

23.5.2015

Kotipsykiatri - Dokumentaatio

Aapo Kokko

293383

EST

Vuosikurssi 2012

23.5.2015

Yleiskuvaus

Ohjelma keskustelee interaktiivisesti käyttäjän kanssa psykiatrin tavoin. Kotipsykiatri muistaa keskusteluhistorian ja reagoi käyttäjän lauseisiin. Se tekee lisäkysymyksiä käyttäjän edellisen vastauksen ja muun keskusteluhistorian pohjalta (sekä antaa neuvoja). Epäselvissä tapauksissa psykiatri pyytää selvennystä ja reagoi myös esim. huutomerkin ja kysymysmerkin käyttöön.

Ohjelma on tehty helppo-keskivaikea tasoisena.

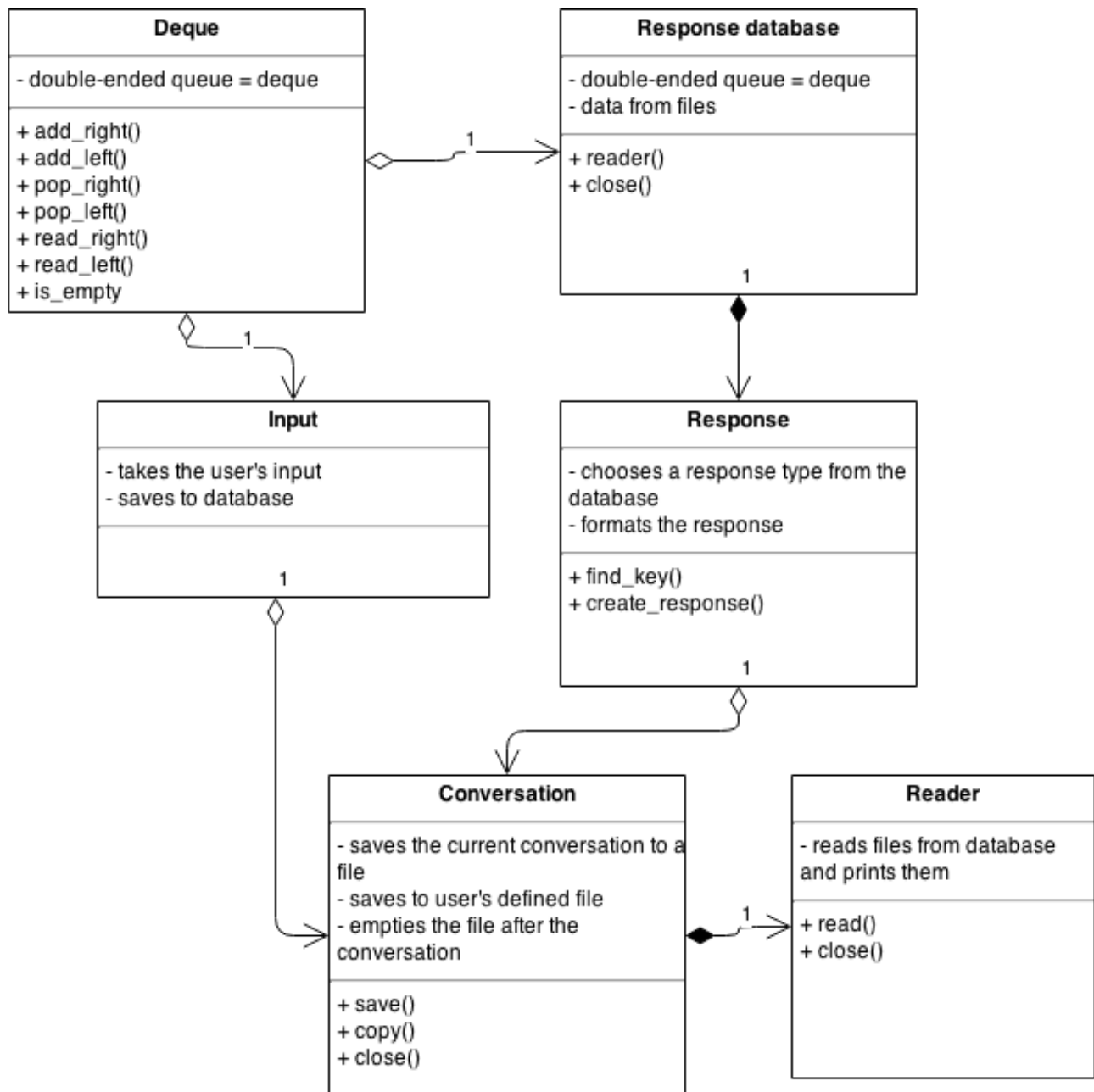
Käyttöohje

Ohjelma kertoo itse mikä se on ja antaa kolme vaihtoehtoa sille mitä käyttäjä haluaa tehdä. Käyttäjä kirjoittaa joko 1, jos hän haluaa keskustella psykiatrin kanssa, 2, jos hän haluaa lukea edellisiä keskusteluja, tai 3, jos hän haluaa sammuttaa ohjelman, ja painaa lopuksi enteriä.

Keskustellessa psykiatrin kanssa, psykiatrille voi puhua kirjoittamalla haluamansa asian ja painamalla enteriä, jolloin psykiatri vastaa. Keskustelun voi päättää käyttämällä sanaa goodbye syötteessä. Tämän jälkeen ohjelma kysyy haluaako käyttäjä tallentaa keskustelun (vaihtoehto 1) tai lopettaa tallentamatta (vaihtoehto 2). Tallentaessaan ohjelma pyytää käyttäjää antamaan tiedostonimen (=keskustelun nimen) johon hän haluaa keskustelun tallentaa. Lopuksi ohjelma taas kysyy mitä käyttäjä haluaa tehdä (1, 2 tai 3).

Vaihtoehdon kaksi valitessa ohjelma antaa listan tiedostojen nimiä (eli keskusteluiden nimiä) joista käyttäjä voi valita yhden luettavaksi kirjoittamalla nimen ja painamalla enteriä. Tämän jälkeen kysyy haluaako käyttäjä lukea edelleen keskusteluja (kirjoittaen 1) tai lopettaa keskustelujen lukemisen (kirjoittaen 2). Jos käyttäjä päättää jatkaa, tulee taas nimivaihtoehdot esiin, mutta jos hän päättää lopettaa ohjelma taas kysyy mitä käyttäjä haluaa seuraavaksi tehdä (1, 2 tai 3).

Ohjelman rakenne



Response database ja Input suurimmaksi osaksi ovat tietorakenteita jotka perustuvat deque luokkaan. Deque luokka on pakka –tietorakenne jota voidaan käsitellä sekä vasemmalta (pakan päältä) että oikealta (pakan alta). Response database lukee tietoja tekstitiedostosta ja kirjoittaa tiedot pakkaan väliaikaista käyttöä varten. Inputin pakka tallentaa keskustelun käyttäjän syötteet väliaikaiseen käyttöön.

Response hyödyntää käyttäjän syötteitä ja response databasea ja luo niiden perusteella vastauksen käyttäjän syötteeseen.

Conversation tallentaa keskustelun väliaikaiseen tiedostoon ja kopioi väliaikaistiedoston keskustelun käyttäjän halutessa uuteen tiedostoon myöhempää tarkastelua varten. Conversation pitää myös yllä listaa tallennetuista keskusteluista. Reader lukee vanhoja keskusteluja.

23.5.2015

Algoritmit

Ohjelman tekoäly on suhteellisen alkeellinen ja se perustuu valmiiden lauserakenteiden käyttöön. Tekoäly ottaa käyttäjän syötteitä ja kääntää ne psykiatrin tavoin kysymyksiksi tai vastaa muuten ympäripyöreesti antaen älyn illuusion. Käyttäjän syötteistä etsitään avain sanoja tai merkkejä, joiden avulla psykiatrin vastaus muokkaantuu. Vastausmallit löytyvät tekstitiedostoista jotka lisätään ja otetaan tietokannasta random-generaattorin avulla, jotta keskustelu ei mene aina samalla kaavalla, vaan vaihtelee kerrasta toiseen mahdollisimman paljon.

Tietorakenteet

Ohjelmassa käytettävä tieto säilytetään tekstitiedostoissa ja ohjelman käynnin aikana osittain deque-tietokannoissa. Koska ohjelma on täysin tekstipohjainen ja perustuu vain ja ainoastaan tekstinkäsittelyyn, on tekstitiedostojen käyttö looginen. Deque-tietokannat on valittu, koska niihin voi sijoittaa ja niistä voi ottaa tietoa molemmista päistä rakennetta.

Tiedostot

Ohjelma käyttää tekstitiedostoja joita on helppo muokata ja niiden kautta tekoäly on osittain muokattavissa ja sitä voidaan monipuolistaa.

Testaus

Suurin osa ohjelman testaamisesta on tehty main-funktion avulla. Tämä johtuu siitä että ohjelman runko eli luokat ovat hyvin pieniä ja niiden ominaisuuksia on helpointa testata yhdessä kokonaisuutena. Main-funktion avulla oli helppo kokeilla ohjelman ominaisuuksia ja löytää korjattavaa niiden ominaisuuksissa. Toimivuutta pystyi arvioimaan tulostuksista ja tiedostoissa tapahtuneista muutoksista.

Ohjelman tunnetut puutteet ja viat

Tekstitiedostojen kanssa on ongelmia. Ohjelma jostain syystä lukee vain joka toisen rivin ja olen siksi ne kirjoittanut sellaiseen muotoon ja käsittelen niitä kyseisellä tavalla. Sen takia myös saattaa joistain tiedostoista löytyä ”turhaaaa” –rivi. Tekstitiedostoihin ei voi myöskään tallentaa useampaa keskustelua kerrallaan, mikä on sellainen ongelma johon en kerennyt paneutua.

En kerennyt miettiä/kokeilla kaikkia virhetilanteita ja ohjelma saattaa käyttäytyä oudosti tai antaa virheitä, jos käyttäjä ei noudata ohjeita syötteissään. Ohjelma myöskään silloin anna ohjeita vääränlaisesta syötteestä.

Tekoäly on edelleen kömpelö eikä osaa vastata aiheen mukaan. Malleja tulisi lisätä ja monipuolistaa ylipäättänsä mallivalikoimaa.

23.5.2015

Ohjelma ei kysy käyttäjän nimeä ja syy miksi en lopulta tätä toteuttanut on, koska en kerennyt toteuttaa nimen käyttöä psykiatrin vastauksissa.

3 parasta ja heikointa kohtaa

3 parasta:

Ohjelma löytää käyttäjän syötteistä avainsanoja ja pystyy pilkkomaan ja hyödyntämään syötteitä. Tätä ominaisuutta on helppo laajentaa.

Käyttämällä erillisiä tekstitiedostoja eri avainsanoille voi niiden valikoimaa laajentamalla parantaa helposti ohjelman tekoälyä.

Ohjelma käyttää random-lukuja tietojen siirtämisessä tietorakenteisiin ja sieltä pois otettaessa. Tämä ominaisuus on laajennettavissa ja paranneltavissa pidemmälle.

3 heikointa:

Tekoäly on edelleen kömpelö ja ei reagoi keskustelun aiheisiin.

Psykiatrin vastaukset noudattavat hyvin paljon samantyyppistä kaavaa.

Ohjelma osaa käyttää käyttäjän syötteitä vain vastausten lopussa.

Poikkeamat suunnitelmasta

Luokissa tapahtui muutoksia. Tietorakenteihin tarvitsin erillisen deque-luokan jota, kutsuin tietorakenne luokkiin. Input databasea en tarvinnutkaan erikseen vaan lisäsin sen Inputin sisään ja sama tapahtui Conversation databasen kohdalla eli lisäsin sen Conversationin sisään. Inputin metodeja siirtyi paljon Responseen ja Response –luokalla on ylipäättänsä suuri paino ohjelmassa.

Aikataulussa en pysynyt alkuunkaan ja en osannut arvatakaan kuinka paha kiire minulla on muiden kouluhommien kanssa ja muutenkin henkilökohtaisessa elämässä. Työjärjestys muuttui siinä mielessä että ohjelman rungon sain koodattua yhdessä illassa ja luokat olivat pääpiirteissä hyvin nopeasti valmiit. Suurin osa lopun työstä oli main funktion parissa.

Hyödynsin tekstitiedostoja paljon enemmän kuin alkuun suunnittelin ja totesin ne parhaaksi vaihtoehdoksi tietojen tallennuksessa. Deque toimi hyvin väliaikaisessa käytössä, mutta tekstitiedostot osoittautuivat äärimmäisen hyödyllisiksi.

Suunnittelin graafisen käyttöliittymän ja debug-moden tekoa lopussa, jos on aikaa, jotai arvioin alunperin olevan. En kuitenkaan kerennyt kumpaakaan toteuttaa.

23.5.2015

Toteutunut työjärjestys ja aikataulu

Alkuperäinen aikataulu:

5.-12.3.2015 : Suunnitelmademot

1.4.2015 : Tähän mennessä tietorakenteet

- ei välttämättä kaikkia lauseita valmiiksi response databasesessa, mutta toimivat tietorakenteet
- testausta
- dokumentti niillä osin valmiiksi mihin tähän mennessä pystyy, ainakin yleiskuvaus, ohjelman rakenne, tietorakenteet, tiedostot (ainakin osittain) ja testaus tähän mennessä

15.4.2015 : Tähän mennessä Input ja Response ja toimiva main-funktio keskustelun toimivuuden ja tekoälyn testaamiseen

- tekoälyn kehitystä ja response datan kasvattamista
- dokumentin täydennys

1.5.2015 : Tähän mennessä keskustelujen tallennus ja Reader

- toimivat ominaisuudet -> vaatii ehkä viimeistelyä ja testailua (ohjelman "rikkomista")
- dokumentin täydennys

2.5-15.5 : Mikäli pysynyt aikataulussa, graafisen käyttöliittymän koodaaminen ja debug-mode

- tärkeintä kuitenkin tekoälyn hiominen ja ohjelman toimivuus
- dokumentti loppuun, erityisesti ohjelman tunnetut puutteet ja viat, 3 parasta ja 3 heikointa kohtaa, poikkeamat suunnitelmasta (ainakin loppuun, jos aloitettu), toteutunut työjärjestys ja aikataulu ja arvio lopputuloksesta

15.5.2015 : Projektin ja projektidokumentin DL

18.-22.5.2015 : Projektidemot

23.5.2015

Toteutunut työjärjestys:

- 18.4 – 19.4 ohjelman runko eli luokat pääpiirteissään + dokumentaatio
- 8.5 lisää luokkien tekoa
- 12.5 – 14.5 lisää luokkia + dokumentaatio
- 16.5 – 17.5 tekoälyn suunnittelua, main funktion aloitus
- 20.5 main funktio, keskustelu
- 21.5 main funktio, aiemmat keskustelut
- 22.5 – 24.5 main funktio kokonaisuudessaan ja testausta + dokumentaatio

Arvio lopputuloksesta

Toteutus suppea, ei graafista käyttöliittymää tai debug-modea, tekoäly ei niin älykäs kuin alkuun suunniteltu. Tekee suurinpiirtein mitä vaaditaan, mutta keskustelun realistisuus jää heikoksi ja tekoäly ei reagoi tarpeeksi laajasti käyttäjän syötteiden aiheisiin ja sanoihin. Vastausmallit liian ympäripyöreitä ja kiireen näkee siinä.

Ohjelman runko on kuitenkin selkeä ja siihen on helppo lisätä ja sitä on helpohko kehittää eteenpäin. On monta muuttujaa millä parantaa tekoälyä kuten random-generaattorin laajentaminen, avainsanojen lisääminen ja siten tietokantojen lisääminen sekä mallilauseiden monipuolistaminen.

Viitteet

Python, yleistä:

<http://www.tutorialspoint.com/python/index.htm>

<https://docs.python.org/3.3/index.html>

<https://docs.python.org/3.3/tutorial/datastructures.html>

<https://docs.python.org/3.3/tutorial/inputoutput.html>

<https://docs.python.org/3.3/library/string.html>

http://www.tutorialspoint.com/python/file_truncate.htm

<https://docs.python.org/2/library/shutil.html>

23.5.2015

<https://docs.python.org/2/library/random.html>

<http://www.cyberciti.biz/faq/howto-get-current-date-time-in-python/>

http://www.tutorialspoint.com/python/string_find.htm

Double-ended queue/deque:

<http://interactivepython.org/XikcZ/courselib/static/pythonds/BasicDS/deques.html>

http://scanfree.com/Data_Structure/duble-ended-queue-dequeue

Liitteet

1. Lähdekoodi (.py)
 - a. deque
 - b. response_database
 - c. response
 - d. input
 - e. conversation
 - f. reader
 - g. debug-mode (jäännö)
 - h. main
2. Tarvittavat tekstitiedostot (.txt)
 - a. files (vanhojen keskustelujen nimet)
 - b. conversation (keskustelun väliaikaistiedosto)
 - c. response_data1
 - d. response_data2
 - e. response_data3
 - f. response_data4
 - g. response_data5
 - h. testi (hyvin lyhyt testikeskustelu, luettavissa ohjelmalla)