OULUN YLIOPISTO
UNIVERSITY of OULU

KANDIDAATINTYÖ

Juho Grundström

Kimmo Penttinen

Juho Purola

SÄHKÖ- JA TIETOTEKNIIKAN OSASTO

TIETOTEKNIIKKA

2011

OULUN YLIOPISTO
UNIVERSITY of OULU

KANDIDAATINTYÖ
SOPBOT –
Keskustelevan tekoälyn kehittäminen sulautettuun järjestelmään

Juho Grundström
Kimmo Penttinen
Juho Purola

Ohjaajat: Juha Röning, Teemu Tokola

SÄHKÖ- JA TIETOTEKNIIKAN OSASTO
TIETOTEKNIIKKA
2011

Grundström J, Penttinen K ja Purola J (2011) SOPBOT - Keskustelevan tekoälyn kehittäminen sulautettuun järjestelmään. Oulun yliopisto, sähkö- ja tietotekniikan osasto. Kandidaatintyö, 35 s.

TIIVISTELMÄ

Keskusteluagentti on ohjelma, jonka tarkoituksena on käydä keskustelua ihmismäisesti luonnollisella kielellä. Ihmistä huijaavaa tekoälyä ei ole vielä pystytty tekemään, mutta keskusteluagenteille on jo nyt useita erilaisia käyttökohteita. Tulevaisuudessa keskusteluagentit voivat korvata ihmisiä niille soveltuvissa tehtävissä. Keskusteluagentti tarjoaa luonnollisen kielen käyttöliittymän tekoälyille soveltuviin käyttökohteisiin. Ne suoriutuvat tehtävistä, jotka ovat matemaattisesti mallinnettavissa, paljon nopeammin ja paremmin kuin ihmiset. Tässä työssä on suunniteltu ja toteutettu muistillinen tekoäly keskusteluagentille, joka pystyy keskustelemaan suomen kielellä. Keskusteluagentteja on mahdollista tehdä eri tekniikoilla. Tämä toteutus on tehty käyttäen paranneltua kaavaansovitustekniikkaa yhdistettynä oppivaan tekoälyyn. Kaavaansovitustekniikka on toteutettu sekä lause- että sanatietokantojen avulla. Lausetietokannan käyttö mahdollistaa kirjoitusvirheiden hyväksymisen ja sanatietokannan avulla voidaan poimia käyttäjän tietoja muistiin sekä löytää vastaavuuksia paremmin. Tekoäly oppii käyttäjän syötteistä uusia lauseita ja vastauksia, joita se voi hyödyntää myöhemmin. Tällä yhdistelmällä saavutetaan mahdollisimman laadukasta ja laajaa keskustelua tuottava tekoäly. Työssä esitellään keskusteluagentin tekoälyn suunnittelu ja toteutus sekä viestittelykäyttöliittymä, jonka kautta tekoäly keskustelee. Menetelmän toteutus onnistui ja se saavutti hyvät testitulokset.

Avainsanat: kaavaansovitus, keskusteluagentti, tekoäly

Grundström J, Penttinen K and Purola J (2011) SOPBOT - Conversational artificial intelligence development for embedded system. Department of Electrical and Information Engineering, University of Oulu, Oulu, Finland. Candidate's thesis, 35 p.

ABSTRACT

A chatbot is a computer program whose purpose is to discuss using human-like natural language. Chatbots provide a natural language interface to computational tasks that are well handled by computer programs. They are able capable of executing mathematically formulated tasks much faster than humans. Chatbots have many appliances as of today, but artificial intelligence capable of completely mimicing human behaviour is yet to be developed. In the future chatbots should be able to replace humans in appropiate positions. This thesis is about designing and implementing a Finnish-speaking chatbot that has artificial intelligence with memory. It is able to successfully discuss with other Finnish-speaking agents, learn and remember details about them, and learn the spoken language during run-time. Of the many techniques used to design chatbots, the one in this thesis is implemented using pattern matching combined with learning capability. The pattern matching functionality is implemented using sentence and word databases. The sentence database is tolerant to spelling errors and the word database helps with picking up information from the user input while also making it possible to understand the meaning of the sentence even if some words do not match the sentence in the database. The artificial intelligence learns new sentences and responses from its input, and uses them to produce new conversations. This combination of techniques should produce the best possible conversational agent. The design and implementation of the chatbot's artificial intellegence using aforementioned methods along with the messaging interface is presented in this thesis. The implementation was successful and it achieved good results in the conducted tests.

Keywords: pattern matching, chatterbot, conversational agent, artificial intelligence

SISÄLLYSLUETTELO

ALKULAUSE

Tämä kandidaatintyö tehtiin Oulun yliopiston sähkö- ja tietotekniikan osaston Sulautettujen Ohjelmistojen Projekti -kurssiin. Työ toteutettiin kolmen hengen ryhmässä. Kurssin tavoite oli suunnitella ja toteuttaa tekoäly keskusteluagentille ja SNMP-protokollaa hyödyntävä keskustelukäyttöliittymä. Työn käytännön toteutus on tehty sähkö- ja tietotekniikan osaston tiloissa luokassa TS135, jossa on tarvittavat työkalut ohjelman toteuttamiseen ja testaamiseen verkossa. Kiitokset työn ohjaajalle Teemu Tokolalle.

31. Toukokuuta 2011

Juho Grundström, Kimmo Penttinen, Juho Purola

LYHENTEIDEN JA MERKKIEN SELITYKSET

AIML	Artificial Intelligence Markup Language
EIR	Elektror Internet Radio
IRC	Internet Relay Chat
OpenOCD	Open On Chip Debugger
RAM	Random access memory
SNMP	Simple Network Management Protocol
SOP	Sulautettujen Ohjelmistojen Projekti
MIB	Management Information Database

1. JOHDANTO

Keskusteluagentteja voi nykypäivänä kohdata kaikilla elämänalueilla yhä laajemmalle levittäytyneiden tietoverkkojen ja tietokoneiden avulla. Niitä on käytetty muun muassa opetustehtävien tukena esimerkiksi opiskellessa uutta kieltä, terveydenhuollossa itsehoito-ohjeiden antajana tai liike-elämän puolella antamassa tietoja jostain tuotteesta. Keskusteluagentteja löytyy myös viihde-elämän puolelta, jossa niihin voi törmätä tietokonepeleissä tai jopa internetin deittipalveluissa huijaamassa ihmisiä ihastumaan itseensä mukavan keskustelun avulla. Juuri se, että keskusteluagentit voivat esiintyä ihmisinä, tekee niistä mielenkiintoisia.

Kuinka hyvän keskusteluagentista voi lopulta tehdä? Kun se keskustelee monipuolisesti ja vaikuttaa niin inhimilliseltä, ettei sitä erota ihmisestä, onko kyseessä jo ajatteleva tekoäly? Yksi tärkein keskusteluagenttien tarjoama hyöty ja syy miksi niitä kehitetään on sen tarjoama luonnollisen kielen käyttöliittymä erilaisiin käytönnönsovelluksiin. Tietoa on nykyään paljon saatavilla erilaisissa muodoissa ja tietokannoissa. Keskusteluagentti pystyy tarjoamaan helpon käyttöliittymän näihin tietokantoihin ilman, että käyttäjän tarvitsee opetella jokaisen tietokannan käyttöä. Monet ongelmat ovat tehokkaasti ratkaistavissa tai mallinnettavissa matemaattisesti. Esimerkiksi strategiapelit, kuten shakki, ovat tekoälylle sopiva tehtävä. Tällaiset tilanteet tarjoavat luonnollisella kielellä keskustelevalle tekoälylle sopivan käyttökohteen, mikä tarjoaa viihdettä ja hyötyä sitä kaipaaville ihmisille.

Tekstimuodossa keskustelevasta ja ymmärtävästä tekoälystä on myös lyhyt hyppäys puhetta ymmärtävään tekoälyyn. Tällainen tekoäly, johon on lisätty robotiikkaa siten, että se pystyy liikkumaan ja vuorovaikuttamaan ympäristön kanssa, on kiehtonut ihmisiä jo 1900-luvun alusta asti tieteiskirjojen ja -elokuvien johdosta.

Luvussa kaksi esitellään keskusteluagenttien kehitystä historiasta nykypäivään käyden läpi erilaisia toteutusmalleja sekä tarkastellaa ongelmia, joita suomeksi keskustelevan tekoälyn toteutus aiheuttaa. Kolmannessa luvussa esitellään, kuinka keskusteluagentti on toteutettu. Sen tulee kyetä ymmärtämään ja tuottamaan suomen kieltä. Sen on myös kyettävä aloittamaan keskustelu ja sillä on oltava muistitoteutus. Neljäs luku kuvaa toteutuksen saavutukset ja käytetyt testausmenetelmät. Viidennessä luvussa on kuvaus projektin työnjaosta ja ajankäytöstä. Kuudennessa luvussa pohditaan, miten käytettyjä ratkaisuja voisi parantaa ja kehittää tulevaisuudessa.

2. TAUSTA

Keskusteluagentti on ohjelma, joka keskustelee ihmisten tai toisten keskusteluagenttien kanssa luonnollisella kielellä. Se on käytännössä tekoäly, joka lukee sille annettuja syötteitä ja muodostaa mahdollisimman ihmismäisen vastauksen[1]. Tässä luvussa käydään läpi keskusteluagenttien historiaa, tutustutaan eri toteutustekniikoihin ja tarkastellaan ongelmia, joita keskusteluagenttien kehittämisessä on.

Tässä luvussa esitellään myös viestittelykäyttöliittymä, jota keskusteluagentit voivat käyttää keskustelussa. Käytetty keskustelukäyttöliittymä perustuu IRC ja SNMP protokolliin. IRC on tekstipohjainen viestittelyohjelma tietokoneille[2]. SNMP on reaaliaikainen protokolla, jolla pyydetään ja lähetetään tietoa keskusteluagenttien välillä.

2.1. Keskusteluagentti

Keskusteluagentteja on rakennettu 1960-luvulta lähtien. Niiden toteuttamiseen on kehitetty tekniikoita, joilla saadaan aikaiseksi jo melko hyvin keskustelevia tekoälyjä. Keskusteluagenttien toteutuksessa ja suunnittelussa on kuitenkin useita ongelmia, joista johtuen ihmistä hämäävän tekoälyn kehittäminen ei ole vielä onnistunut.

2.1.1. Historia

Ensimmäisen keskusteluagentin ELIZA:n kehitti Joseph Weizenbaum vuonna 1966[3]. Se kehitettiin psykoterapian vastaanotolle tekemään alustavaa haastattelua potilaille. Se on muistiton, yksinkertaiseen avainsanojen havaitsemiseen tekstisyötteestä perustuva malli, johon on valmiiksi ohjelmoitu vastaukset. Tähän konseptiin perustuu suurin osa myöhemmin tehdyista keskusteluagenteista.[3, 1]

Psykiatri Kenneth Colby kehitti seuraavan kuuluisan keskusteluagentin PARRY:n vuonna 1972. Se oli hieman ELIZA:a kehittyneempi, sillä se toimi samalla periaatteella, mutta siinä oli lisäksi mallinnettuna pyrkimys luoda karkeaa keskustelua. 1970 ja -80 luvuilla kehitettiin keskusteluagentteja eri ohjelmointikielillä, jotka toteuttivat Weizenbaumin ELIZA mallia.[1]

Vuonna 1990 Hugh Loebner kehitti kilpailun keskusteluagenteille, joka perustuu Turingin testiin[4]. Sen palkintona on 100 000 dollaria Turingin testin läpäisevän keskusteluagentin kehittäjälle. Loebner prize -kilpailussa laitetaan ihminen ja keskusteluagentti keskustelemaan tuomarille, joka keskustelun päätteeksi arvioi kumpi puhujista oli kone ja kumpi ihminen. [5]

Eräs kuuluisa Loebner kilpailun voittaja on Richard Wallacen kehittämä A.L.I.C.E (Artificial Linguistic Internet Computer Entity), jota hän alkoi kehittää vuonna 1995. Sen kehittelyä hän on jatkanut tähän päivään asti[1]. A.L.I.C.E perustuu Wallacen luomaan AIML kieleen[6]. A.L.I.C.E. voitti Loebner palkinnot vuonna 2000,2001 ja 2004[5].

Useat uusimmat keskusteluagentit, kuten Jabberwacky, käyttävät tekniikkaa, missä ei ole valmiita keskustelutietokantoja, vaan se oppii jatkuvasti ihmisten syötteistä ja hyödyntää tätä historiatietoa keskustelussa[7]. Jabberwacky voitti Loebner palkinnot vuosina 2005 ja 2006[5].

2.1.2. Ongelmia

Turingin testistä suoriutuakseen tekoälyllä on oltava tarpeeksi tietämystä, sen on tehtävä järjellistä päättelyä sekä opittava keskustelun aikana[8]. Tekoälyn, joka toteuttaa edellä mainitut periaatteet, kehittäminen on heikon tekoälyn kannattajien mukaan mahdotonta[9 s.13-17].

Vahvan tekoälyopin mukaan mielen voidaan ajatella olevan toiminnaltaan tietokoneohjelma. Mikäli aivojen tarkka toiminta pystyttäisiin selvittämään, voitaisiin tämän opin perusteella luoda täysin ihmismielen kaltainen tekoäly. Vahvalla tekoälyllä ei ole nykyään monia kannattajia, koska aivojen toiminta on havaittu hyvin kompleksiseksi ja vaikeaksi tutkia. Nykyisin onkin voimassa heikon tekoälyn ajatus, jonka mukaan voidaan luoda malleja ja simulaatioita aivojen toiminnasta, jotka ainoastaan jäljittelevät sitä laskennallisten ominaisuuksien suhteen. Koska mallilla ei ole mieltä ei sen sisällä tapahdu ajattelua eikä se koe tunteita.[9 s.13-17]

Myös kieltä tutkimalla havaitaan keskusteluagentin suunnittelun olevan haastavaa. Ihmisen sanavarasto on kymmeniä tuhansia ellei lähemmäs satatuhatta sanaa. Lisäksi suomessa on sanoilla eri muotoja. Suomen substantiivilla on 2000 ja verbillä peräti 12000-15000 muotoa, kun otetaan huomioon johdokset ja yhdyssanat. Tilannetta hankaloittaa vielä suomen monimutkainen kielioppi. [10] Täydellisesti toimivan keskusteluagentin pitäisi pystyä ymmärtämään ja tuottamaan näin valtavasta sanamäärästä koostuvia laajoja lauserakenteita. Tämä kuulostaa jo puhtaasti laskennallisestikin hyvin hankalalta.

Kielellinen kommunikaatio ei ole kieliopillisten, hyvin muodostuneiden lauseiden kooste vaan ihminen käyttää koko tietovarastoaan tuottaessaan ja purkaessaan kielellisiä viestejä. Lauseiden muoto ja merkitys eivät välttämättä aina kohtaa. Kysyessä “tiedätkö paljonko kello on?” ei asiayhteyttä ymmärtämättä voi sano odotetaanko vastaukseksi kellon aikaa vai “kyllä tiedän“ toteamusta. Kommunikaation vaikuttavat kulttuuri ja asiayhteys eikä pelkästään formaali kielioppi, jota pystyttäisiin periaatteessa tietokoneohjelmalla mallintamaan.[9 s.276]

2.1.3. Keskusteluagenttien arviointi

Keskusteluagenttien toteutuksia voidaan arvioda erilaisilla testeillä, joiden tarkoituksena on todeta kuinka hyvin kukin toteutus ratkaisee keskusteluagenttien toteutukseen liittyviä ongelmia.

Turingin testi on eräs keskusteluagentille tehtävä testi, jota käytetään Loebner-prize kilpailussa. Sen tarkoituksena on selvittää, voiko kone huijata ihmistä siten, että ihminen luulee koneen olevan toinen ihminen. Koska edellä mainittu testausmenetelmä on testaajasta riippuvainen, on käytettävä myös jotain helpommin mitattavaa evaluointimenetelmää. Voidaan esimerkiksi mitata kysymykseen annetun vastauksen oikeellisuutta vertaamalla sitä hakukoneen antamiin tuloksiin. Mitattavissa on myös kuinka usein keskusteluagentti kykenee antamaan hyvän vastauksen eikä tyydy yleiskommenttiin. Yksi tapa on pisteyttää vastaukseen johtaneita polkuja kuten Bayan Abu Shawar ja Eric Atwell esittävät artikkelissaan[11]. Siinä vastausta haetaan vertaamalla syötettä lauseisiin, ensimmäiseen sanaa, ja merkityksellisimpään sanaan. Myös botin vastaukset arvioidaan asteikolla järkeenkäypä, outo, järjetön.

2.1.4. Keskusteluagenttien toteutustekniikoita

Keskusteluagentteja on yritetty saada keskustelemaan ihmismäisesti useilla eri tekniikoilla. Osa tekniikoista perustuu avainsanojen tunnistamiseen syötteestä ja osa keskusteluagenteista on kehitetty oppimaan keskustelun aikana. Niitä on myös toteutettu käyttäen hyväksi tilastollisiin todennäköisyyksiin perustuvaa Markovin mallia sekä semanttiseen verkkoon pohjautuvaa oppimiskykyä. Keskusteluagentteja varten on myös kehitetty luonnollista kieltä tuottava AIML kieli.

Kaavaansovitus

Kaavaansovitus (engl. pattern matching) on menetelmä, jolla tutkitaan esiintyykö jokin tietty kaava vertailtavana olevassa isommassa tietomäärässä. Kun etsitty kaava on merkkijono, jota etsitään toisesta merkkijonosta, niin puhutaan yleensä merkkijonovertailusta(engl. string matching)[12]. Tällöin haetaan joko yksi tai kaikki ilmentymät etsitystä merkkijonosta ja näiden vastaavuuksien täytyy olla aina täysin samanlaiset[13]. Esimerkki kaavaansovitustekniikan yksinkertaisimmasta hakualgoritmista: Etsitään merkkijono ”ab”, merkkijonosta ”cabc”

Vertaillaan merkkijonon ab ensimmäistä kirjainta merkkijonon cabc vastaavaan:
->[a]b
->[c]abc
Ei löydy vastaavuutta.

Vertaillaan merkkijonon ab ensimmäistä kirjainta merkkijonon cabc toiseen kirjaimeen:

–>[a]b
->c[a]bc
Löytyy ensimmäinen vastaavuus a-merkkien osalta.

Vertaillaan merkkijonon ab toista kirjainta merkkijonon cabc kolmanteen kirjaimeen
–>a[b]
->ca[b]c
Löytyy toinen vastaavuus b-merkkien osalta ja tiedetään, että merkkijono ab löytyy ainakin kerran.

Algoritmi lopettaa vastaavuuksien etsimisen, jos merkkijonon ab täytyy löytyä vain kerran. Jos kaikki sen mahdolliset ilmentymät täytyy löytää, niin vertailua jatketaan merkkijonon ab ensimmäisestä merkistä ja merkkijonon cabc kolmannesta merkistä, joka tulee seuraavaksi. Tämä algoritmi käy läpi merkkijonon cabc jokaisen merkin ja on täten hidas, jos merkkijono olisi pidempi. On kehitetty nopeampia algoritmeja, jotka vähentävät tutkittavien merkkien määrää [13].

Useissa ohjelmointikielissä on kirjastoja, jotka tarjoavat valmiita merkkijonovertailutekniikoita, jota kutsutaan säännöllisiksi lauseiksi[14].

Likimääräinen merkkijonovertailu (engl. approximate string matching) on tekniikka, joka sallii myöskin kirjoitusvirheet vertailtavissa merkkijonoissa. Tällöin merkkijonovertailutekniikoilla tutkitaan merkkijonoja. Käyttöön on otettu myös lisävertailu,

joka sallii esimerkiksi yhden tai kahden merkin vaihtelut ja vastaavuuksien ei tällöin tarvitse olla täydelliset.[15]

AIML

AIML (Artifical Intelligence Markup Language) on Richard Wallacen kehittämä merkkauskieli, jota käytetään luonnollista kieltä käyttävien keskusteluagenttien tekemiseen[6]. Kieli koostuu useammasta elementistä. Kategoriat ovat tietämyspaketteja, jotka koostuvat kaavasta (engl. pattern) ja mallista (engl. template). Kaavat kuvaavat merkkiketjuja, joista käyttäjän syöte koostuu. Kaavoissa voi olla myös korvausmerkkejä (engl. wildcards), jotka voivat sisältää mitä tahansa sanoja. Malli kuvaa vastausta sopivaan kaavaan ja se voi sisältää myös muuttujia, joihin voidaan tallentaa historiatietoa keskustelusta. Alla on esimerkki AIML tekniikan käytöstä:

<category>
<pattern> MITÄ KUULUU</pattern>
<template> Hyvää, entä itsellesi?</template>
</category>

Kun tähän lisätään korvausmerkit, saadaan aikaiseksi monipuolisempi vastausrunko. Kaavaan laitetaan ’*’-merkki kuvaamaan mitä tahansa sanaa tai lausetta. Alla olevalla tietoyksiköllä voidaan vastata useisiin kysymyksiin, kun ’*’-merkki korvataan sen kohdalle osuvalla sanalla tai lauseella.

<category>
<pattern>TYKKÄÄTKÖ SINÄ *</pattern>
<template>En pahemmin välitä *</template>
</category>

Tällä saataisiin kysymykseen "Tykkäätkö sinä pizzasta?"vastaus "En pahemmin välitä pizzasta".

Markovin malli

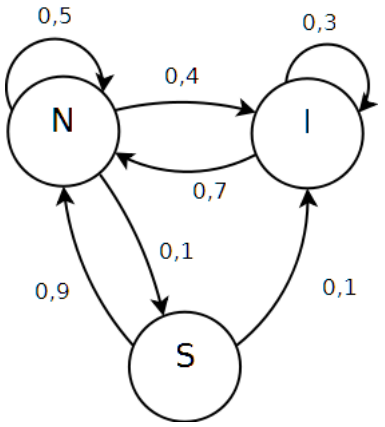
Markovin ketju on tilastollisiin todennäköisyyksiin perustuva prosessi, jossa uuden tilan todennäköisyys riippuu vain sitä edeltävästä tilasta. Markovin ketju koostuu joukosta tiloja $S=s_1,s_2,...,s_n$. Prosessi aloittaa yhdestä näistä tiloista ja siirtymistä tilasta toiseen kutsutaan askeleeksi. Kun ketju on tilassa s_i ja siirtyy tilaan s_j seuraavalla askeleella, niin tämän siirtymisen todennäköisyyttä merkitään p_{ij}. Todennäköisyys ei riipu tiloista joissa käytiin ennen tilaan s_i siirtymistä. Todennäköisyyksiä p_{ij} kutsutaan siirtymistodennäköisyyksiksi. Prosessi voi myöskin säilyä nykyisessä tilassaan ja tätä todennäköisyyttä merkitään p_{ii}. Siirtymistodennäköisyydet voidaan esittää matriisimuodossa p_{ij}:n matriisina P. Prosessille asetetaan yleensä jokin aloitustila tilojen S joukkoon, josta se lähtee liikkeelle.[16]

Esimerkki Markovin mallista:
On kehitetty keskusteluagentti, jolle on ohjelmoitu tunteet, joita se osaa simuloida

toiminnassaan. Keskusteluagentin tunnetilat vaihtelevat asetetuin todennäköisyyksin. Nämä tunteet voidaan esittää Markovin mallin tiloina S = N,S,I, missä N on neutraali, S on surullinen ja I on iloinen tunnetila. Tila N on samalla aloitustila. Siirtymistodennäköisyydet voidaan esittää matriisina.

$$P = \begin{array}{c} \\ N \\ S \\ I \end{array} \begin{array}{c} \begin{array}{ccc} N & S & I \end{array} \\ \begin{pmatrix} 0{,}5 & 0{,}1 & 0{,}4 \\ 0{,}1 & 0 & 0{,}1 \\ 0{,}4 & 0 & 0{,}3 \end{pmatrix} \end{array}$$

Kuvassa 1 on matriisia P vastaava Markovin mallin tilasiirtymien graafinen esitys.



Kuva 1. Markovin mallin tilasiirtymät.

Markovin ketjusta on myös mahdollista laskea todennäköisyydet eri tilojen etenemisille, kuten esimerkiksi millä todennäköisyydellä kolme neutraalia tunnetilaa tapahtuu peräkkäin.[16]

Markovin malli keskusteluagenteissa:
Markovin mallia on mahdollista käyttää keskusteluagenteissa tuottamassa vastauksia niille annetuille syötteille. Yksi tällä menetelmällä toteutetuista keskusteluagenteista on Jason Hutchensin ja Michael alderin vuonna 1998 tekemä MegaHAL. Tämän toteutuksen Markovin mallissa sanat ovat mahdollisia tiloja, ja näiden välisten askeleiden todennäköisyydet on saatu opettamalla, millä todennäköisyydellä sanat esiintyvät peräkkäin tekstikorpuksessa eli suuressa määrässä kirjoitettua tekstiä.

MegaHAL toimii siten, että kun se on saanut viestin, niin se jakaa tämän viestin sanoihin. Näistä sanoista etsitään avainsana, josta lähdetään tuottamaan vastaus. Lause saadaan siten, että muodostetaan avainsanasta eteenpäin lauseen loppuun mahdollisten eri sanojen joukot. Tämän jälkeen muodostetaan samalla menetelmällä mahdolliset sanat avainsanasta lauseen alkuun päin. Tällä tavalla saadaan muodostettua useita erilaisia lauseita, joista valitaan paras. Paras vastaus on se, joka toteuttaa Markovin mallin suurimmalla todennäköisyydellä.

Tässä menetelmässä on mahdollista käyttää erilaisia tekstejä opettamaan keskusteluagentille Markovin malli, jolloin keskusteluagentti oppii luomaan vastauksia eri aihealueista tai eri kielillä. Tekstit voivat olla tietosanakirjamaisia lähteitä, keskustelulokeja internetin keskustelupalstoilta tai vaikka kuuluisien siteerauksien listoja. Mitä

isompi tekstilähde on, sitä suurempi ja parempi Markovin mallista tulee, koska eri sanojen ja näiden välisten askeleiden määrä kasvaa. Samalla myös keskusteluagentista tulee parempi, koska se pystyy tuottamaan suuremman määrän erilaisia vastauksia.[17]

Lause- ja sanavertailumalli

AIML tekniikkaan perustuvaa mallia helpommin toteutettavissa on lauseiden ja sanojen vertailuun perustuva malli, jota Ehab Ahmed El-agizy esittelee nettiartikkelissaan [18]. Tämän mallin mukaan käyttäjän syötettä verrataan kahteen tietokantaan, joista toinen sisältää lauseita ja toinen peräkkäisiä sanoja. Jotta vastaavuus löytyisi lausetietokannasta, on syötteen löydyttävä sellaisenaan. Sanatietokannassa riittää, että siinä olevat sanat löytyvät lauseesta, jolloin useampi erilainen syöte aiheuttaa tietokantaosuman. Kun osuma on löytynyt jommasta kummasta tietokannasta, haetaan sitä vastaava lause vastauksena käyttäjälle.

Tätä mallia voidaan parantaa kirjoittamalla useampia vastauksia samalle syötteelle ja linkittää samankaltaisia syötteitä tälle vastausjoukolle. Tällöin keskusteluagentti ei vastaa aina samalla tavalla ja vaikuttaa ihmismäisemmältä[19].

Datanhakija

Keskusteluagenttien eräs toteustekniikka on prosessoida saatu syöte käyttämällä hyväksi olemassa olevia ulkoisia tietolähteitä, kuten Wikipediaa[20, 21]. Tämä tekniikka ei välttämättä yksinään ole kovin käyttökelpoinen, koska keskustelu on verrattavissa internetin hakukoneen käyttöön. Keskusteluagentti vain vastaa löytämällään vastauksella, eikä kunnollista keskustelua synny. Tekniikka sopiikin hyvin käytettäväksi yhdessä muiden toteutustapojen kanssa, sillä se antaa ihmismäisen vaikutelman, kun ilmiselviin kysymyksiin annetaan hyviä vastauksia.

Oppiva tekoäly

On myös keskusteluagentteja, jotka eivät käytä ainoastaan valmista sana- tai lausetietokantaa keskustelun rakentamiseen. Tällainen oppivaan tekoälyyn perustuva keskusteluagentti on esimerkiksi Jabberwacky[7]. Sen toiminta ei perustu ollenkaan valmiisiin keskustelutietokantoihin, vaan se oppii jatkuvasti ihmisten syötteistä, ja hyödyntää tätä historiatietoa keskustelussa. Myös monet muut internetin keskusteluagentit käyttävät tätä tekniikkaa.

Oppiva keskusteluagentti voi myös perustua valmiiseen keskustelutietokantaan, joka päivittyy sitä mukaa kun se oppii keskustellessaan muiden kanssa.

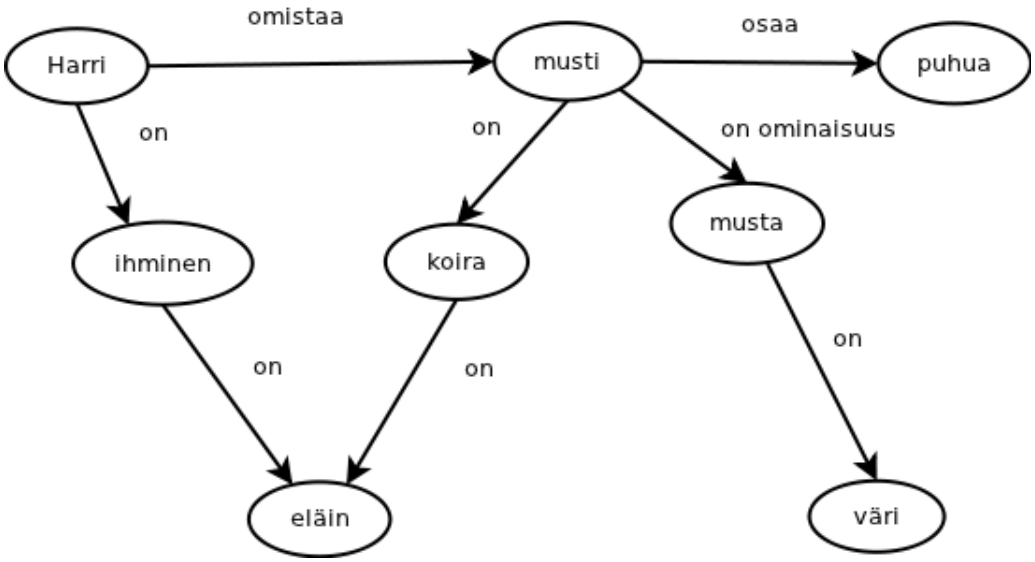
Semanttiset verkot oppivuuden toteutuksessa

Ihmisen ymmärrys ympäröivästä maailmasta perustuu osittain semanttiseen muistiin. Siinä tieto on säilössä jäsennellyssä muodossa siten, että asiat on esitetty erilaisten käsitteiden tai merkitysten avulla, ja näiden välillä on erilaisia yhteyksiä.
Semanttinen verkko perustuu tähän, ja sillä voidaan kuvata tietoa, jota esimerkiksi keskusteluagentti saa sille lähetyistä viesteistä. Semanttinen verkko voidaan esittää graafina, jossa käsitteet tai merkitykset ovat graafin solmuja, ja näiden väliset yhteydet ovat erilaisia käsitesuhteita. Semanttiseen verkkoon voidaan lisätä jatkuvasti uusia

solmuja ja näiden yhteyksiä, sekä muokata vanhoja.[22]

Mahdollisia käsitesuhteita voivat olla esimerkiksi:
A on B
A:lla on ominaisuus B
A osaa kyvyn B
A omistaa B:n

Ajatellaan että keskusteluagentti saa seuraavat viestit:
Harrilla on koira, jonka nimi on Musti. Musti on musta ja osaa puhua.
Nyt keskusteluagentti voi esittää saamansa tiedon osana semanttista verkkoa, edellyttäen että sille on jo opetettu, että koirat ovat eläimiä, Harri on ihmisen yksi mahdollinen nimi, ihminen on eläin sekä musta on väri. Kuvassa 2 esitetään esimerkistä muodostettu semanttinen verkko.



Kuva 2. Esimerkin semanttinen verkko.

2.2. SOP-IRC

Työssä käytetty keskustelukäyttöliittymä SOP-IRC käyttää viestinvälitykseen SNMP protokollaa. Protokollassa voidaan eri pyyntöjen avulla lähettää ja pyytää tietoa. SNMP:ssä data sijaitsee olioissa, joihin viitataan olio id:illä. Oliotieto on tallennettu MIB tietokantaan, josta sitä voi pyynnöillä hakea tai muuttaa.[23]

2.3. Käytettävä laitteisto

Työn toteutus tapahtuu Ethernutin EIR evaluointilaudalle, joka koostuu Atmelin mikrokontrollerista, muistista ja io-laitteista [24]. Ohjelma kirjoitetaan Eclipse kehitysympäristöä

käyttäen ja se käyttää NutOS käyttöjärjestelmän tarjoamia palveluita. Muita työkaluja ovat Yagarto, mikä sisältää kehitystyökaluja kuten kääntäjän ja linkkerin sekä OpenOCD, joka luo kommunikointiyhteyden työaseman ja sulautetun järjestelmän välillä. Keskustelutietokantaa säilytetään SD-muistikortilla, josta se luetaan laitteen muistiin.[25]

3. RATKAISUJEN KUVAUS

Tässä luvussa kuvataan, kuinka keskusteluagentti on rakennettu, ja kerrotaan valitut ratkaisut ja suunnitteluperiaatteet. Luvussa esitellään myös tekoälyn tarkka toiminta vastauksen luomiseksi käyttäjälle.

3.1. Johdanto

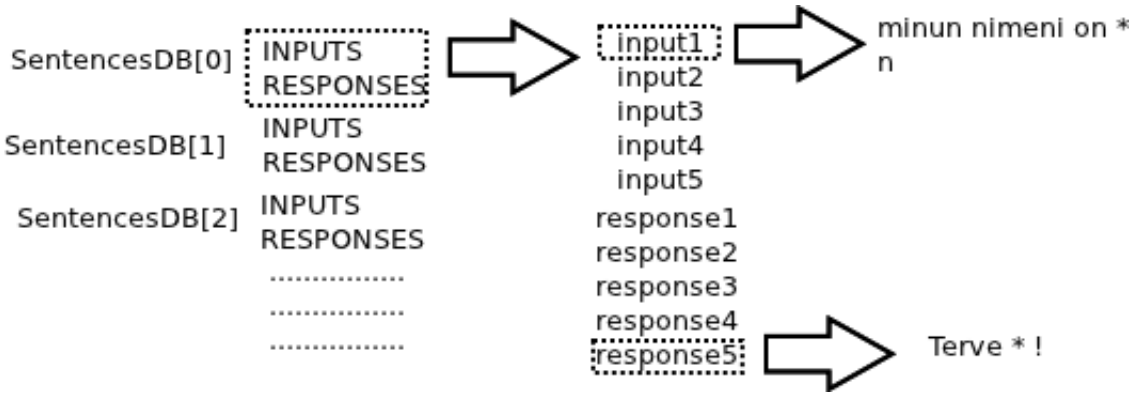
Tässä työssä toteutettu keskusteluagentti käyttää kahta erilaista tekniikkaa tuottaakseen keskustelua. Kaavaansovitusta käytetään haettaessa vastaavuuksia tietokannasta, joka sisältää yleisesti käytettyjä lauseita ja niille sopivia vastauksia. Toinen käytetty tekniikka hyödyntää oppivan tekoälyn ajatusmallia, jossa keskusteluagentti oppii lauseita ja vastauksia sille annetuista syötteistä. Toteutuksessa paneudutaan siihen, että käyttäjän syöte tunnistettaisiin mahdollisimman hyvin, jolloin sille voidaan muodostaa järkevä vastaus. Käyttäjää pyritään keskusteluagentin syötteillä ohjaamaan siten, että tietokantaan saataisiin tallennettua mahdollisimman hyviä ja jatkossa hyödynnettäviä uusia lauseita. Suomen kieltä ei käsitellä toteutuksen kannalta muuten kuin käyttämällä sanatietokantaa, joka mahdollistaa tietokantaosuman löytymisen eri sanajärjestyksiin kirjoitetuista lauseista. Keskusteluagentille ei toteuteta kielioppisääntöjä eikä asiayhteysmuistia edellisistä keskusteluista.

3.2. Rajoitukset

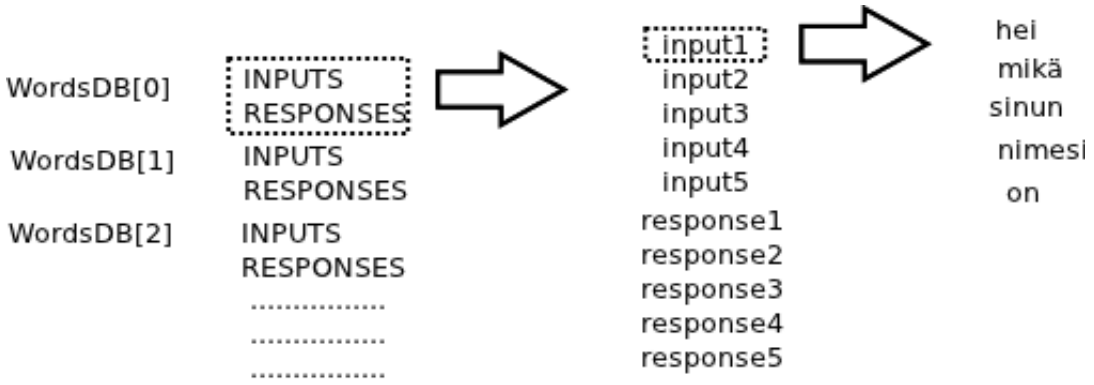
Työssä käytetty laitteisto asettaa rajoituksia muistinkäytön suhteen. Laitteen 64 megatavun RAM-muistin on riitettävä ohjelman tarpeisiin, joten muistinkäytön suhteen on oltava tarkkoja. Tietokanta luetaan SD-muistikortilta laitteeseen, joten laitteen muistin määrä asettaa rajat oppivan tekoälyn keräämien lauseiden määrälle.

3.3. Tietorakenteet

Tietorakenteiden toteutukseen on valittu kaksi tietokantaa, sana- ja lausetietokannat, joista kaavaansovitustekniikalla haetaan vastauksia. Tietokannat ovat tietuetaulukoita, joissa tietueet sisältävät merkkijono-osoitintaulukot syötteille ja vastauksille sekä niiden määrät. Lausetietokannassa on myös tallennettuna mahdollinen lippu, joka kuvaa lauseen aihetta. Sanatietokannassa lauseet on tallennettu merkkijono-osoitintaulukoihin sanoina. Tietokantojen rakenne on kuvattu alla kuvissa 3 ja 4.
Esimerkkitietokannassa oleva ’n’ tarkoittaa nimeä. Tutkimalla lippuja voidaan tallentaa henkilön nimi, asuinpaikka, harrastus sekä kiinnostuksen kohde erilliseen tietorakenteeseen. Näitä tietoja voidaan hyödyntää keskustelussa luoden tunnetta siitä, että tekoälyllä on muisti.



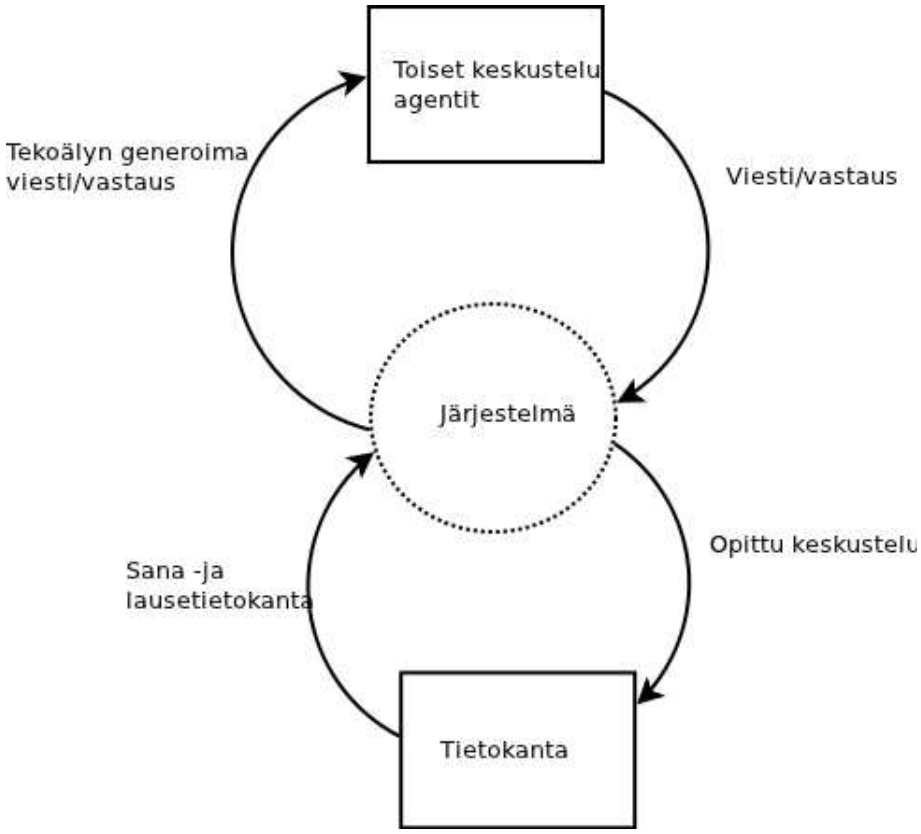
Kuva 3. Lausetietokanta.



Kuva 4. Sanatietokanta.

3.4. Arkkitehtuurikuvaus

Arkkitehtuurin pääkomponentit on esitetty kuvassa 5 olevassa asiayhteyskaaviossa.



Kuva 5. Asiayhteyskaavio.

Keskusteluagentin pääkomponentit ovat viestinvälityksen hoitava SNMP-protokollaa hyödyntävä tietoliikennerajapinta ja itse keskustelun hoitava tekoäly. Ohjelman osien välinen tiedonvälitys tapahtuu käyttäen jonotietorakenteita.

3.4.1. Tiedonsiirtoon käytettävien jonojen kuvaus

Tiedonsiirtoon käytetyt jonot ovat yksinkertaisia tietorakenteita, jotka sisältävät käsiteltävän datan lisäksi tiedon datan pituudesta, tyypistä ja alkuperästä. Jonot toimivat ohjelman komponenttien välisinä rajapintoina. Kaikkien komponenttien syötteet välitetään jonojen avulla, joihin muut komponentit sijoittavat vasteensa.

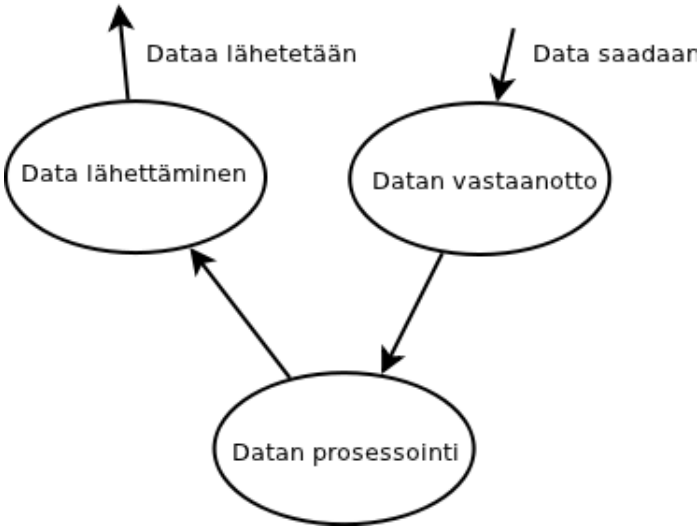
3.4.2. Ohjelman suorituksen eteneminen

Ohjelman kunkin osan syötejonoa tutkitaan vuorotellen. Mikäli jonossa on käsiteltävää dataa, suoritetaan kyseisen ohjelman osa, muulloin siirrytään tutkimaan seuraavan osan syötejonoa. Ohjelman eri osien suorittaminen on jatkuvasti toistuvaa.

3.4.3. Tietoliikennerajapinnan kuvaus

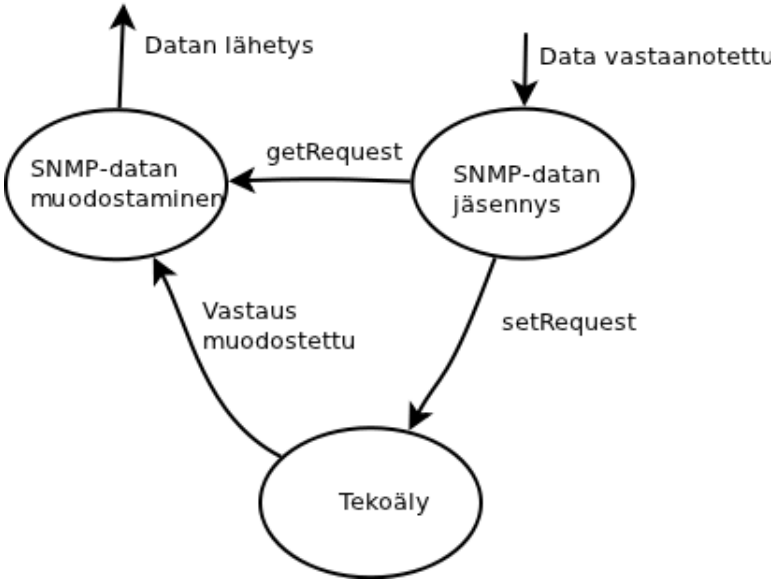
Ohjelman käyttöliittymänä toimiva tietoliikennerajapinta perustuu SNMP-protokollaan. Sekä syöte että vaste koostuvat SNMP-paketeista. Saapuvat tietoliikennepaketit tarkastetaan virheiden varalta ja virheelliset paketit hylätään kokonaan, samoin SNMP-protokollaa noudattamattomat paketit. Pakettien siirto ja jäsentäminen on toteutettu kahdessa eri osassa, joiden välinen tiedonsiirto tapahtuu jonojen avulla. Ohjelman sisäinen tiedonsiirto ei noudata SNMP-protokollaa, vaan siihen käytetään yksinkertaisia merkkijonoja. SOP-IRC protokollan mukaiset pyynnöt käsitellään sisäisesti tietoliikennekomponentissa. Ainoastaan tekoälyn käsiteltäväksi tarkoitetut viestit toimitetaan muille komponenteille.

Alla olevassa kuvassa 6 esitetään datan yleinen kulku järjestelmässä.



Kuva 6. Datan kulku järjestelmässä.

Kuva 7 esittää tarkemmin datan prosessoinnin järjestelmässä.



Kuva 7. Datan prosessointi.

3.4.4. SOP-IRC:n toteutus

Kaikki SOP-IRC -standardin toiminnot on toteutettu. Viestejä voi lähettää ja vastaanottaa, muiden käyttäjien nimiä voidaan kysyä ja oma nimi voidaan lähettää. Viesti välitetään SNMP SET-REQUEST -pyynnöllä, nimeä pyydetään SNMP GET-REQUEST -pyynnöllä ja pyyntöihin vastataan SNMP-GET-RESPONSE -vastauksella. Muunlaisia viestejä ei tueta. Protokollan mukaiset vasteet lähetetään SNMP-porttiin 161, sekä siihen porttiin, mistä ohjelman saama syöte on tullut.

3.4.5. Tekoälyn toiminnan kuvaus

Keskusteluagentin tekoäly toimii siten, että tekoälykomponentti saa tietoliikennerajapinnalta viestin, jolle se generoi vastauksen ja antaa sen takaisin. Vastauksen generointi koostuu useasta kohdasta, jossa vastausta haetaan ensin laustetietokannasta ja sitten sanatietokannasta. Lopuksi käytetään erikoisvastauksen muodostamista. Aliluvuissa esitetään tarkemmin tekoälykomponentin osien toiminta ja vastauksen muodostaminen.

Lausetietokannan toiminta

Lausetietokantahaku toimii siten, että saatua syötettä verrataan jokaiseen tietokannassa olevaan syötteeseen. Vertailufunktiona käytetään Levenšteinin etäisyyttä. Se on kahden lauseen välillä oleva luku, jonka verran merkinsiirto-operaatioita joudutaan tekemään, jotta saataisiin lause muutettua toiseksi[26]. Levenšteinin etäisyys on valittu käytettäväksi siitä syystä, että lauseiden ei tarvitse olla identtisiä, jotta vastaavuus löytyisi.

Syötteen pituudesta riippuen on käytetty Levenšteinin etäisyyksiä nolla, yksi ja kaksi. Kun syötteen pituus on välillä [1,2], käytetään arvoa nolla, koska tähän jokkoon kuuluu sanoja, kuten “OK“, joiden on tunnistuttava täysin oikein, jottei virhetulkintoja tulisi. Välillä [3,4] hyväksytään yksi virhe ja viidestä eteenpäin hyväksytään Levenšteinin etäisyydeksi kaksi. Näihin arvoihin päädyttiin järkeilemällä parhaat etäisyydet, jotta tahattomia kirjoitusvirheitä sekä pieniä taivutusmuotoeroja hyväksyttäisiin, mutta tekoäly ei kuitenkaan kelpuuttaisi erimerkityksellisiä lauseita.

Sanatietokannan toiminta

Sanatietokanta perustuu siihen, että syöte jaetaan ensin sanoiksi, joita etsitään tietokannan lauseiden sanoista. Mitä useampi yhteinen sana löytyy syötteestä ja sanatietokannan lauseesta, sitä varmemmin ne ovat merkitykseltään samankaltaisia. Vastaus valitaan suurimman määrän yhteisiä sanoja sisältävästä tietorakenteen kohdasta. Sanatietokantaan vertaillessa sanan on oltava identtinen tietokannan sanan kanssa. Jottei hyväksyttäväksi tulisi vain yhden yhteisen sanan lauseita, käytetään vähimmäismääränä kolmea yhteistä sanaa. Testien perusteella kolme samaa sanaa antaa melko tarkkoja osumia, kunhan tietokannan lauseisiin ei kirjoiteta paljoa täytesanoja, jotka ovat yleisiä useissa lauseissa.

Sanatietokannassa voidaan tiettyjä sanoja painottaa kirjoittamalla ne useamman kerran. Tällöin harvinaislaatuisempia sanoja osataan yhdistää varmemmin oikeaan tietokannan osaan, painotettujen sanojen antaessa useampia osumia.

Oppivan tekoälyn toiminta

Oppiva tekoäly toimii siten, että keskustelijan sanomat lauseet, joita ei vielä ole tietokannassa, tallennetaan sinne, ja asetetaan lauseelle q-lippu kuvaamaan opittua lausetta. Välillä, kun keskusteluagentti ei ymmärrä käyttäjän lausetta, se vastaa sanomalla q-lipullisen lauseen, jolle käyttäjän antama vastaus tallennetaan vastaukseksi.

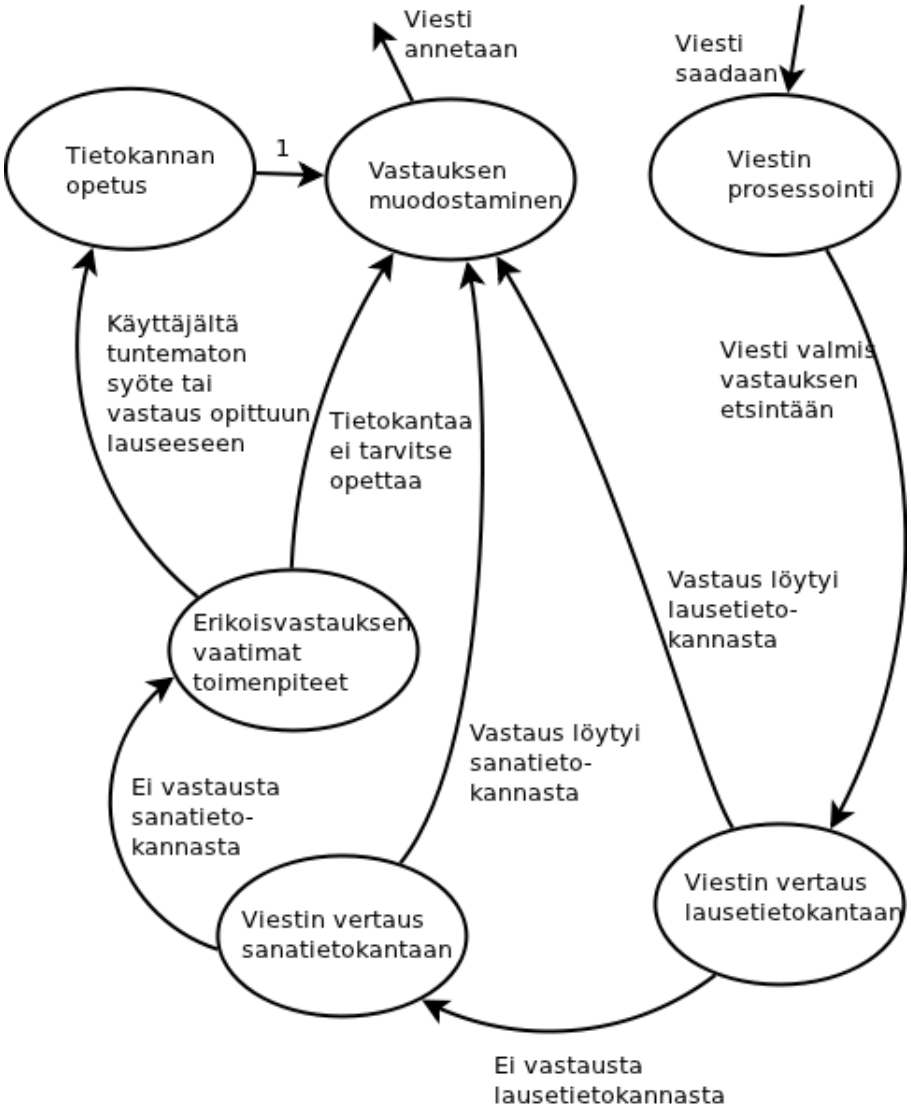
Vastauksen muodostaminen

Keskusteluagentti aloittaa vastauksen muodostamisen erottamalla syötteestä erikoismerkit ja muuttamalla sen pieniksi kirjaimiksi. Tämä operaatio tehdään siksi, että saadaan syöte yhdenmukaistettua tietokannan kanssa. Ensimmäinen vertailu tapahtuu lausetietokantaan. Mikäli sieltä ei löydy vastaavuutta, etsitään osumia sanatietokannasta.

Siinä tapauksessa, ettei sopivaa vastausta ole muodostettu tietokantahakujen jälkeen, joutuu keskusteluagentti muodostamaan erikoisvastauksen. Keskusteluagentilla on useita tapoja jatkaa keskustelua sen jälkeen, kun se ei ole ymmärtänyt käyttäjän syötettä. Se voi arpoa kysymyksen tietokannasta ja kysyä sen käyttäjältä. Keskusteluagentti voi myös kysyä käyttäjältä tietoa, kuten harrastusta, jota ei vielä ole tietokannassa. Näitä tietoja voi myös käyttää erikoisvastauksen muodostamisessa johdattaen keskustelua henkilöön itseensä. Jottei keskusteluagentti alkaisi toistaa itseään, käyttää se joskus erikoisvastauksena yleisiä kysymyksiä, kuten "mistä sinä haluaisit puhua"ja niin edelleen. Mikäli käyttäjä toistaa itseään, huomauttaa keskusteluagentti siitä, ja toisella toistokerralla muodostaa erikoisvastauksen.

Siinä vaiheessa, kun keskusteluagentti on onnistunut muodostamaan vastauksen joko lause- tai sanatietokannasta tai muodostanut erikoisvastauksen, tutkitaan sisältääkö syöte lipun tai korvausmerkin. Jos vastaus sisältää korvausmerkin ja keskusteluagentti on onnistunut parsimaan käyttäjän syötteestä sen tilalle sopivan sanan, korvataan tämä merkki sanalla. Lipullisista syötteistä poimitaan korvausmerkin kohdalla oleva sana lipun osoittamaan tietokenttään nimeksi, asuinpaikaksi, harrastukseksi tai kiinnostuksen kohteeksi. Tämän jälkeen vastaus annetaan tietoliikennerajapinnalle eteenpäin lähetettäväksi.

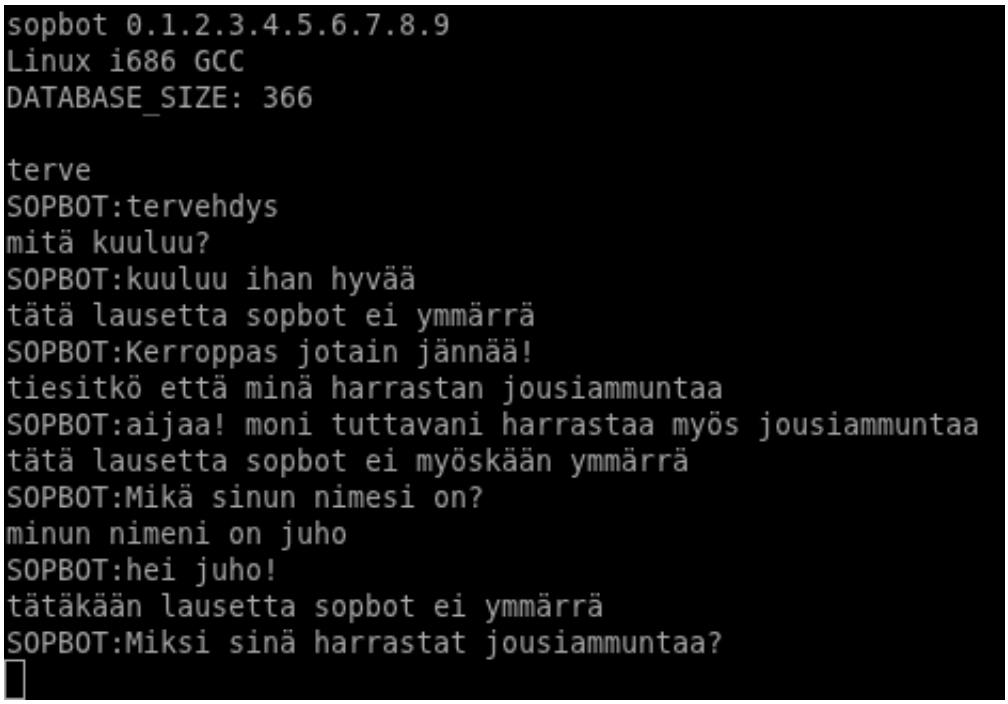
Kuvassa 8 on esitetty vastauksen muodostaminen tilakaavion avulla.



Kuva 8. Tilakaavio vastauksen generoinnista.

3.5. Käyttöliittymäkuvaus

Kuvassa 9 esitetään tietokoneella käytettävä käyttäliittymä, jossa käydään keskustelua työssä toteutetun SOPBOT-nimisen keskusteluagentin kanssa.



Kuva 9. Sopbotin keskustelukäyttöliittymä tietokoneella.

4. SAAVUTUKSET

Tässä luvussa tarkastellaan kandidaatintyössä toteutetun keskusteluagentin saavutukset, sekä esitellään keskusteluagentille suoritettavat testit ja tulokset. Lopuksi pohditaan saavutettuja tuloksia.

4.1. Toiminnallisuus

Työssä toteutettu keskusteluagentti toteuttaa sille asetetut vaatimukset. Se kykenee onnistuneesti keskustelemaan suomen kielellä ja oppimaan sekä muistamaan tietoja keskustelukumppaneistaan. Se oppii käytettyä kieltä keskustelun aikana ja käyttää oppimaansa tietoa keskustellessaan.

Keskusteluagentti hyväksyy pieniä kirjoitusvirheitä sekä taivutuseroja. Se löytää myös osumia lauseista, joissa on täytesanoja ja joiden sanajärestys on eri kuin tietokannassa olevalla lauseella. Tärkeiden sanojen painotus toimii sanatietokannassa. Jos käyttäjän syötteessä on avainsana, osuma yleensä ohjaa järkevään vastaukseen.

Keskusteluagentti oppii tietoja käyttäjältä ja osaa hyödyntää näitä tietoja keskustelun aikana. Myös korvausmerkin käyttö toimii, joten keskusteluagentti osaa poimia lauseesta vastauksessa hyödynnettävän sanan. Keskustelun aloituskyky ja ylläpito toimivat, joten keskusteluagentti osaa aloittaa keskustelun uudesta aiheesta. Se voi myös kysellä käyttäjältä henkilökohtaisia tietoja, ja käyttää niitä keskustelun avauksissa.

Oppiva tekoäly tallentaa keskustelijan antamia syötteitä, joita sen tietokannassa ei vielä ole, ja välillä sanoo niitä tallentaen useita erilaisia vastauksia opittuun lauseeseen. Keskusteluagentti ei usein toista itseään, koska samalle syötteelle on asetettu useita vastauksia.

4.2. Testaussuunnitelma

Keskusteluagentille suoritetaan kaksi eri tyyppistä testiä. Ensimmäinen on white-box-tyyppinen validointitesti ja toinen on black-box-tyyppinen vertaileva testi, jossa toteutettua keskusteluagenttia verrataan toiseen keskusteluagenttiin.

4.2.1. Validointitesti

Validointitesti suoritetaan liitteessä 1 olevan taulukon mukaisesti. Siinä on esitetty keskusteluagentille ohjelmoidut toiminnallisuudet, miten toiminnallisuuksia testataan, ja miten keskusteluagentin oletetaan niistä suoriutuvan. Tarpeen vaatiessa tarkastellaan testiohjelmalla keskusteluagentin käyttämiä sisäisiä muuttujia ja tiedostoja testien aikana.

Testien tulokset -luvussa esitetään kuinka keskusteluagentti suoriutui validointitestistä. Lisäksi kerrotaan erikseen miten keskusteluagenttiin toteutettu lause- ja sanatietokanta -malli toimi testeissä.

4.2.2. Vertaileva testi

Vertailevalla testillä tutkitaan sitä, kuinka toteutettu tekoäly suoritutuu vertailussa Jabberwacky-keskusteluagentin kanssa. Jabberwacky perustuu samaan oppivan tekoälyn malliin, mitä käytetään tässä työssä. Jabberwacky valittiin vertailutoteutukseksi siitä syystä, että se on harvoja suomen kieltä puhuvia keskusteluagentteja.

Vertailevan testin tarkoituksena on mitata, parantavatko työssä käytetyt menetelmät keskusteluagenttia suhteessa Jabberwacky-keskusteluagenttiin, vai riittääkö yksistään sen oppivuus tuottamaan hyvää keskustelua. Testissä annetaan viidelle eri henkilölle testattavaksi sekä työssä kehitetty keskusteluagentti että Jabberwacky. Testihenkilöitä pyydetään suorittamaan kahdenkymmenen viestin keskustelu molempien keskusteluagenttien kanssa. Testaajat käyttävät liitteessä 2 olevaa arviointilomaketta luokittelemaan vastaukset kategorioihin järkevä, outo ja järjetön. Lisäksi testaajia pyydetään arvioimaan kuinka hyvältä ja luonnolliselta käyty keskustelu tuntui asteikolla 1-5.

Tällä testillä saadaan molemmilta keskusteluagenteilta 100 viestiä, jotka on luokiteltu kolmeen eri luokkaan. Näistä kootaan tilastot testien tulokset -lukuun, jossa esitetään, missä suhteessä luokiteltuja viestejä on saatu. Samalla kerrotaan myös minkälaiset keskiarvot molemmat keskusteluagentit saivat testaajilta, kun arvioitiin koko keskustelua yleensä.

Viestien kategoriajakaumasta pitäisi näkyä, että kehitetty menetelmä, missä käytetään käyttäjäkohtaisia tietoja ja muutamia valmiiksi annettuja syöte/vastaus-pareja on vähintään yhtä hyvä, kuin Jabberwackyn kaltainen, vain käyttäjien syötteisiin perustuva malli. Vertailevassa testissä kerätyistä tuloksista pitäisi huomata, että työssä kehitetyn mallin kuuluisi tuottaa käyttäjien mielestä parempaa keskustelua, koska se ottaa muistiin käyttäjien tietoja ja käyttää näitä myöhemmin keskustelussa. Työn keskusteluagentti pyrkii myös harhauttamaan keskustelijaa, jos se ei tiedä mistä puhutaan, kun taas Jabberwacky vastaa aina parhaaksi kuvittelemallaan vastauksella.

4.3. Testien tulokset

Tässä luvussa esitellään testien tulokset ja arvoidaan niiden merkitystä. Validointitestistä kerrotaan, kuinka testitapauksista suoriuduttiin, ja vertailevasta testistä esitetään kaavio, joka kuvaa käyttäjien antamia arvioita.

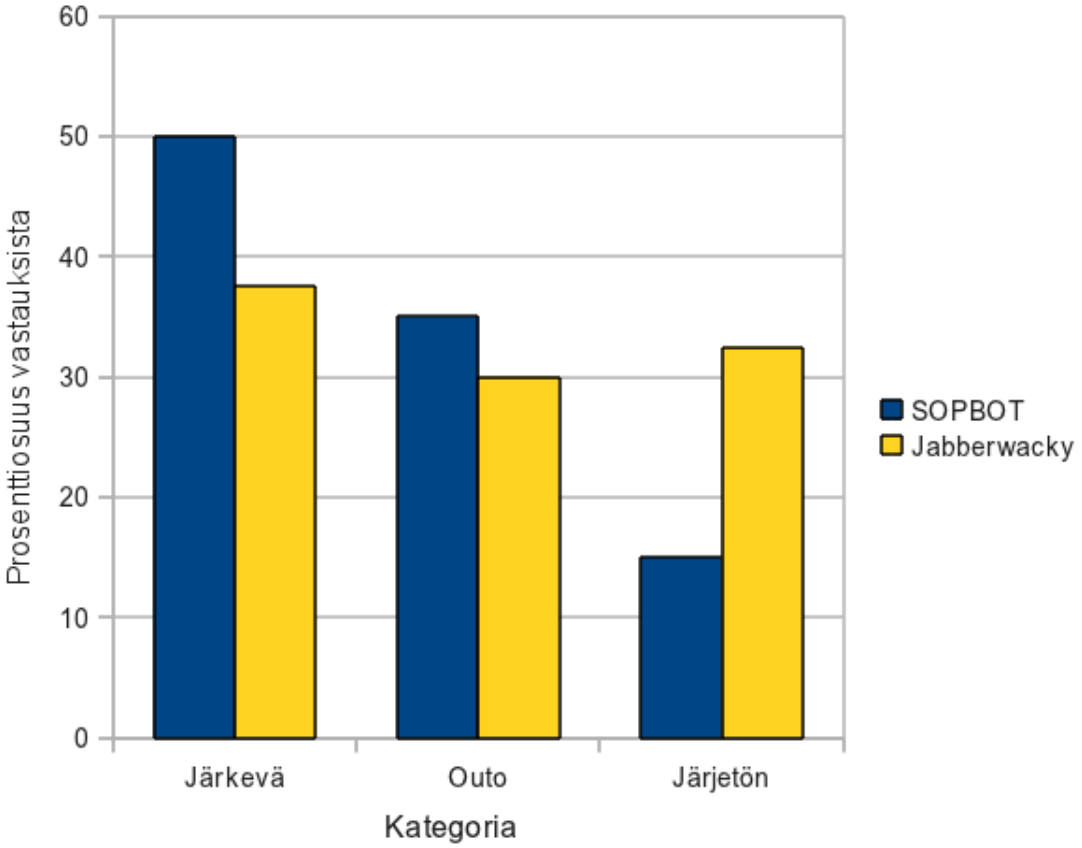
4.3.1. Validointitestin tulokset

Toteutettu keskusteluagentti läpäisi toiminnallisuudeltaan kaikki validointitestitapaukset. Testituloksista huomattiin, että pitkillä syötteillä sanatietokannan osumavaatimukseksi olisi hyvä valita suurempi ja lyhyillä syötteillä mahdollisesti pienempi, kuin työssä käytetty osumavaatimus kolme. Keskusteluagentin lauseiden arpomiskyky oli välillä puutteellinen johtuen heikosti toimivasta satunnaislukujen muodostajasta.

Sanatietokantamalli toimi hyvin löytäen aina osumat mikäli syötteen sana esiintyi tietokannassa riippumatta sanajärjestyksestä tai täytesanoista. Myös lausetietokantamalli toimi odotusten mukaisesti, ja onnistui löytämään osumia kirjoitusvirheistä huolimatta.

4.3.2. Vertailevan testin tulokset

Kuvassa 10 on esitetty vertailevan testin tulokset.



Kuva 10. Testin tulokset sadan kysymyksen otannalla.

Kaaviosta nähdään, että työssä toteutetun SOPBOT:in vastaukset olivat keskimäärin parempia, kuin Jabberwackyllä. Erityisesti järjettömien vastausten määrä on SOPBOT:lla huomattavasti pienempi. Tämä johtunee siitä, että se ei pyri väkisin tuottamaan vastausta, vaan yrittää harhauttaa keskustelijaa, mikäli järkevää vastausta ei löytynyt.

SOPBOT-keskusteluagentti saavutti keskustelun yleisarvosanaksi 2,5 asteikolla yhdestä viiteen, kun Jabberwacky sai samasta arviosta tulokseksi kaksi. Ero ei ole tilastollisesti kovin merkittävä, mutta on huomioitava, että Jabberwackyn tietokanta on paljon suurempi, kuin SOPBOT:n käyttämä. Suuremmalla tietokannalla SOPBOT:n voi olettaa saavuttavan selkeämmän eron.

4.4. Pohdinta

Toteutettu keskusteluagentti saavutti sille asetetut tavoitteet. Se suoriutui hyvin testeistä, ja kaikki ominaisuudet toimivat oletetulla tavalla. Koska keskusteluagentti ei ymmärrä asiayhteyttä, ei keskustelu samasta aiheesta onnistu, ellei keskustelija anna aiheeseen sopivia syötteitä. Keskusteluagentti on riippuvainen käyttäjän syötteiden laadusta, sillä

asiasisällöttömiin lauseisiin on hankala löytää keskustelun aiheeseen liittyvää vastausta.

Keskusteluagentti käyttää hyväkseen oppimiaan asioita keskustelijasta, mikä voi pitkässä keskustelussa olla huono asia, sillä tällöin keskusteluagentti alkaa toistamaan itseään kysellessään asioita henkilön kiinnostuksen kohteista, harrastuksesta ja niin edelleen.

Työssä toteutettu keskusteluagentti soveltuu hyvin viihdekäyttöön, koska sen oppimat lauseet rajoittuvat ainoastaan käyttäjäkunnan mielikuvituksen mukaan.

5. PROJEKTIN KUVAUS

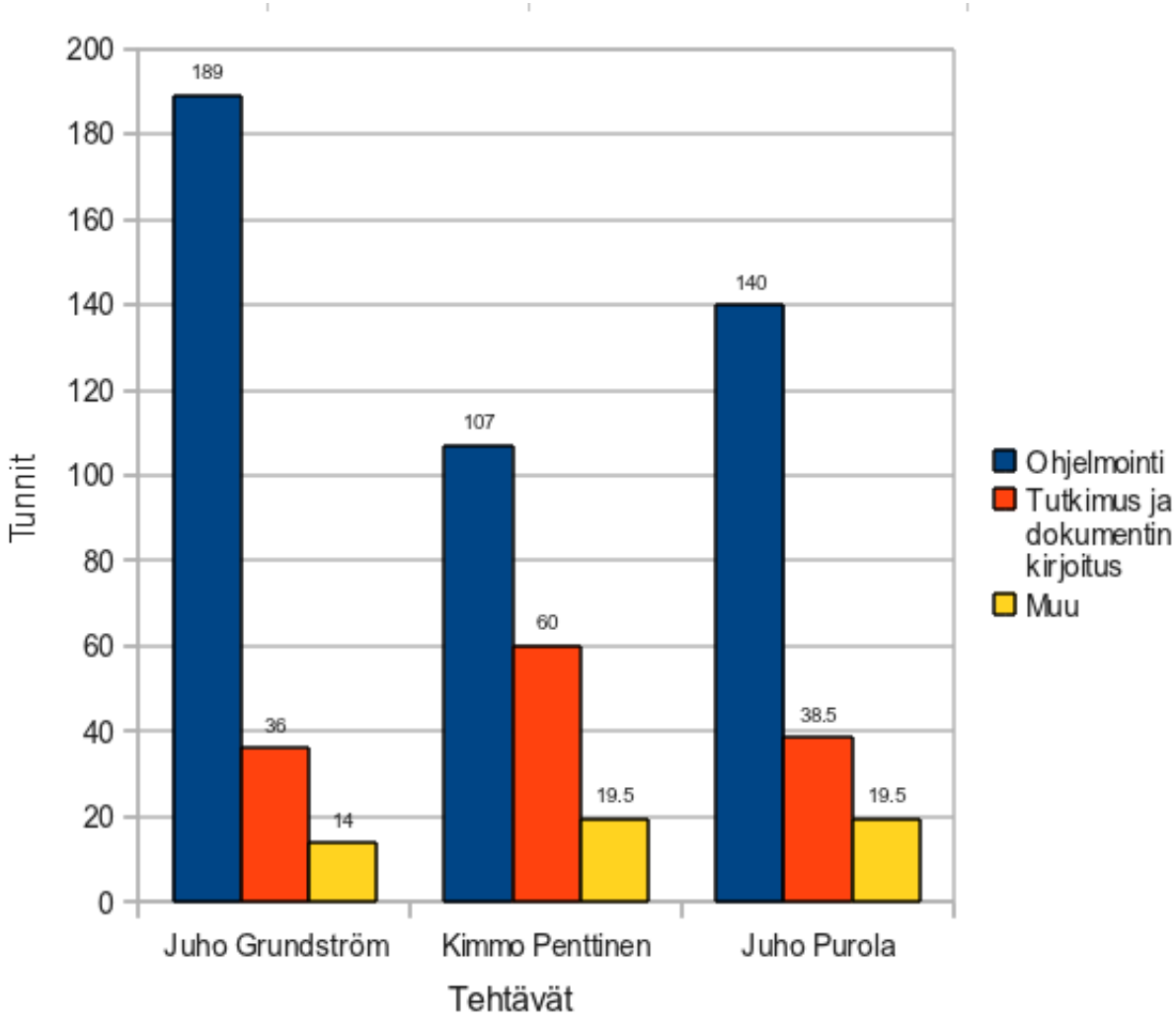
Tässä luvussa on esitetty ryhmän työnjako sekä ajankäyttö. Työn versionhallintana käytettiin Git-työkalua. Dokumentaation ja muun materiaalin jakamisessa hyödynnettiin Dropbox tiedostonjakopalvelua.

5.1. Työnjako

Työskentely on tapahtunut pääosin ryhmässä. Jokainen on osallistunut työn eri osa-alueiden tekemiseen.

5.2. Ajankäyttö

Kuvassa 11 on esitetty ryhmän jäsenten ajankäytön jakautuminen eri tehtäväalueisiin.



Kuva 11. Ryhmän ajankäytön jakautuminen eri tehtäväalueisiin.

Alla olevassa taulukossa 1 on koottuna käytetyt kokonaistuntimäärät.

Taulukko 1. Ajankäyttö

Henkilö	Tuntimäärä
Juho Grundström	239
Kimmo Penttinen	186,5
Juho Purola	198,5
Ryhmä	623,5

6. TULEVA KEHITYS

Tässä luvussa esitellään, miten kandidaatintyössä toteutettua keskusteluagenttia voisi edelleen kehittää. Parannusehdotukset on jaettu osa-alueittain, joissa kerrotaan kehityskohteet ja toteutusehdotukset.

6.1. Ihmismäisyyden parantaminen

Keskusteluagentin ihmismäisyyttä voidaan lisätä luomalla sille vaikutelma tunteista. Tällöin keskusteluagentti pystyy olemaan eri olotiloissa ja sen tuottamat vastaukset käyttäjien syötteisiin vaihtelevat olotilojen mukaan. Käyttäjillä olisi myös mahdollisuus joko loukata tai kehua keskusteluagenttia, mihin se reagoisi.

Toteutus on mahdollista lisätä työssä käytettyyn menetelmään. Tietokannassa voi yhdellä syötteellä olla useampia erilaisia vastaksia. Keskusteluagentin vastaukset asetetaan järjestykseen positiivisesta negatiiviseen ja vastaukset painotetaan jommalle kummalle puolelle keskusteluagentin olotilasta riippuen.

6.2. Tehokkuus

Siirtymällä lause- ja sanatietokannan käytöstä pelkästään sanatietokantaan, on mahdollista ottaa käyttöön malli, missä tietokannan sanat korvataan numeroilla. Kun yhteen sanakirjamaiseen tietokantaan kerätään kaikki käytetyt sanat ja niihin viitataan lauseissa numeroilla, niin merkkijonojen vertailu muuttuu numeroiden vertailuksi, mikä on huomattavasti kevyempi operaatio. Lisäksi muistin käyttö tehostuu, kun samoja sanoja ei esiinny useasti monissa sanatietokannan viesti-vastaus ryhmissä.

6.3. Luonnollisen kielen tuotto

Keskusteluagentin suomen kielen tuottoa voitaisiin parantaa siten, että se ei olisi täysin käyttäjien antamien uusien syötteiden varassa. Keskusteluagentille voisi opettaa suomen kielioppia asettamalla tietokantaan säännöt, miten suomen kielessä sanaluokat, kuten substantiivit, adjektiivit ja verbit, taipuvat liittämällä sanojen perusmuotojen loppuun eri päätteitä. Tällöin keskusteluagentin tuottaman luonnollisen kielen määrä kasvaisi, koska sen olisi mahdollista muuttaa käyttämiään vastauksia. Esimerkiksi sanatietokantahaussa voi löytyy paljon vastauksia, jossa on käytetty lauseesta vain monikkomuotoa. Tällöin keskusteluagentti pystyisi luomaan vastauksen yksikkömuodossa käyttämällä sanojen taivutussääntöjä.

6.4. Oppivuuden parantaminen

Toteutetussa mallissa ei ole varsinaisesti minkäänlaista asiayhteyksien oppimista, vaan keskusteluagentin oppivuus perustuu laajenevaan varastoon mahdollisia syöte/vastauspareja. Keskusteluagentin oppivuutta voidaan parantaa ottamalla käyttään tausta-luvussa

esitetty semanttinen verkko. Tällöin keskusteluagentille kertyy asiayhteyksien verkko, josta se pystyy tarvittaessa luomaan yhdenkin asiasanan perusteella järjellisiä vastauksia. Tämä edellyttää että kyseiseen sanaan on opittu joitain muita yhteyksiä.

Lauseiden merkitysten ymmärrystä voidaan myös saavuttaa asettamalla tietokannan lauseille asiayhteyslippuja. Samoja lippuja asetetaan samoihin aiheisiin liittyviin lauseisiin, ja mikäli keskusteluagentti ei ymmärrä mitä sille sanottiin, se yrittää jatkaa keskustelua edellisestä aiheesta.

6.5. Ryhmäkeskusteluun soveltuvuus

Nykyisellään keskusteluagentti vastaa kaikkiin sille tuleviin viesteihin. Ryhmäkeskusteluissa ominaisuus voi olla haitaksi, sillä kaikkiin viesteihin ei ole tarpeellista vastata. Yhden keskustelijan ei tarvitse olla koko ajan äänessä, ja muillakin täytyy olla mahdollisuus viedä keskustelua eteenpäin. Nykyiseen toteutukseen voidaan lisätä ominaisuus, että useammasta lähteestä tuleviin viesteihin ei aina vastata ellei niitä ole suoraan esitetty keskusteluagentille. Varsinkin, kun keskusteluagentti ei löydä ryhmäkeskustelussa tietokannastaan sopivaa vastausta niin vastaamatta jättäminen tähän viestiin ja sen tallentaminen omaan tietokantaan on paras vaihtoehto.

6.6. Keskusteluketjujen luominen

Jotta pystyttäisiin luomaan pidempiä ohjattuja keskusteluja, voidaan viestejä ketjuttaa toisiinsa. Tämän toteutuksena toimisivat esimerkiksi puumaiset tietorakenteet, jossa niihin on tallennettu osoite missä paikkaa tietokantaa käyttäjälle lähetettävä viesti on. Riippuen käyttäjältä saamasta vastauksesta haaraudutaan keskustelupuussa seuraavaan viestiin kunnes se on käyty loppuun. Tämä mahdollistaa käytävien keskustelujen laadun parantumisen, koska keskusteluagentti ei vain reagoi yksittäisiin viesteihin, jolloin keskustelua ei välttämättä edes synny.

7. YHTEENVETO

Kandidaatintyön aihealueena oleva keskusteluagentit ja tekoäly osoittautuivat erittäin mielenkiintoisiksi. Aiheen taustaan tutustuessa siitä löytyi paljon erilaisia lähtökohtia, toteutustekniikoita sekä jopa mielipide-eroja siitä, onko ajattelevan tekoälyn toteuttaminen yleensäkään mahdollista.

Tässä kandidaatintyössä tehtyyn toteutukseen valittu tekniikka pohjautuu kaavaansovitustekniikkaan ja oppivan tekoälyn malliin. Suunnitteluvaiheessa kehitettiin kaavaansovitustekniikasta tähän työhön paremmin soveltuva malli, jossa käytetään sekä sana- että lausetietokantaa. Tietokanta ei perustu aikaisempiin totetutuksiin, vaan se kehitettiin itse. Näin tietokantaan saatiin laitettua monia ominaisuuksia, joita voitiin hyödyntää ohjelmassa.

Oppiva tekoäly lisättiin toteutukseen, jotta ohjelman tietokanta kehittyisi jatkuvasti keskustelujen aikana, koska riittävän laajan tietokannan kirjoittaminen on erittäin haastava tehtävä. Oppivasta tekoälystä johtuen keskusteluagentti voi oppia myös muita kieliä.

Useiden erilaisten toteutustapojen yhdisteleminen osoittautui toimivaksi, koska toteutuksessa pystyttiin hyödyntämään jokaisen tavan hyviä puolia. Tehtyä keskusteluagenttia verrattiin Jabberwacky-keskusteluagenttiin, ja testituloksista huomattiin, että useampaa toteutusmenetelmää hyödyntävä keskustelutekoäly tuottaa laadukkaampaa keskustelua, kuin pelkästään oppivaan tekoälyyn perustuva keskusteluagentti.

Työssä tehdyn keskusteluagentin toteutustapa mahdollistaa hyvin tulevan jatkokehityksen sulautettuun ympäristöön, sillä suunniteltuja ominaisuuksia jäi vielä toteuttamatta.

8. LÄHTEET

[1] Deryugina O.V (2010). *Chatterbots. Scientific and Technical Information Processing 2, s.143-147.*

[2] Internet relay chat protocol (luettu 9.2.2011). URL: `http://tools.ietf.org/html/rfc1459#section-1`.

[3] Weizenbaum J (1966). *ELIZA - A Computer Program For the Study of Natural Language Communication Between Man And Machine. Communications of the ACM 1, s. 36-45.*

[4] Turing AM (1950). *Computing machinery and intelligence. Mind 59, s. 433–460.*

[5] Home page of the loebner prize in artificial intelligence (luettu 8.2.2011). URL: `http://www.loebner.net/Prizef/loebner-prize.html`.

[6] *AIML: Artificial Intelligence Markup Language (luettu 9.2.2011).* URL: `http://www.alicebot.org/TR/2005/WD-aiml`.

[7] *About the Jabberwacky AI (luettu 9.2.2011).* URL: `http://www.jabberwacky.com/j2about`.

[8] Russell S.J & Novrig P (2003). *Artificial Intelligence: A Modern Approach (2nd ed.).*

[9] Revonsuo A & Lang H & Aaltonen O (1999). *Mieli ja aivot. Kognitiivinen neurotiede.*

[10] Karlsson F & Koskenniemi K (1985). *A process model of morphology and lexion. Folia Linguistica 29, s. 207–231.*

[11] Shawar B.A & Atwell E (2007). Different measurements metrics to evaluate a chatbot system. proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies, s. 89-96.

[12] Maxime Crochemore & Thierry Lecroq (1996). *Pattern-Matching and Text-Compression Algorithms. Computing Surveys 1, s. 38-41.*

[13] Donald E. Knuth James H. Morris Jr. & Vaughan R. Pratt (1977). *Fast Pattern Matching in Strings. SIAM journal on Computing 2, s. 323–350.*

[14] Ken Thompson (1968). *Programming Techniques: Regular expression search algorithm. Communications of the ACM 6, s. 419-422.*

[15] Patrick A.V. Dowling & Hall Geoff R (1980). *Approximate String Matching. Computing Surveys 4, s. 381-402.*

[16] *Markov Chains chapter in American Mathematical Society's introductory probability book (luettu 2.5.2011).* URL: `http://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/Chapter11.pdf`.

[17] Jason L. Hutchens and Michael D. Alder (1998). *Introducing MegaHal. NeMLaP3/CoNLL98 Workshop on Human Computer Conversation ACL, s. 271-274.*

[18] *Developing AI chatbots (luettu 10.2.2011).* URL: http://www.codeproject.com/KB/scrapbook/robomatic.aspx.

[19] Gonzales Cenelia (2010). *Tutorial on making an Artificial Intelligence Chatbot (luettu 10.2.2011).* URL: http://www.codeproject.com/KB/recipes/bot_tutorial.aspx.

[20] *Skynet-AI (luettu 3.5.2011).* URL: http://www.chatbots.org/chatbot/skynet-ai/.

[21] *YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia (luettu 3.5.2011).* URL: http://www.mpi-inf.mpg.de/yago-naga/yago/.

[22] W. Duch J. Szymanski ja T. Sarnatowicz (2007). *Towards Avatars with Artificial Minds: Role of Semantic Memory. Journal of Ubiquitous Computing and Intelligence 1, s.1-11.*

[23] Case J & Fedor M & Schoffstall M & Davin J (1990). *A Simple Network Management Protocol (luettu 9.2.2011).* URL: http://www.ietf.org/rfc/rfc1157.txt.

[24] Elektor internet radio version 1.0 hardware manual (luettu 8.2.2011). URL: http://www.ethernut.de/pdf/eir_um_1_0_6.pdf.

[25] Sop/hardware (luettu 8.2.2011). URL: https://www.raippa.fi/SOP/Hardware2010.

[26] Levenštein V.I (1966). Binary codes cabale of correcting deletions, insertions, and reversals. soviet physics-doklady 8, s.707-710.

9. LIITTEET

Liite 1: Validointitestitapaukset
Liite 2: Kaavake keskusteluagentin arvioimiseen

Liite 1

Validointitestitapaukset

Mitä testataan	Miten testataan
Muistiominaisuutta ja korvausmerkin toimivuutta	Annetaan erilaisia syötteitä muodossa: Minä olen Harri Minä harrastan jääkiekkoa
Odotettu lopputulos	Keskusteluagentin tulisi pystyä poimimaan ja tallentamaan haluttu sana. Esimerkiksi Harri nimikenttään tai jääkiekkoa harrastuskenttään.

Keskustelunaloituskykyä	Käynnistetään keskusteluagentti eikä anneta sille viestejä.
Odotettu lopputulos	Keskusteluagentin tulisi omatoimisesti aloittaa keskustelu muiden kanssa sille asetetun viiveen kuluttua.

Keskustelun ylläpito-ominaisuutta	Annetaan keskusteluagentille viestejä, joissa on annettu keskustelijasta tietoja ja lopetetaan viestin antaminen.
Odotettu lopputulos	Viestien tulon loputtua keskusteluagentin tulisi alkaa lähettää viestejä sille asetetun ajan sisällä käyttäen tietoja sen kanssa keskustelleesta henkilöstä. Esimerkiksi “miten menee Harri”, jos nimi on kerrottu.

Keskusteluagentin vastauskykyä yksinkertaisiin syötteisiin.	Annetaan keskusteluagentille joukko erilaisia syötteitä, joiden tiedetään olevan sen tietokannassa.
Odotettu lopputulos	Keskusteluagentin viestien tulisi vastata tietokannassa olevia syöte/vastaus pareja.

Liite 1

Keskusteluagentin lause- ja sanatietokantaan suoritettavien hakujen toimivuutta	-Lausetietokanta:Syötetään keskusteluagentille viestejä, joiden tiedetään olevan tietokannassa. Näihin viesteihin lisätään pieniä kirjoistusvirheitä. -Sanatietokanta: Syötetään keskusteluagentille viestejä, joiden sanoja tiedetään olevan useissa tietokannan lauseissa.
Odotettu lopputulos	-Lausetietokanta: Keskusteluagentin viestien tulisi vastata tietokannassa olevia syöte/vastaus pareja huolimatta pienistä kirjoitusvirheistä. -Sanatietokanta: Keskusteluagentin tulisi valita lause joissa syötteen sanat esiintyvät eniten.

Keskusteluagentin kykyä selviytyä tunnistamattomista viesteistä.	Annetaan keskusteluagentille joukko erilaisia viestejä, joita ei ole sen tietokannassa.
Odotettu lopputulos	Keskusteluagentin tulisi poimia nämä viestit talteen ja vastata niillä myöhemmin muille. Saadut vastaukset tallennetaan kysytyn viestin vastauksiksi. Keskusteluagentin tulisi muodostaa vastaus myös tuntemattomalle syötteelle joko syöttämällä sille satunnainen vastaus tietokannasta tai käyttämällä käyttäjän aikaisempia antamia tietoja itsestään hyväksi.

Keskusteluagentin kyky huomata viestien toisto	Annetaan keskusteluagentille joukko peräkkäisiä samoja viestejä.
Odotettu lopputulos	Keskusteluagentin tulisi huomata toisto ja mainita tästä.

Liite 2

Kaavake keskusteluagentin arvioimiseen

Ohjeet:
Käy 20:n viestin keskustelu keskusteluagentin kanssa. Arvioi jokainen keskusteluagentin antama viesti alhaalla olevaan taulukkoon laittamalla rasti kyseiseen kenttään. Vastaukset arvioidaan asteikolla järkevä, outo tai järjetön.

Lopuksi arvioi kuinka hyvä käymäsi keskustelu oli asteikolla 1-5. Missä 1 tarkoittaa, että keskusteluagentin kanssa oli ikävä jutella eikä varsinaista keskusteluyhteyttä juurikaan syntynyt ja 5 tarkoittaa, että keskustelu oli hyvä ja tuntui luonnolliselta.

Vastaus nro	Järkevä	Outo	Järjetön
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			
11.			
12.			
13.			
14.			
15.			
16.			
17.			
18.			
19.			
20.			

Keskustelun yleisarvosana:

1	2	3	4	5