka, henkilön nimi ja erilai-

set hakulausekkeet mm. Wikipediaan. Numeroille tai rahan arvoille ei löydy suomen kielen versiossa valmiita entiteettejä, vaikka yksi esirakennetuista entiteeteistä on kuitenkin tarkoitettu matemaattisten lausekkeiden tunnistamiseen.

LUISissa kehittäjän itse määrittelemien entiteettien tyypit voivat olla simple, list, hierarchial tai composite. Simple on geneerinen entiteetti, joka kuvaa yhtä yksittäistä konseptia. List-entiteettityypillä käyttäjä voi lisätä sovellukseen listan avainsanoja sekä näihin liitettäviä synonyymejä. LUISin tunnistessa avainsanan tai sen synonyymien, palauttaa se entiteetin arvona ainoastaan avainsanan. LUIS ei käytä kuitenkaan käytä koneoppimista lisätäkseen uusia sanoja listaan, joten lista on ainoastaan suljettu sarja tunnettuja arvoja. Hierarchical-entiteettityypillä on mahdollista määritellä kategoria ja sen jäsenet. Esimerkiksi *paikka*-entiteetti voi koostua kahdesta lapsientiteetistä, joita ovat lähtöpaikka ja määränpää. Composite-entiteettityypin voi sisältyä kaikkia aiemmin esiteltyjä entiteettityyppejä ja siinä useampi entiteetti luo yhden kokonaisuuden. (Microsoft Azure 2017e.)

Wit.ai:ssa kehittäjän itse määrittelemien entiteettien tyypit ovat dokumentaatiossa kuvattu epäselvemmin. Käyttöliittymän ja dokumentaation perusteella entiteettejä voidaan etsiä erilaisilla tavoilla. Nämä ovat free-text, free-text and keywords sekä keywords. Free-text toimii pitkälti samoin kuin LUISin entiteettityyppi simple. Keywords taas vastaa LUISin listatyyppiä. Free-text and keywords käyttää hyväkseen listaa, mutta pyrkii tunnistamaan lausahduksista entiteettejä, jotka voisivat vastata listan arvoja. (Wit.ai 2017b; Wit.ai 2017c.) Lisäksi Wit.ai:n blogiteksti kertoo myös composite-entiteettityypin olevan mahdollinen, mutta kirjoittaja ei ehtinyt kokeilla tämän toimivuutta opinnäytetyön aikana (Wit.ai 2015).

Molemmat sovellukset pitävät kirjaa niiden rajapintojen vastaanottamista syötteistä, joiden tunnistamisessa palvelut eivät ole olleet täysin varmoja. Kehittäjä pääsee tarkastelemaan syötteitä ja voi korjata niistä virheellisesti tunnistetun aikeen tai entiteetin. Toiminnon kautta luonnollisen kielen ymmärtämisen tarkkuutta voidaan parantaa ja ongelmia voidaan korjata, kun syötteitä ollaan saatu ajoilta käyttäjiltä. LUISissa toiminnon nimi on *Suggested Utterances* ja Wit.ai:ssa *Inbox*.

Sovellukset tarjoavat myös omat toteutuksensa mallien testaamiselle. LUISissa testaus voidaan suorittaa käyttöliittymään sisällytetyn konsolin avulla. Wit.ai mahdollistaa cUrl-käskyjen generoimisen konsoliin kirjoitetuista syötteistä. Nämä käskyt voidaan ajaa esimerkiksi tietokoneen komentorivillä tai online-palveluissa. Kirjoittaja ajoi testauksessa cUrl-käskyjä <http://onlinecurl.com/>-sivuston palvelun avulla.

Palveluiden tavoissa mallin harjoittamiselle on pieniä eroja. LUISissa käyttäjän tehtyä muutoksia malliin, on harjoittaminen käynnistettävä erikseen erillisten ikkunoiden kautta. Wit.ai:ssa harjoittamista ei tarvitse käynnistää erikseen, vaan se suorittaa harjoittamista automaattisesti käyttäjän validoidessa uusia lausahduksia.

### **6.2.9 Mallien harjoittamisen ongelmat**

Kirjoittajalla ei ollut valmiina olemassa olevaa esimerkkidataa käyttäjien tekemistä kamerahauista englanniksi tai suomeksi. Esimerkkidataa kutsutaan luonnollisen kielen käsittelyn yhteydessä korpukseksi. Verkon kautta on mahdollista saada erilaisia avoimia korpuksia erityisesti englannin kielelle, mutta myös suomen kielelle on saatavilla muutamia korpuksia. Kirjoittaja ei kuitenkaan löytänyt selkeästi prototyypin ongelmaan sopivaa korpusta. Tämä voisi olla yksi jatkokehityksessä huomioitava seikka. Soveltuvan korpuksen löytyessä, sen avulla voitaisiin mahdollisesti parantaa luotavien mallien tarkkuutta ja vähentää manuaalisen työn määrää esimerkkilausahdusten osalta. Asiakaspalvelubottien tapauksessa yrityksellä olisi optimitilanteessa jo olemassa olevaa dataa, joka olisi saatu esimerkiksi yrityksen livechat- tai puhelulokeista. Tästä datasta voitaisiin pyrkiä määrittämään asiakkaiden yleisimmät ja usein toistuvat ongelmat sekä tavat, joilla asiakkaat niitä ilmaisevat.

Korpuksen puuttuessa esimerkkilausahdusten laatu, määrä ja vaihtelevuus riippuvat kehittäjän käytössä olevasta ajasta sekä siitä kuinka monipuolisia ja aitojen asiakkaiden viestintää vastaavia lausahduksia kehittäjä osaa ottaa huomioon. Il-

man olemassa olevaa korpusta voidaan toimintaperiaatteena ensin pyrkiä luomaan mahdollisimman kattava luonnollisen kielen malli suhteessa kehitykselle varattuun aikaan. Tämän jälkeen chatbot voidaan julkaista prototyypinä ja kerätä ajan kuluessa aidoilta asiakkailta saatavia lausahduksia. Tietyin väliajoin chatbot-sovelluksen ylläpitäjä käy läpi sovelluksen saamat lausahdukset ja tekee näiden perusteella malliin tarvittavat lisäykset ja korjaukset.

Toinen ongelma prototyypin luonnollisen kielen ymmärtämisen toteutuksessa oli kameran valmistajien ja mallien tunnistaminen. Valmistajat ja mallit eivät ole yleisanoja ja täten vaikeita palveluille ilman laajaa korpusta. Erityisesti kameroiden mallit ovat hankalia, sillä ne voivat koostua useista erillisistä merkkijonoista, sisältää erikoismerkkejä ja niistä ei ole löydettävissä selkeitä säännöllisiä muotoja kuin korkeintaan saman valmistajan malleissa. Testiverkkokaupassa esiintyviä malleja ovat esimerkiksi *EOS 5D Mark IV* ja *Cyper-shot DSC-HX80*.

Edellisen kaltaiset ongelmat ovat kuitenkin hyvin aihealuekohtaisia ja ratkaistavasta ongelmasta riippuvia. Esimerkiksi lentoyhtiön chatbot voi tarjota asiakkaalle tietyn lennon tietojen tarkastelemista lennon tunnuksen mukaan. Tunnus ei myöskään ole yleissana ja koostuu esimerkiksi yhdistelmästä kirjaimia ja numeroita. Tunnuksissa numerot ja kirjaimet kuitenkin esiintyvät tietyn järjestelmällisen kaavan mukaisesti ja esimerkiksi LUIS sisältää toiminnon, jossa voidaan määrittellä säännöllisien lausekkeiden mukaisia entiteettejä.

Kamerahaulle suunniteltiin mahdollisuutta rajata haettavia tuotteita hintojen perusteella. Suureksi ongelmaksi havaittiin, kuinka voidaan määrittellä tarkoittaako asiakas ilmoittaessaan hinnan hakevansa tätä halvempia vai kalliimpia tuotteita. Asiakas voi sanoa esimerkiksi: *Etsin digikameraa, joka on halvempi kuin 500 €*. Tästä voidaan siis kuvitella määrittäväksi entiteetiksi sanat *halvempi* tai *halvempi kuin*. Merkitys kuitenkin muuttuu täysin esimerkiksi syötteissä *Etsin digikameraa, joka ei saisi olla halvempi kuin 500 €* tai *Etsin digikameroita. Hinta ei halvempi kuin 500 €*. LUIS ja Wit.ai eivät tarjoa kehittäjälle vielä työkaluja, joilla tämänkaltaisen lauseen merkityksen muuttuminen kieltosanan yhteydessä voitaisiin määrittellä tai tunnistaa.

Tämän vuoksi prototyypissä jouduttiin ratkaisuna luomaan tietyt säännöt, joita käyttäjän tulisi noudattaa ilmoittaessaan hintaan perustuvia kriteerejä. Ratkaisu on kuitenkin ongelmallinen ja vie osin pohjaa koko tavoitteelta luoda luonnollisella kielellä toimivia tuotehakuja. Hakutoimintoja verkkokaupoille kehittävän Loop54:n blogitekstissä pohditaan ongelmaa ja kerrotaankin joidenkin hakukoneiden pyrkivän luomaan kieltä, joka on lähes sääntöihin pohjaavaa. Näissä ongelmaa ratkotaan siten, että hintojen rajausta toimii esimerkiksi käyttämällä sanoja *alle* (engl. *under*) tai merkillä *<*. Blogitekstin kirjoittaja pitää ongelmallisena, että tämän kaltaisen tapa kommunikaatiolle ei ole kuitenkaan kovin luontaista verkkokauppojen asiakkaille. Koska nämä säännöt eivät ole myöskään standardisoituja, joutuvat asiakkaat opettelemaan kunkin hakukoneen toimintatavat erikseen. (Odin 2017.) Ongelma on kuitenkin hankala ratkoa muulla tavalla ja kirjoittajankin opinnäytetyön aikana kohtaamissa tuotehakuja luonnollisella kielellä mahdollistavissa chatboteissa ratkaisuna oli ainoastaan sääntöihin pohjaavat tavat tuotteiden rajaamiselle hintojen perusteella.

Hintojen perusteella hakemisen mahdollistaminen kahden raja-arvon väliltä oli myös yhtä ongelmallista. Prototyypissä ratkaisuna on ainoastaan oletus, että käyttäjä tarkoittaa kahden hinnan välistä rajaa, kun sovellus tunnistaa syötteestä kaksi hintaa merkitsevää entiteettiä. Ongelmana on myös ettei LUIS- tai Wit.ai-malleja prototyypissä saatu tunnistamaan molempia hintoja, jos käyttäjä jättää hinnan perästä pois €-symbolin, sanan *euro* tai *e*-merkin. Esimerkiksi syötteestä *Hae kameroita välillä 100–500 €* saatetaan tunnistaa ainoastaan jälkimmäinen arvo, eikä chatbot ymmärrä, että käyttäjä pyrkii rajoittamaan tuotehakua kahden hinta-arvon välillä. Vastaavaa ongelmaa havaittiin myös muutamissa olemassa olevissa chatboteissa, joihin kirjoittaja tutustui opinnäytetyön aikana. Ainoa kirjoittajan keksimä ratkaisu on pyytää käyttäjää chatbotin antamassa ohjeessa kirjoittamaan numeron perään jokin edellä mainituista merkeistä tai sanoista.

Kolmas erityisesti Shopify-verkkokauppaan tehtäviä hakuja koskettanut ongelma oli, kuinka saada LUIS- ja Wit.ai-palvelujen palauttamien arvojen vastaamaan Shopify:n tuotteiden tageja, joihin prototyypin haut kohdistuivat. Jos palvelut pyritään opettamaan tunnistamaan esimerkiksi kameran malli tai tyyppi ainoastaan teko-

ällyn voimin, palauttaa se mahdollisesti oikein tunnistamansa arvon sen syötteessä esiintyvässä kirjoitusmuodossa. LUIS saattaa esimerkiksi palauttaa kameran merkin osalta entiteetin arvona sanat *Canon*, *Canon's* tai *Canons*. Suomen kielen kohdalla Wit.ai-palvelua käytettäessä ongelma on entistä suurempi sijamuotojen runsaan määrän vuoksi. Se voi palauttaa entiteetin arvona esimerkiksi *Canon*, *Canonilta*, *Canonin*, *Canonia*, *Canonista* jne.

Tämän kaltaiseen ongelmaan on olemassa jo joitain työkaluja, jotka pyrkivät poistamaan sanoista niiden sijamuodot ja palauttamaan sanan perusmuodossa. Suomen kielen kohdalla esimerkiksi Voikko voisi olla yksi kokeilemisen arvoinen työkalu, mutta kirjoittaja ei ehtinyt tutustumaan tähän tarkemmin opinnäytetyön aikana. Kirjoittaja ei myöskään löytänyt Voikolle dokumentoitua JavaScript-kirjastoa. Voikkoon voi tutustua kirjoittamisen hetkellä osoitteessa <http://voikko.puimula.org/>.

Ongelmaa pyrittiin opinnäytetyössä ratkomaan ainoastaan antamalla LUIS- ja Wit.ai-palveluille listat avainsanoista sekä lisäämällä näille manuaalisesti synonyymejä eri sijamuodoilla. Kyseinen menettelytapa ei ole missään nimessä optimaalinen ratkaisu ongelmaan. Se oli opinnäytetyön kirjoittajalle kuitenkin ainoa ratkaisu, jolla saatiin aikaan kohtuullinen prototyyppi luonnollisen kielen tuotehauista Shopify-verkkokauppa-alustalle suhteessa käytössä olleeseen aikaan. Menettelytapana se on melko työläs sekä aikaa vievä riippuen verkkokaupan tuotteiden määrästä. Ratkaisu ei ole kuitenkaan täysin virheellinen mutta soveltuu parhaiten tapauksiin, joissa verkkokaupan tuotteiden määrä on melko pieni, ja kun tiedetään, etteivät verkkokaupan tuotteet vaihdu useasti. Testiverkkokaupan 30 tuotteen kohdalla se oli työmäärältään ja sen vaatimalta ajalta vielä kohtuullinen ratkaisu.

### **6.2.10 Prototyypille luodut mallit**

Mallien luominen oli kirjoittajalle uusi oppimiskokemus ja sitä oli hankala suunnitella kattavasti etukäteen muutoin kuin luotavien aikeiden ja entiteettien osalta.

Kirjoittajan mallien luominen oli sarja yrityksiä ja erehdyksiä kunnes lopulta päästiin prototyyppin lopputulokseen. Malleista luotiin kehittämisen aikana useita versioita ja kirjoittaja kokeili ratkoa ongelmia erityisesti entiteettien kanssa useilla eri keinoilla. Monet suurista esiin tulleista ongelmista, jotka pakottivat kirjoittajan pohtimaan aiemmista suunnitelmista poikkeavia ratkaisuja, kuvattiin jo edellisessä luvussa. Tässä luvussa kuvataan ainoastaan lopullisen toteutuksen rakenne sekä huomioidaan ongelmia, jotka eivät esiintyneet edellisessä luvussa.

Englanninkieliselle toteutukselle luotiin LUIS-palveluun sovellus *Product Search*. Sovellukseen luotiin neljä aietta. *Greeting* pyrkii tunnistamaan tervehdykset. Se sisältää 13 esimerkkilausahdusta englanninkielisistä tervehdyksistä. *Utilities.help*-aie pyrkii tunnistamaan käyttäjän pyynnöt avulle tai tarpeen ohjeelle. Tämä aie on alun perin yksi LUISin esirakennetuista aikeista ja sisältää LUISin alustavat kymmenen lausahdusta sekä 12 kirjoittajan lisäämää lausahdusta. *None*-aie sisältää kymmenen sekalaista lausahdusta, jotka eivät liity chatbotin muihin toimintoihin. *SearchCameras*-sisältää 216 lausahdusta, jotka mallintavat kamerahakuja.

Ainoastaan *SearchCameras*-aie sisältää lausahduksia, joissa esiintyy entiteettejä. Entiteettejä on viisi kappaletta. Kirjoittaja ei saanut LUISista tietoja, joka kertoisi kuinka monessa lausahduksessa kukin entiteetti esiintyy.

*CameraBrand*-entiteetti on tyypiltään lista, jossa avainsanoina ovat yhdeksän testiverkkokaupassa esiintyvää kameran valmistajaa. Kullekin avainsanalle on annettu synonyyminä valmistajan nimi s-päätteellä. Esimerkiksi avainsanan *Canon* synonyyminä on *Canons*. Antamalla LUISille avainsanana valmistajan nimi perusmuodossa, se osaa tunnistaa tämän myös muodoista, jotka sisältävät heitomerkin esim. *Canon's* ja palauttaa ainoastaan arvon *Canon*.

*CameraModels*-entiteetti on tyypiltään lista, jossa avainsanoina ovat testiverkkokaupassa esiintyvien mallien nimien yksittäiset merkkijonot. Esimerkiksi mallin nimi *EOS 5D Mark IV* tuottaa neljä avainsanaa. Pilkkomalla mallin nimen merkkijonot, on käyttäjän mahdollista hakea malleja, vaikka mallin nimestä ei muistet-



taisi kuin vain yksi tai muutama merkkijono. Kirjoitusvirheen kohdalla tunnistamista ei kuitenkaan tapahdu. Testiverkkokaupan 30 tuotetta tuottivat mallien avainsanoja 53 kappaletta ja näille ei määritelty synonyymejä.

*CameraType*-entiteetti on tyypiltään lista, jossa avainsanoina ovat *digital*, *dslr* ja *action*, mitkä vastaavat testiverkkokaupan kameroiden eri kategorioita. LUIS erottaa nämä avainsanat, jos ne on kirjoitettu yksittäisenä sanana tai niitä seuraa merkki - tai välilyönti. Esimerkiksi muotoja *digital cameras* ja *digital-camera* ei tarvitse lisätä synonyymeihin. Jos sanat on kirjoitettu yhteen esim. *digitalcameras*, ei LUIS kuitenkaan tunnista sanaa *digital*. Tämän vuoksi synonyymeihin oli lisättävä kaikki kirjoittajalle mieleen tulleet tavat, joilla käyttäjä voi kirjoittaa kameran tyyppiä yhdessä yhteen kirjoitetussa sanassa. Lisäksi synonyymeissä on otettu huomioon, kun käyttäjä kirjoittaa sanan *digital* muodossa *digi* tai sanan *cameras* muodossa *cams*. Sama kaava toistuu myös *action*- ja *dslr*-sanojen yhteydessä. Näille kolmelle avainsanalle luotiin yhteensä 18 synonyymia.

Koska kirjoittajalla ei ole asiantuntemusta kameroiden ja kuvaamisen yhteydessä käytetystä alan sanastosta, oli synonyymien pohtiminen erityisesti englanniksi erittäin haastavaa. Kirjoittaja teki parhaansa pyrkiessään saamaan alustavan listan mahdollisista kameratyyppien synonyymeistä. Aitojen asiakkaiden käyttämien erilaisten synonyymien löytämiseksi tarvittaisiin ehdottomasti olemassa olevaa korpusta tai kamera-alan asiantuntijan tukea.

*PriceDefinition*-entiteetti on tyypiltään lista, jonka avainsanat ovat *min* ja *max*. Prototyypissä käyttäjälle joudutaan määrittämään ainoastaan muutama sallittu sana, jolla käyttäjä voi kertoa halutaanko tuotteita etsiä alle tai yli tietyn hinnan. *Min*-sana saa synonyymikseen sanan *under* ja *max*-sana saa synonyymikseen sanan *over*.

*Money*-entiteetti on LUISin esirakentama entiteetti, joka pyrkii tunnistamaan raha-arvot. Tämän toimintaan kehittäjä ei pysty itse vaikuttamaan. Kirjoittajaa jää vaivaamaan, ettei esirakennettu entiteetti kuitenkaan tunnista useasti pikavies-tinnässä käytettyä tapaa kirjoittaa *euro*-sanan tai €-symbolin sijasta ainoastaan kirjain e. Numeroita ei myöskään tunnisteta, jos käyttäjä kirjoittaa ne sanallisessa

muodossa ja hinta-arvon määrä saatetaan myös tunnistaa virheellisesti, jos käyttäjä lisää numeroiden väliin pisteitä, pilkkuja tai tyhjiä välejä.

Wit.ai:n malli suomen kielelle sisältää yhtä monta aietta kuin LUIS-malli. Aikeiden nimet ovat kirjoitettu englanniksi ja vastaavassa muodossa kuin LUIS-mallissa. Ainoa poikkeus on *Help*-aie, jonka nimi on LUISissa *Utilities.help*. Myös aikeiden tarkoitukset ovat vastaavat kuin LUISille luodussa mallissa. Wit.ai:n malli sisältää yhteensä 191 esimerkkilausahdusta. Näistä *Greeting*-aie sisältää 21 lausahdusta, *Help*-aie 11 lausahdusta, *None*-aie 6 lausahdusta ja *SearchCameras*-aie 153 lausahdusta.

Entiteettejä on LUISille luodusta mallista poiketen ainoastaan neljä, sillä tuotteiden hakua mallin mukaan ei toteutettu suomenkielisessä versiossa. Päätös tehtiin keskustelussa ohjaajan kanssa. Toiminnon toteutus olisi ollut täysin vastaava kuin LUIS-mallissa ja aikataulullisten ongelmien vuoksi toteutuksen toistolle ei nähty välttämätöntä tarvetta. Entiteetti *price* vastaa LUISin entiteettiä *money*. Muutoin entiteettien nimet ja tarkoitukset vastaavat LUIS-mallin entiteettejä.

*Camerabrand*-entiteetti vastaa pitkälti LUIS-mallin vastaavaa entiteettiä. Sen avainsanoja ovat yhdeksän valmistajan nimet. Synonyymeiksi annettiin valmistajien nimien genetiivi- ja ablatiivimuodot, jotka kirjoittaja arvioi yleisiksi suomenkielisissä hakulausekkeissa. Esimerkiksi valmistaja *Canon* saa synonyymeikseen muodot *Canonin* sekä *Canonilta*. Valmistaja *Epic* saa myös kaksi eri ablatiivimuotoa, jotka ovat *Epiciltä* ja *Epicilta*. Kirjoittaja arvioi, että käyttäjät voisivat mahdollisesti kirjoittaa sanan molemmissa muodoissa.

*CameraType*-entiteetin kohdalla ongelmat olivat LUIS-mallin kohdalla esiintyviä ongelmia vastaavat. Nämä olivat synonyymien runsas määrä ja kirjoittajan puutteellinen asiantuntemus kamera-alan sanastosta. Ongelmaa kasvattivat edelleen suomen kielen useat sijamuodot. Avainsanat ovat englanniksi ja vastaavat kameroiden kategorioita. Näille avainsanoille pyrittiin löytämään suomenkielisiä avainsanan vastineita sekä synonyymejä.

Avainsana *dslr* saa synonyymeikseen muun muassa sanat *järjestelmä*, *järjestelmäkamera*, *järkkäri* ja *järkkärikamera*. *Digital*-avainsana saa synonyymeikseen *digi*, *digitaali*, *digitaalikamera* ja *digikamera*. *Action*-kameroille ei kirjoittaja löytänyt muita synonyymejä kuin sanan *actionkamera*. Kaikki synonyymit paitsi *digi*, *digitaali* ja *järjestelmä* saavat myös viisi eri sijamuotoa, jotka ovat nominatiivin yksikkö- ja monikkomuoto, genetiivi sekä partitiivin yksikkö ja monikkomuoto. Näin esimerkiksi synonyymi *digitaalikamera* saa muodot *digitaalikamera*, *digitaalikamerat*, *digitaalikameran*, *digitaalikameraa* ja *digitaalikameroita*. Lisäksi sanasta *kamera* kirjoitettiin kaikissa tapauksissa muoto *kameroja*.

Synonyymien ei tässäkään tapauksessa voida arvioida kattavan läheskään kaikkia tapoja ilmoittaa haettavan kameran tyyppi, mutta nämä muodot voisivat kirjoittajan arvion mukaan olla yleisiä suomenkielisissä hakulauseissa. Lopulta kolme avainsanaa vastaavia synonyymejä on yhteensä 39 kappaletta ja tarve työkalulle, joka osaisi karsia sanoista sijamuotoja tulee selkeästi esille.

Wit.ai-mallin *priceDefinition*-entiteetti toimii vastaavasti kuin LUIS-mallissa. Se on tyyppiltään lista ja saa avainsanoikseen sanat *min* ja *max*. Englannin kielen synonyymit korvataan suomen kielen sanoilla alle ja yli.

Wit.ai ei tarjonnut suomen kielelle valmiita entiteettejä numeroiden tunnistamiseen. Tämä ongelma ennakoitiin jo ennen opinnäytetyön aloittamista. Kirjoittaja loi tarkoitusta varten uuden *price*-entiteetin ja mallia pyrittiin opettamaan manuaalisesti tunnistamaan rahan arvoja. Lausahduksista pyrittiin esimerkiksi maalamaan hiirellä ainoastaan numero-osa, jättämällä pois *e*-merkki, *€*-symboli tai *euro(a)*-sana. Maalaustoiminto ei kuitenkaan prototyypin toteutuksessa tuottanut toivottua tulosta.

Prototyypin viimeisessä versiossa *price*-entiteetille on esitetty 96 lausahdusta, joilla malli alkoi lopulta tunnistaa numeroita kohtalaisella tarkkuudella. Se kuitenkin edelleen sisällyttää palautettavaan arvoon myös numeroa seuraavia merkkejä. Chatbotin Search.js-tiedostoon luotu merkkijonoista ainoastaan numerot palauttava funktio onnistuu kuitenkin karsimaan nämä erikoismerkit pois. Wit.ai

ei myöskään LUISin tapaan tunnista numeroita, jos ne kirjoitetaan sanana tai numeroarvojen keskellä käytetään pisteitä, pilkkuja tai välejä.

### 6.2.11 Chatbotin hakudialogi

Chatbotin hakudialogi laadittiin erilliseen `Search.js`-tiedostoon. Tiedoston sisältämä hakudialogi käynnistetään, kun NLU-malli palauttaa sovellukselle todennäköisimpänä aikeena nimen *SearchCameras*.

Bot Builder välittää LUISin tai Wit.ai:n käyttäjän syötteestä tunnistaman aikeen sekä entiteetit hakudialogille *args*-parametrin mukana. Hakudialogi pyrkii poimimaan tunnistetut entiteetit käyttämällä Bot Builderin `EntityRecognizer`-luokan metodia *findEntity* tai metodia *findAllEntities*, kun voidaan odottaa, että samaa entiteettiä löytyy syötteestä useampi kappale. Prototyypissä näin tapahtuu esimerkiksi mallin nimien eri osien tai hintojen kohdalla. LUIS ja Wit.ai palauttavat joidenkin entiteettien kohdalla niiden sisältämät arvot hieman poikkeavalla tietorakenteella. Tämän vuoksi tiettyjen entiteettien kohdalla joudutaan tarkastamaan ennen entiteettien poimimista kumpi palveluista on kyseisellä hetkellä sovelluksen käytössä.

Tunnistetut entiteetit talletetaan Bot Builderin tarjoamaan *conversationData*-tietorakenteeseen, johon luodaan uusi objekti nimeltä *SearchParameters*. *ConversationData*-tietorakenteeseen talletetut tiedot ovat pysyviä koko keskustelun ajan, ellei niitä ohjelmakoodin aikana muuteta tai alusteta uudelleen. Dialogissa talletetut tiedot olisivat siis käytettävissä myös chatbotin muissa dialogeissa. Prototyypin kohdalla tietojen tallettaminen *conversationData*-tietorakenteeseen ei ole täysin välttämätöntä, koska tietoja hyödynnetään vain hakudialogin sisällä. Talletus kuitenkin tehtiin mahdollista jatkokehitystä ajatellen.

Lisäksi käyttäjän syötteestä tunnistetut kameran merkkiä, mallia tai tyyppiä merkitsevät entiteetit lisätään ennen hakua taulukkotietorakenteeseen nimeltään *tags*, joka annetaan myöhemmin ohjelmakoodissa parametrina haut Shopifylle

tekeville funktioille. Kameran mallin nimen eri merkkijono-osat talletetaan tauluk-  
koon omina alkioinaan. Kameran tyyppiä merkitsevät entiteetit ovat arvoltaan  
joko *digital*, *action* tai *dslr*. Näiden arvojen perään lisätään sana *camera*, jotta  
merkkijono lopulta vastaa verkkokaupan tuotteille lisättyjä tageja.

Käyttäjän syöttämien hintojen kohdalla tarkastellaan onko syötteestä tunnistettu  
entiteetti, joka kuvaa tarkoittaako käyttäjä haun kohdistamista annetun hinnan yli  
tai alle. Mikäli tätä tietoa ei löydetä, määrittelee chatbot oletuksena haun kohdis-  
tuvan ilmoitetun hinnan alle. Jos taas käyttäjän syötteestä on tunnistettu kaksi  
hintaa merkitsevää entiteettiä, olettaa chatbot käyttäjän haluavan rajata hakua  
kahden hinnan väliltä. Tunnistettaessa kaksi hintaa merkitsevää entiteettiä, ver-  
rataan kahden arvon suuruutta, jotta näistä löydetään hintojen ala- ja ylärajaa  
merkitsevät luvut. Tunnistetut hinnat talletetaan ainoastaan *conversationData*-tie-  
torakenteen *searchParameters*-objektiin.

Prototyypin chatbot tulostaa tunnistetut hakukriteerit käyttäjälle ennen haun suo-  
rittamista. Näin käyttäjä voi mahdollisen virheellisen hakutuloksen kohdalla pyrkiä  
itse tarkastamaan, mikä tai mitkä syötetyistä hakukriteereistä ovat mahdollisesti  
jääneet chatbotilta tunnistamatta tai tunnistettu virheellisesti.

*Tags*-taulukko lähetetään lopulta parametrina haun tekeville funktioille, jotka on  
määritelty sovelluksen *Shopify.js*-tiedostossa. Lisäksi hakufunktioille annetaan  
parametrina callback-funktio, joka suoritetaan, kun hakufunktiot ovat palauttaneet  
*Shopify*sta saadut hakutulokset.

Callback-funktion vastaanottaessa hakutulokset JSON-muodossa, tehdään pa-  
lautetuille tuotteille ensin suodatus hintojen perusteella, mikäli käyttäjä on ha-  
kusyötteessä esittänyt tälle rajoituksia. Lopulliset hakusyötettä vastaavat tuotteet  
talletetaan uuteen taulukkoon, jonka alkioista luodaan käyttäjälle tulostettavat  
tuotteita esittävät kortit.

Kortit luodaan Bot Builderin tarjoamaa *HeroCard*-luokkaa hyödyntäen. Luokan  
avulla voidaan muodostaa chat-ikkunaan tulostettavia kortteja, jotka voivat sisäl-  
tää otsikoita, aliotsikoita, tekstiä, kuvia sekä painikkeita. Prototyypissä otsikoksi

asetetaan löytyneiden tuotteiden otsikot, tekstiksi tuotteiden hinta ja kuvaksi tuotteen oletuskuva. Lisäksi kortteihin lisätään kaksi painiketta. Näistä ensimmäinen mahdollistaa tuotteen url:in avaamisen selaimessa ja toinen käynnistää uuden dialogin *ShowMore*, joka tulostaa käyttäjälle valitusta tuotteesta uuden lisätietoja sisältävän kortin.

Tuotteen url:in avaamiseksi selaimessa hyödynnetään BotBuilderin CardAction-luokan metodia *openUrl*. Tämä saa yhtenä parametrina selaimessa avattavan verkkosivun osoitteen. Osoite luodaan tässä tapauksessa verkkokaupan tuotteiden juuri-url:ista sekä lisäämällä tämän perään kyseisen tuotteen handle.

Lisätietoja näytävä painike luodaan CardAction-luokan metodilla *dialogAction*. Tämä saa parametreinä esimerkiksi käynnistettävän dialogin osoitteen sekä käynnistettävälle dialogille osoitetut argumentit. Tässä tapauksessa käynnistettävälle dialogille annetaan argumenttina kunkin tuotteen id:n.

*ShowMore*-dialogin toiminnallisuus on määritelty prototyypissä omassa tiedostossaan. Dialogi kutsuu Shopify.js-tiedostoon määriteltyä hakufunktiota, joka hakee yksittäisen tuotteen sen id:n perusteella. Palautetusta tuotteesta luodaan jälleen uusi kortti hyödyntämällä Bot Builderin HeroCard-luokkaa. Kortti sisältää aikaisemman kortin tietojen lisäksi tuotteen kaikki kuvat sekä tuotteen kuvauksen. Shopify palauttaa tuotteen kuvauksen html-muodossa. Tämä html-muodossa oleva kuvaus muunnetaan Bot Builderin tukemaan markdown-muotoon, hyödyntämällä tähän erillistä to-markdown-kirjastoa. Kirjasto asennettiin sovellukseen npm-paketinhallinnan kautta ja kirjasto sekä sen dokumentaatio löytyvät kirjoittamisen hetkellä osoitteesta <https://github.com/domchristie/to-markdown>.

Luodut kortit lisätään prototyypissä uuteen taulukkoon. Kun taulukko on luotu, luodaan prototyypissä uusi instanssi Bot Builderin Message-luokasta. Tämä on tarkoitettu monipuolisempien viestin rakentamiseen. Prototyypissä viestin layoutiksi asetetaan karuselli ja viestin liitteeksi lista luoduista korteista. Näin käyttäjä saa chat-ikkunaan viestinä listan löytyneistä tuotteista kortteina karuselimuodossa. Mikäli haku ei löytänyt tuotteita, tulostetaan tieto tästä käyttäjälle.

Search.js-tiedosto sisältää haut mahdollistavan dialogin lisäksi myös kaksi apufunktiota. Ensimmäinen funktio *capitalizeFirstLetter* palauttaa sille parametrina annetun merkkijonon isolla alkukirjaimella. LUIS- ja Wit.ai-palvelut palauttavat entiteettien arvot lähes aina muodossa, jossa kaikki merkkijonon kirjaimet ovat kirjoitettu pienillä kirjaimilla. Funktiota käytetään tulostettaessa tunnistettuja kriteerejä, jotta valmistajan nimi saadaan kirjoitettua isolla alkukirjaimella.

Toinen funktio palauttaa sille parametrina annetusta merkkijonosta numerot. LUIS ja Wit.ai palauttavat tunnistetut numeroarvot sovellukselle merkkijonoina ja erityisesti Wit.ai mallin palauttamat *hint*a-entiteetit saattavat sisältää muita merkkejä kuin numeroita. Kun käyttäjä haluaa hakea tuotteita kahden hinnan väliltä, kirjoittaja olettaa, että käyttäjä usein saattaa kirjoittaa hinnat esimerkiksi muodossa *100 €–500 €*. Niin LUIS kuin Wit.ai:kin saattavat tässä tapauksessa tulkita välissä sijaitsevan viivan osaksi hinta-arvoa, jolloin ne palauttavat arvon *-500 €*. Yksi ratkaisu olisi väliviivan määrittelemisen omaksi entiteetikseen, jolloin mallit eivät sekoita sitä kuuluvaksi numeroarvolle. Tämä ei kuitenkaan ratkaisuna ole kovin käytännöllinen. Apufunktio kuitenkin onnistuu ratkaisemaan ongelman poistamalla merkkijonosta kaiken muun paitsi numerot ennen merkkijonon sisältämän arvon muuntamista numero-tietotyypisiin.

### 6.2.12 Shopify-hakufunktiot

Haut Shopify-verkkokauppa-alustalle tekevät funktiot sijoitettiin prototyypissä omaan tiedostoonsa nimeltä *Shopify.js*. Tiedosto käyttää hauissa hyväkseen aiemmin kuvattua Shopifyn JavaScript Buy SDK:ta. Tämä asennettiin npm-paketinhallinnan kautta komennolla *npm install -save shopify-buy* ja otettiin käyttöön ohjelmakoodissa käskyllä *const shopifyBuy = require('shopify-buy');*. Yhdistettävän verkkokaupan tiedot asetetaan luomalla instanssi Buy SDK:n *ShopClient*-luokasta *buildClient*-metodilla. Tarvittavat tiedot ovat verkkokaupan *accessToken*, *domain* sekä *appId*. Nämä tiedot saatiin siirtymällä Shopifyn Partners Dashboardilla kyseessä olevan testiverkkokaupan sovellusten hallintaan.

Tiedostoon luotiin kaksi erilaista hakufunktiota. *GetProductsByTag*-funktio ottaa vastaan parametreina taulukon tageja sekä callback-funktion, joka suoritetaan, kun funktio on vastaanottanut palautetut hakutulokset. Tagit sisältävä taulukko asetetaan parametriksi Buy SDK:n *fetchQueryProducts*-metodille, joka palauttaa tageja vastaavat tuotteet JSON-muodossa. JSON-muodossa palautetut tuotteet annetaan jälleen parametrina callback-funktiolle, jonka toiminnallisuus on prototyypissä määritetty aiemmin esitellyssä hakudialogissa.

Toinen funktio *GetProductsById* toimii vastaavasti, mutta vastaanottaa tageja sisältävän taulukon sijasta merkkijonona tuotteen id:n. Prototyyppi käyttää funktiota hakiessaan lisätietoja yksittäisestä tuotteesta. Vastaanotettu id asetetaan parametriksi Buy SDK:n *fetchProduct*-metodille, joka palauttaa ainoastaan yhden id:tä vastaavan tuotteen JSON-muodossa. Palautettu tuote asetetaan jälleen parametriksi callback-funktiolle, jonka toiminnallisuus on prototyypin kohdalla määritetty tiedostossa *ShowMore.js*.

### 6.2.13 Lokalisointi

Prototyypin eri versiot suomen ja englannin kielelle hyödyntävät Shopify-verkko-kauppaan tehtäville hauille samaa hakudialogia. Koska hakudialogin toiminnallisuudessa käyttäjälle tulostetaan viestejä kuten esimerkiksi käyttäjän syötteestä tunnistetut hakukriteerit, jouduttiin dialogin tulosteiden sisältämät merkkijonot lokalisoimaan.

Lokalisointia varten lähdekoodiin lisättiin hakemisto *locale*. Tämän sisälle luotiin kaksi hakemistoa, joiden nimet vastaavat standardisoituja IETF-kielikodeja. Prototyypin tapauksessa kielikoodit ovat *en* ja *fi*. Näiden molempien hakemistojen sisälle luotiin *index.json*-nimiset tiedostot. Tiedoston sisälle määritellään eri tulosteiden id:t ja niitä vastaavan tulosteen sisältö kyseessä olevalla kielellä.

Prototyypin eri versioissa *App.js*-tiedostosta *UniversalBotin* konstruktorissa määritellään sovellukselle lokalisoititiedostojen hakemiston juuripolku sekä sovelluk-



sen oletuskieli. Oletuskielen asettaminen konstruktorissa, ei kuitenkaan prototyypin kohdalla toiminut odotetulla tavalla, mahdollisesti johtuen prototyypin toimimisesta ainoastaan emulaattorissa. Tämän vuoksi prototyypissä käytettävä kieli asetetaan hakudialogin käynnistyessä `session.preferredLocale`-metodilla, joka saa parametrikseen IEFT-kielikoodin. Kielen määrittäminen olisi chatbotin kohdalla manuaalisen määrittämisen lisäksi mahdollista toteuttaa kysymällä haluttua kieltä käyttäjältä tai esimerkiksi hyödyntämällä Azuren Text Analytics API:a kielen automaattiseen tunnistamiseen käyttäjän syötteiden perusteella.

Hakudialogissa halutun kielen tuloste saadaan käyttöön antamalla `session.send`-metodille lokalisointitiedostoissa käytetty tulosteen id. Jos tulosteeseen halutaan liittää muuta sisältöä kuten esimerkiksi muuttujien arvoja, joudutaan tähän käyttämään `session.localizer.getText`-metodia, jolle annetaan parametrina IEFT-kielikoodi sekä tulosteen id.

## 7 Tulokset ja johtopäätökset

Tässä luvussa arvioidaan toiminnallisen opinnäytetyön aikana saavutettuja tuloksia. Opinnäytetyön teoreettisessa taustassa käsiteltiin yleisesti tekoälyn hyödyntämistä liiketoiminnan ja erityisesti yritysten asiakaspalvelun tukena, chatboteja sekä luonnollisen kielen käsittelyä. Tässä luvussa pyritään kuitenkin arvioimaan opinnäytetyön käytännön osuuden tuloksia, peilaten niitä opinnäytetyölle asetettuihin tutkimuskysymyksiin- ja ongelmiin.

Ensimmäisenä tarkastellaan saatiinko opinnäytetyön aikana toteutettua siihen valituilla työkaluilla prototyyppi asiakaspalveluun suuntautuvasta chatbotista, joka tekee tuotehakuja Shopify-verkkokauppa-alustaan luonnollisella kielellä esitettyjen hakukyselyjen välityksellä. Toisena ongelmana käsitellään havaittuja eroja prototyypin suomen ja englannin kielten versioiden toteutuksessa valittujen työkalujen osalta. Kolmantena kirjoittaja arvioi prototyypin hakutoiminnon kattavuutta ja onnistumista. Lopussa pohditaan myös mahdollisia ideoita prototyypin jatkokehitykselle.

## 7.1 Prototyypin valmistuminen ja arviointi

Opinnäytetyön aikana saatiin aikaiseksi prototyyppi asiakaspalveluun suuntautuvasta chatbotista. Valmistuneessa prototyypissä chatbotin toimintoina olivat käyttäjän tervehdyksiin vastaaminen, ohjeiden esittäminen sekä tuotehakujen tekeminen Shopify-verkkokauppa-alustalla toimineen testiverkkokaupan tuotteille. Käyttäjän on mahdollista esittää hakukriteerit luonnollisella kielellä joko suomeksi tai englanniksi. Chatbot tulostaa käyttäjälle tunnistetut hakukriteerit ja esittää käyttäjälle hakukriteereillä löytyneet tuotteet. Chatbotin lähettämät viestit sekä tulosteet esitetään suomen tai englannin kielellä valitusta versiosta riippuen. Lisäksi prototyypin yhteydessä toteutettiin kameroita myyvä testiverkkokauppa Shopify-verkkokauppa-alustalle.

Prototyypissä toteutettu chatbot, NLU-palveluille laaditut luonnollisen kielen mallit, testiverkkokauppa ja sovelluksen toiminta esiteltiin opinnäytetyön ohjaajalle ja toimeksiantajalle videoneuvottelussa 2.11.2017. Opinnäytetyön ohjaaja ja toimeksiantaja antoivat hyväksyntänsä tuloksille ja arvioivat prototyypin täyttävän sille asetetut vaatimukset ja odotukset.

Toteutukseen valitut työkalut osoittautuivat käyttökelpoisiksi luotaessa asiakaspalveluun tarkoitettuja chatboteja sekä lisättäessä chatbotin toiminnallisuuteen kyky luonnollisen kielen ymmärtämiseen. Microsoftin Bot Framework ja siihen sisällytetty Bot Builder SDK tarjosivat kirjoittajalle tarpeeksi erilaisia työkaluja, joilla chatbotin toiminnallisuus oli mahdollista toteuttaa. LUISin ja Wit.ai:n avulla toteutetun luonnollisen kielen ymmärtämisen kehityksessä ja niiden avulla saavutetuissa lopputuloksissa esiintyi eroja, mutta pääasiassa molempien työkalujen avulla päästiin lopulta melko yhteneväiseen ja toisiaan vastaavaan lopputulokseen.

Työkaluja arvioitaessa tulee kuitenkin huomioida, että prototyypin tapauksessa kyseessä oli yksi yksittäinen käyttötapaus ja asiakaspalveluksi luokiteltavat tehtävät sekä toiminnot voivat chatbotien tapauksessa olla hyvinkin erilaisia ja monipuolisia. Työkalujen soveltuvuutta eri tilanteisiin ja käyttötapauksiin ei voida täysin arvioida pelkästään opinnäytetyössä toteutetun prototyypin avulla.

Lisäksi on mainittava, ettei prototyypin toimintaa päästy vielä testaamaan suuremmalla joukolla testikäyttäjiä tai aitoja asiakkaita, joten luonnollisen kielen ymmärtämistä varten luotujen mallien toimivuutta aidoissa tilanteissa on lähes mahdotonta arvioida. Chatbotien kyky luonnollisen kielen ymmärtämiseen voidaan myös nähdä jatkumona, jossa mallien toimivuutta ja niiden tarkkuutta voidaan kehittää edelleen chatbotin julkaisemisen jälkeen aidoista käyttötilanteista saata-vaan dataan perustuen.

## 7.2 Kielten väliset erot prototyypissä

Suomen ja englannin kielen väliset erot tulivat selkeimmin esille LUIS- ja Wit.ai-palveluiden palauttamien entiteettien arvojen kohdalla. Wit.ai:n kyyvyssä tunnistaa suomenkielisiä entiteettejä ei havaittu suurta ongelmaa, mutta palvelun lopulta palauttaman tunnistetun entiteetin merkkijono voi olla hyvin monessa erilaisessa muodossa johtuen suomen kielen eri sijamuotojen runsaasta lukumäärästä. Myös englannin kielen sanojen havaittiin olevan mahdollisesti eri muodoissa, mutta muotojen määrä on kuitenkin selkeästi rajoitetumpi.

Tilanne on ongelmallinen, kun tunnistetun sanan muoto on olennaista sovelluksen toiminnallisuudelle. Prototyypin tapauksessa ongelmana oli saada sanat vastaamaan Shopify-verkkokaupassa tuotteiden tageiksi asetettuja sanoja. Lopulta prototyypissä ratkaisuna hyödynnettiin molempien palvelujen tarjoamaa mahdollisuutta luoda listoja avainsanoista sekä näihin liitettävistä synonyymeistä. Tämän kaltaisten listojen toteutus vaatii kuitenkin runsaasti manuaalista työtä kehittäjältä sekä käyttäjien syötteissä mahdollisesti esiintyvien sanojen ja niiden sijamuotojen tarkkaa arviointia sekä suunnittelua ennalta. Suomen kielen kohdalla työmäärä kasvaa selkeästi verrattuna englannin kieleen. Tuotteiden tai tuotekategorioiden määrän kasvaessa tarvitaan manuaalisen sijamuotojen määrittämisen korvaamiseksi työkalua, joka pysyisi erottamaan sanoista perusmuodon ja karsimaan sijamuodot pois.

Työkalujen eroja suomen- ja englanninkielisten aikeiden ja entiteettien tunnistamisessa ei opinnäytetyön aikana ehditty mittaamaan luotettavasti. Kirjoittajan

pyrkiessä arvioimaan tunnistamisen tarkkuutta kriittisesti opinnäytetyön aikana saatujen kokemusten perusteella, ei tarkkuuksissa määrältään pienien esimerkkisyötteiden osalta kyetty havaitsemaan huomattavaa eroa. Käyttäjän tervehdystä ja ohjeiden pyytämistä tunnistavien aikeiden kohdalla Wit.ai vaikutti suomen kielen kohdalla toimivan jopa hieman paremmin. Wit.ai pystyi tunnistamaan aikeen usein oikein, vaikka käyttäjän syöte ei sisältänyt ainuttakaan mallin esimerkkilausehduksissa esiintyvää sanaa vastaavassa muodossa. LUISin kohdalla aikeen tunnistamisen tarkkuus oli vaihtelevampaa, vaikka sille annettiin hieman enemmän esimerkkilausehduksia. Tulosta ei kuitenkaan voida pitää luotettavana, sillä tarkkuuksia ei ennätetty mitata luotettavalla menetelmällä. Lisäksi asiaan saattaa vaikuttaa kirjoittajan malleille esittämien esimerkkilausehduksen monipuolisuus ja laatu, joiden vertaaminen kahden eri kielen välillä on erittäin hankalaa

Yksi havaittu ero luotaessa luonnollisen kielen ymmärtämistä suomen kielelle Wit.ai:lla ja englannin kielelle LUISilla on niiden tarjoamien esirakennettujen aikeiden ja entiteettien määrässä sekä esirakennettujen entiteettien tyypeissä. Wit.ai ei tarjoa suomen kielelle vielä lainkaan esirakennettuja aikeita, kun taas LUIS tarjoaa niitä englannin kielelle useampia. Mikäli käyttötapauksessa voidaan hyödyntää jotakin LUISin tarjoamaa esirakennettua aietta, voidaan tällä saada kehittämiseen tarvittavassa ajankäytössä pientä ajallista etumatkaa verrattuna suomen kielen mallien kehittämiseen Wit.ai-palvelulla.

Molemmat palvelut tarjoavat lähes yhtä suuren määrän esirakennettuja entiteettejä, mutta niiden tyypit eroavat toisistaan. Prototyypissä pystyttiin esimerkiksi hyödyntämään LUISin esirakennettua entiteettiä raha-arvojen tunnistamiseen, kun taas Wit.ai:ssa jouduttiin itse harjoittamaan mallia niiden tunnistamiseksi. Tämä vaikutti selkeästi eri mallien kehitykseen käytettyyn aikaan.

### 7.3 Hakutoiminnon kattavuus

Prototyypitoteutuksessa luodun chatbotin sisältämän luonnollisen kielen hakutoiminnon kattavuutta on erittäin hankala arvioida tarkasti ja luotettavasti. Jo luonnollisen kielen ymmärtämisen tason määrittelemisen voidaan nähdä olevan monimutkainen ongelma. Nykypäivän algoritmit eivät vielä kykene ymmärtämään ihmisten tuottamaa kieltä sen kaikessa monipuolisuudessaan täydellisesti. Teknologiajättien suurilla resursseilla luodut virtuaaliassistentit kuten esimerkiksi Applen Siri tai Amazonin Alexa pyrkivät nykypäivänä ymmärtämään luonnollista kieltä, mutta myös ne joutuvat asettamaan käyttäjiensä ilmaisumuodoille rajoituksia. Kirjoittaja kutsuisisikin tämän hetkisiä pyrkimyksiä mieluummin illusioiden luomiseksi luonnollisen kielen ymmärtämisestä.

Itse hakutoiminto saadaan prototyypissä toimimaan melko luotettavaksi, jos käyttäjän syötteestä pystytään tunnistamaan aie haulle sekä hakua rajaavat kriteerit oikein. Kaikkiin eri tapoihin, joilla käyttäjät ilmaisevat aikeensa tuotehauille tai esittävät hakukriteerit ei millään kyetä varautumaan. Kirjoittaja pyrki kuitenkin prototyypissä luomaan hakutoiminnon, joka tunnistaisi monia erilaisia hakua ilmaisevia lauseita sekä sanamuotoja eri hakukriteerien esittämiselle. Mallin toimintaa ei kuitenkaan opinnäytetyön aikana testattu ulkopuolisilla käyttäjillä, joten toiminnon kattavuudesta ei tätä kautta saatu tilastollista tai mitattavaa tietoa.

Ohjaajalta ja toimeksiantajalta saadun suullisen palautteen mukaan esitelty prototyyppi mahdollistaa haut monipuolisilla ja erilaisilla hakulauseilla molempien kielten kohdalla. Prototyypissä joudutaan kuitenkin rajoittamaan käyttäjän syöttämiä hakulauseita esimerkiksi hintojen kirjoitusmuodossa sekä tavassa ilmaista haetaanko tuotteita alle vai yli ilmoitetun hinta-arvon. Rajoitteet kuitenkin ilmoitetaan käyttäjälle chatbotin tarjoamassa ohjeessa ja ohjeet sisältävät myös esimerkkejä testatuista hakulauseista, joilla haun tulisi onnistua.

Ohjaaja ja toimeksiantaja arvioivat kuvattujen rajoitteiden olevan odotettuja prototyypin kohdalla erityisesti suhteessa sen kehittämiseen käytettyyn aikaan ja resursseihin sekä kirjoittajan aiempaan kokemukseen chatbotien rakentamisesta ja

luonnollisen kielen ymmärtämisen toteuttamisesta. Toimeksiantaja arvioi prototyypin ja sen kehityksen aikana saatujen tuloksien antavan eväitä pohdittaessa mahdollisuuksia jatkokehitykselle sekä uusien liiketoimintatapauksen toteuttamiseksi.

#### **7.4 Prototyypin jatkokehitys**

Kirjoittaja näkee jatkokehityksen osalta tarvetta erityisesti käyttökokemuksen ja käyttöliittymän tarkemmalle suunnittelulle sekä viimeistelylle. Chatbotien tekstiin pohjaavien käyttöliittymien suunnittelu tulee vaatimaan täysin uudenlaista osaamista verrattuna esimerkiksi verkkosivujen suunnitteluun. Esimerkiksi chatbotin lähettämissä tulosteiden muotoilussa tulisi ottaa tarkasti huomioon tavoiteltava käyttäjäkunta tai asiakassegmentit. Vaihtoehtoisesti tulosteet voitaisiin laatia siten, että ne ovat kaikille samoja ja robottimaisen neutraaleja tai ne voisivat vaihtua käyttäjästä riippuen. Tällöin chatbotin viestintätyyli saattaisi muuttua esimerkiksi käyttäjän iän perusteella.

Lisäksi voidaan myös pohtia täysin prototyypistä poikkeavia mahdollisuuksia haun toteuttamiselle. Yksittäisen hakulauseen sijasta haku voisi olla mahdollista toteuttaa esimerkiksi vuoropuheluna chatbotin ja käyttäjän välillä, jolloin chatbot pyrkisi joka vastausvuoron välissä poimimaan käyttäjältä yhä enemmän hakua rajaavia kriteerejä.

Ominaisuuksien osalta chatbotin hakutoimintoon olisi tärkeää lisätä vähintään mahdollisuus palautettujen hakutulosten järjestelemiseen eri tavoilla kuten esimerkiksi hintojen perusteella. Jatkokehityksessä tulisi myös löytää ratkaisu saman tuotteen eri versioiden esittämiseen käyttäjälle.

Hyvä hakukone ei myöskään saisi milloinkaan palauttaa tulosta, jossa käyttäjälle ei esitetä ainuttakaan hakutulosta. Jatkokehityksessä tulee pohtia miten käyttäjälle esitetään kyseisissä tilanteissa vaihtoehtoisia tuloksia. Tämä voitaisiin pyrkiä toteuttamaan esimerkiksi jättämällä hausta pois jokin tuloksia rajaava kriteeri, mutta siten että tulokset vastaavat kuitenkin muita käyttäjän esittämiä kriteerejä.

Mikäli millään käyttäjän esittämälle kriteerillä ei ole saatu tuloksia tai käyttäjä on esittänyt niitä ainoastaan muutamia, on tuotteita esiteltävä jollakin muulla perusteella.

Jatkokehityksessä myös mahdollisten hakukriteerien määrää voitaisiin pyrkiä kasvattamaan. Esimerkiksi kameroiden tapauksessa kameran tarkkuuden ja megapikselien määrän voidaan ajatella olevan yksi olennainen hakukriteeri. Prototyypissä mahdollistetut hakukriteerit olivat myös melko perinteisiä ja uudenlainen tapa lähestyä hakuja voisi olla tuotteiden kategorisointi erilaisten adjektiivien perusteella. Kameroiden tapauksessa verkkokaupan tuotteille asetettuihin tageihin voitaisiin lisätä kuvauksia kuten *vedenkestävä* tai *helppokäyttöinen*. Luonnollisen kielen ymmärtämistä lisäävissä palveluissa taas pyrittäisiin määrittelemään erilaisia tapoja ja synonyymejä, joilla asiakkaat näitä adjektiiveja ilmaisevat. Tuotteiden kuvaaminen adjektiiveilla voisi mahdollistaa entistä luonnollisemman tavan tuotteiden hakemiselle ja muistuttaisi yhä enemmän aitoa ihmisten välistä asiakaspalvelutilannetta.

## 8 Pohdinta

Kirjoittajalle opinnäytetyö oli kokonaisuudessaan erittäin kattava ja monia eri aihekokonaisuuksia sisältänyt oppimisprosessi. Sen sisälle mahdutettiin useampia suuria, ajankohtaisia ja haastavia teemoja kuten tekoälyn hyödyntäminen liiketoiminnan ja asiakaspalvelun tukena, luonnollisen kielen käsittely sekä chatbotit ja niihin viime vuosien aikana liitetetyt ilmiöt. Teemojen yhteydessä tehty taustatutkimus sekä prototyypin toteuttaminen saivat kirjoittajan pohtimaan aiheiden ongelmia aina filosofian tasolta käytännön konkreettisiin ongelmiin asti. Jo tekoälyn ja luonnollisen kielen käsittelyn määrittelemisen ovat filosofisia ongelmia, joihin edes alojen asiantuntijoilla ei ole antaa täysin yksiselitteisiä vastauksia. Tekoälyn ja luonnollisen kielen käsittelyn toteuttamista pidetään myös konkreettisella tasolla tietotekniikan haastavimpiin ongelmiin kuuluvina ja syyt tulivat kirjoittajalle opinnäytetyön aikana selkeästi esille.

Tekoälyn ja chatbotien hyödyntäminen asiakaspalvelun tukena havaittiin ajankohtaiseksi aiheeksi, joka on kirvoittanut runsaasti keskustelua kansainvälisellä tasolla. Aiheen parissa tehdyn tutkimustyön aikana chatbotien hyödyntämiselle asiakaspalvelun tukena löytyi useita argumentteja niin sen puolesta kuin sitä vastaan. Myös kirjoittajan omat mielipiteet ovat asiakaspalvelubottien viime vuosina kasvaneesta trendistä pitkänkin pohdinta-ajan jälkeen edelleen kaksijakoiset.

Esimerkiksi pikaviestintäpalveluiden ja ääniohjattujen sovellusten suosion kasvu tukevat argumentteja chatbotien puolesta. Opinnäytetyön teoreettisessa taustassa esitettyjen tutkimusten mukaan myös asiakkaat ja käyttäjät ovat entistä avoimempia asiakaspalvelulle, joka tapahtuu ilman ihmiskontaktia. Yritysten tuodessa kehittäjiä saataville yhä useampia tekoälyn ja chatbotien toteutukseen tarkoitettuja työkaluja, voidaan aiheen parissa tehtyjen uusien innovaatioiden olettaa mahdollisesti lisääntyvän tulevien vuosien aikana.

Asiakaspalvelun automatisoinnin mahdollisuuksiin on kuitenkin vielä suhtauduttava maltillisesti ja todettava, että nykypäivänä pystytään automatisoimaan vasta hyvin yksinkertaisia ja rutiinitason asiakaspalvelutehtäviä. Selvimmän edun saavat edelleen suuret yritykset, joiden asiakaspalvelukeskukset vastaanottavat suurella volyyminä toistuvia ongelmia. Lisäksi suuret yritykset saavat kerättyä runsaat määrät asiakaspalvelutilanteisiin liittyvää dataa helpommin kuin pienet- ja keski-suuret yritykset.

Pienemmät yritykset voivat kuitenkin pyrkiä hyödyntämään esimerkiksi pikaviestintäsovellusten suosiota tarjoamalla niissä chatbotien avulla uuden palvelukanavan. Esimerkiksi ravintolaan tehtävien tilausten mahdollistamiseksi ei välttämättä tarvita tekoälyä tai big dataa. Pikaviestintäsovellukset ovat jo kehittyneet alustoiksi, jotka voivat tarjota käyttöliittymässään esimerkiksi valikkorakenteita sekä suorittaa monia toimintoja, joita aiemmin toteutettiin mobiiliapplikaatioiden tai verkkosivujen välityksellä.

Viime aikoina Suomessakin on alettu kiinnittämään entistä enemmän huomiota chatbotien mahdollisuuksiin asiakaspalvelun automatisoinnissa. Samalla myös



kotimainen keskustelu ja aiheeseen liittyvät tutkimukset ovat lisääntyneet. Esimerkiksi ammattikorkeakoulujen opinnäytetöiden julkaisemiseen tarkoitettuun Theseus.fi-palveluun on vuonna 2017 lisätty muutamia chatboteja vähintään jollakin tasolla sivuavaa työtä. Chatbotit ja niiden hyödyntämismahdollisuudet tarjoavatkin runsaasti mahdollisuuksia uusille aiheita eri näkökulmista tarkasteleville tutkimuksille ja opinnäytetöille. Kirjoittajaa kiinnostaisi esimerkiksi tutkimustieto chatboteja jo julkaisseiden kotimaisten tai kansainvälisten yritysten kokemuk- sista. Erityisesti tarvittaisiin tietoa kuinka paljon asiakaspalvelun automatisoinnilla on aidosti pystytty vähentämään asiakaspalveluun käytettyjä resursseja tai pa- rantamaan asiakaskokemusta.

Kirjoittaja sai opinnäytetyön aikana runsaasti arvokasta tietoa ja oppia monesta kirjoittajalle aiemmin tuntemattomasta aiheesta, teknologiasta ja työkalusta. Chatbotien toteuttaminen, luonnollisen kielen ymmärtämisen lisääminen chatbo- tiin sekä testiverkkokaupan toteuttaminen Shopify-verkkokauppa-alustalle olivat kaikki aivan uusia kokemuksia kirjoittajalle. Kirjoittaja uskoo opinnäytetyön aikana hankittujen taitojen tulevan vielä tarpeeseen vähintään jollakin tasolla valmistu- misen jälkeisen työuran aikana.

Kiitokset erinomaisesta ja haastavasta aiheesta sekä tuesta kuuluvat opinnäyte- työn toimeksiantajana toimineelle Projektitoimisto Proper Oy:lle. Lisäksi kirjoittaja haluaa kiittää opinnäytetyön ohjaajana toiminutta Seppo Nevalaista tuesta opin- näytetyöprosessin aikana.

## Lähteet

- Accenture Interactive. 2016. Chatbots in customer service. Accenture. [https://www.accenture.com/t00010101T000000\\_\\_w\\_\\_/\\_br-pt/\\_acnmedia/PDF-45/Accenture-Chatbots-Customer-Service.pdf](https://www.accenture.com/t00010101T000000__w__/_br-pt/_acnmedia/PDF-45/Accenture-Chatbots-Customer-Service.pdf). 5.10.2017.
- Arppe, A., Bartis, I., Carlson, L., Hyvärinen, M., Lennes, M., Linden, K., Koskeniemi, K., Nuolijärvi, P., Piehl, P., Vainio, M & Westerlund, H. 2012. Suomen kieli digitaalisella aikakaudella. META-NET. <http://www.meta-net.eu/whitepapers/e-book/finnish.pdf>. 9.10.2017.
- Aspect Software Inc. 2016. 2016 Aspect Consumer Experience Index. [https://www.aspect.com/globalassets/2016-aspect-consumer-experience-index-survey\\_index-results-final.pdf](https://www.aspect.com/globalassets/2016-aspect-consumer-experience-index-survey_index-results-final.pdf). 3.10.2017.
- Austin, K. 2017. The history and future of chatbots. In the Chat. <http://inthechat.com/chatbots/the-history-and-future-of-chatbots/>. 3.10.2017.
- Baldwin, S. 2017. From the unlikely setting of Ottawa, Shopify has quietly been powering an e-commerce revolution. Fortune. <http://fortune.com/2017/03/15/shopify-ecommerce-revolution/>. 30.10.2017.
- Best, J. 2017. IBM Watson: The inside story of how the Jeopardy-winning supercomputer was born, and what it wants to do next. TechRepublic. <http://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>. 5.10.2017.
- BI Intelligence. 2016. Messaging apps have surpassed social networks. Business Insider. <http://www.businessinsider.com/the-messaging-app-report-2015-11?r=US&IR=T>. 5.10.2017.
- BRB. 2017. Customer experience/Unified commerce survey 2017. <https://brpconsulting.com/wp-content/uploads/2017/05/2017-BRP-Unified-Commerce-Survey-051817.pdf>. 2.10.2017.
- Chatbots Magazine. 2017. China, WeChat, and the origins of chatbots. <https://chatbotsmagazine.com/china-wechat-and-the-origins-of-chatbots-89c481f15a44>. 5.10.2017.
- Chopra, A., Prashar, A. & Sain, C. 2013. Natural language processing. International journal of technology enhancements and emerging engineering research, vol 1, issue 4. <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=77B7E31AE432FA17E5EB1C943613BC70?doi=10.1.1.407.6907&rep=rep1&type=pdf>. 9.10.2017.
- CIW Team. 2016. Top WeChat stats and trends 2017. China Internet Watch. <https://www.chinainternetwatch.com/19524/wechat-data-report-2016/>. 5.10.2017.
- Codecademy. 2017. What is REST. <https://www.codecademy.com/articles/what-is-rest>. 7.11.2017.
- Columbus, L. 2014. Software companies are dominating Deloitte's Technology Fast 500. Forbes. <https://www.forbes.com/sites/louis-columbus/2014/11/20/software-companies-are-dominating-deloitte-technology-fast-500/#ff95e4517371>. 30.10.2017.
- Crunchbase. 2015. Shopify. <https://www.crunchbase.com/organization/shopify>. 11.10.2017.

- Daugherty, P & Purdy, M. 2016. Why artificial intelligence is the future growth. Accenture. [https://www.accenture.com/lv-en/\\_acnmedia/PDF-33/Accenture-Why-AI-is-the-Future-of-Growth.pdf](https://www.accenture.com/lv-en/_acnmedia/PDF-33/Accenture-Why-AI-is-the-Future-of-Growth.pdf). 2.10.2017.
- Freedman, D. 2017. A reality check for IBM's ambitions. MIT Technology Review. <https://www.technologyreview.com/s/607965/a-reality-check-for-ibms-ai-ambitions/>. 5.10.2017.
- Fuscaldo, D. 2017. Report: Facebook scales back on chatbots. Investopedia. <http://www.investopedia.com/news/report-facebook-scales-back-chatbots/>. 3.10.2017.
- Gartner Summits. 2011. Gartner customer summit 360 summit 2011. [https://www.gartner.com/imagesrv/summits/docs/na/customer-360/C360\\_2011\\_brochure\\_FINAL.pdf](https://www.gartner.com/imagesrv/summits/docs/na/customer-360/C360_2011_brochure_FINAL.pdf). 2.10.2017.
- Gonsalves, F & Shah, K. 2017. Detailed study on natural language processing services. International Research Journal of Engineering and Technology. <https://www.irjet.net/archives/V4/i6/IRJET-V4I6590.pdf>. 9.10.2017.
- Intento. 2017a. <https://inten.to/>. 12.9.2017.
- Intento. 2017b. <https://www.slideshare.net/KonstantinSavenkov/nlu-intent-detection-benchmark-by-intento-august-2017>. 12.9.2017.
- Johnson, K. 2017. Facebook Messenger hits 100,000 bots. VentureBeat. <https://venturebeat.com/2017/04/18/facebook-messenger-hits-100000-bots/>. 5.10.2017.
- JSON.org. 2017. Introducing JSON. <http://www.json.org/>. 8.11.2017.
- Lehtiniitty, M. 2017. Kotipizzaa voi nyt tilata botilta Facebook Messengerissä. Mobiili.fi. <http://mobiili.fi/2017/06/28/kotipizzaa-voi-nyt-tilata-botilta-facebook-messengerissa/>. 5.10.2017.
- Liikenne- ja viestintäministeriö, Tekes, Teknologiateollisuus & Verkkoteollisuus. 2017. Digibarometri 2017. Helsinki: Taloustieto Oy. [http://www.digibarometri.fi/uploads/5/8/8/7/58877615/digibarometri\\_2017.pdf](http://www.digibarometri.fi/uploads/5/8/8/7/58877615/digibarometri_2017.pdf). 2.10.2017.
- Loebner, H. 2015. The first Turing Test. <http://www.loebner.net/Prize/loebner-prize.html>. 3.10.2017.
- McKinsey Global Institute. 2017. Artificial intelligence the next digital frontier. [file:///C:/Users/Tuomas/Downloads/MGI-Artificial-Intelligence-Discussion-paper%20\(1\).pdf](file:///C:/Users/Tuomas/Downloads/MGI-Artificial-Intelligence-Discussion-paper%20(1).pdf). 3.10.2017.
- Microsoft. 2017a. Bot Framework frequently asked questions. <https://docs.microsoft.com/en-us/bot-framework/resources-bot-framework-faq>. 10.10.2017.
- Microsoft. 2017b. About the Bot Framework. <https://docs.microsoft.com/en-us/bot-framework/overview-introduction-bot-framework>. 10.10.2017.
- Microsoft. 2017c. Bot Framework SDKs and tools. <https://docs.microsoft.com/en-us/bot-framework/resources-tools-downloads>. 10.10.2017.
- Microsoft. 2017d. Bot Framework REST APIs. <https://docs.microsoft.com/en-us/bot-framework/rest-api/bot-framework-rest-overview>. 10.10.2017.
- Microsoft. 2017e. Define conversation steps with waterfalls. <https://docs.microsoft.com/en-us/bot-framework/nodejs/bot-builder-nodejs-dialog-waterfall>. 10.10.2017.
- Microsoft. 2017f. Manage State Data. <https://docs.microsoft.com/en-us/bot-framework/dotnet/bot-builder-dotnet-state>. 10.10.2017.

- Microsoft Azure. 2017a. Learn about Language Understanding Intelligent Service (LUIS). <https://docs.microsoft.com/en-us/azure/cognitive-services/luis/home>. 10.10.2017.
- Microsoft Azure 2017b. What is Azure? <https://azure.microsoft.com/en-us/overview/what-is-azure/>. 7.11.2017.
- Microsoft Azure. 2017c. Use prebuilt domains in LUIS apps. <https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-how-to-use-prebuilt-domains>. 10.10.2017.
- Microsoft Azure. 2017d. Cognitive Services pricing – Language Understanding Intelligent Services. <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/language-understanding-intelligent-services/>. 10.10.2017.
- Microsoft Azure. 2017e. Entities in LUIS. <https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-concept-entity-types>. 7.11.2017.
- Mindbrowser. 2017. Chatbot survey 2017. <https://www.slideshare.net/Mobileappszen/chatbots-survey-2017-chatbot-market-research-report>. 7.9.2017.
- Odin, M. 2017. Is Natural Language Processing (NLP) the future of e-commerce product search?. Loop54. <https://www.loop54.com/blog/is-natural-language-processing-nlp-the-future-of-ecommerce-site-search>. 14.11.2017.
- Oracle. 2016. Can virtual experiences replace reality?. [https://www.oracle.com/webfolder/s/delivery\\_production/docs/FY16h1/doc35/CXResearchVirtualExperiences.pdf](https://www.oracle.com/webfolder/s/delivery_production/docs/FY16h1/doc35/CXResearchVirtualExperiences.pdf). 3.10.2017.
- Oxford University Press. 2017. Chatbot. English Oxford Living Dictionaries. <https://en.oxforddictionaries.com/definition/chatbot>. 4.10.2017.
- Pascual-Nieto, I. & Perez-Marin, D. 2011. Conversational agents and natural language interaction: Techniques and effective practices. Hershey, PA: Information Science Reference (an imprint of IGI Global).
- Projektitoimisto Proper Oy. 2015. <http://konsultointi.proper.fi/>. 7.9.2017.
- Purmonen, T. 2017. Toimitusjohtaja. Projektitoimisto Proper Oy. Nauhoitettu haastattelu 6.12.2017.
- RND Works. 2017. Kierrätysbotti Kikka. <https://rnd.works/case-studies/lassilatikanoja>. 10.10.2017.
- Reese, H. 2016. Why Microsoft's 'Tay' AI bot went wrong. TechRepublic. <http://www.techrepublic.com/article/why-microsofts-tay-ai-bot-went-wrong/>. 3.10.2017.
- Schneider, C. 2017. 10 reasons why AI-powered, automated customer service is the future. <https://www.ibm.com/blogs/watson/2017/04/10-reasons-ai-powered-automated-customer-service-future/>. 3.10.2017.
- Shopify. 2009. Shopify launches API platform and App Store. <https://investors.shopify.com/Investor-News-Details/2009/Shopify-Launches-API-Platform-and-App-Store/default.aspx>. 30.10.2017.
- Shopify. 2017a. About Shopify. <https://www.shopify.com/about>. 11.10.2017.
- Shopify. 2017b. What does Shopify do?. <https://www.shopify.com/faq/what-does-shopify-do>. 11.10.2017.
- Shopify. 2017c. JavaScript Buy SDK. <https://help.shopify.com/api/sdks/custom-storefront/js-buy-sdk>. 14.11.2017.
- Shopify. 2017d. ShopClient. <https://shopify.github.io/js-buy-sdk/api/classes/ShopClient.html>. 10.11.2017.

- Shopify. 2017e. Object handles. <https://help.shopify.com/themes/liquid/basics/handle>. 10.11.2017.
- Shopify. 2017f. Creating and using tags in Shopify. <https://help.shopify.com/manual/productivity-tools/using-tags>. 10.11.2017.
- Shopify. 2017g. About the Shopify Partner Program. <https://help.shopify.com/partners/about>. 30.10.2017.
- Shopify. 2017h. Creating development stores. <https://help.shopify.com/partners/dashboard/development-stores>. 30.10.2017.
- Silcoff, S. 2013. A rare startup success story: Shopify hits \$1-billion milestone. The Globe and Mail. <https://www.theglobeandmail.com/report-on-business/small-business/sb-money/a-rare-startup-success-story-shopify-hits-1-billion-milestone/article15892998/?arc404=true>. 11.10.2017.
- Simonite, T. 2017. Microsoft: AI isn't yet adaptable enough to help businesses. MIT Technology Review. <https://www.technologyreview.com/s/603944/microsoft-ai-isnt-yet-adaptable-enough-to-help-businesses/>. 3.10.2017.
- TechCrunch. 2015. Facebook acquires Wit.ai to help its developers with speech recognition and voice interfaces. <https://techcrunch.com/2015/01/05/facebook-wit-ai/>. 11.9.2017.
- TechCrunch. 2017. Wit.ai is shutting down Bot Engine as Facebook rolls NLP into its updated Messenger Platform. <https://techcrunch.com/2017/07/27/wit-ai-is-shutting-down-bot-engine-as-facebook-rolls-nlp-into-its-updated-messenger-platform/>. 11.9.2017.
- TechTarget. 2016. Chatterbot (chatbot). <http://searchmicroservices.techtarget.com/definition/chatterbot-chatbot>. 4.10.2017.
- The Stanford NLP Group. 2017. CS224d: Deep learning for natural language processing. <http://cs224d.stanford.edu/>. 9.10.2017.
- Tutorialspoint. 2017. AI – Natural language processing. [https://www.tutorialspoint.com/artificial\\_intelligence/artificial\\_intelligence\\_natural\\_language\\_processing.htm](https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_natural_language_processing.htm). 9.10.2017.
- Wepopedia. 2017. Chat bot. QuinStreet Inc. [http://www.wepopedia.com/TERM/C/chat\\_bot.html](http://www.wepopedia.com/TERM/C/chat_bot.html). 4.10.2017.
- Wilhelm, A. 2015. Tech IPO Scorecard: Shopify skyrockets 51%, While Baozun Rises a slimmer 4.6%. TechCrunch. <https://techcrunch.com/2015/05/21/tech-ipo-scorecard-shopify-skyrockets-51-while-baozun-rises-a-slimmer-4-6/>. 30.10.2017.
- Wit.ai. 2015. Introducing composite entities. <https://wit.ai/blog/2015/05/05/composite-entity>. 1.11.2017.
- Wit.ai. 2017a. <https://wit.ai/>. 10.10.2017.
- Wit.ai. 2017b. Recipes for your apps you can talk to. <https://wit.ai/docs/recipes#extract-a-keyword-entity>. 1.11.2017.
- Wit.ai. 2017c. Which entity should I use. <https://wit.ai/docs/recipes#extract-a-keyword-entity>. 1.11.2017.
- Zhou, A. 2017. How artificial intelligence is transforming enterprise customer service. Forbes. <https://www.forbes.com/sites/adelynzhou/2017/02/27/how-artificial-intelligence-is-transforming-enterprise-customer-service/#2697c2bd1483>. 2.10.2017.