DEGREE PROJECT, IN COMPUTER SCIENCE , FIRST LEVEL
STOCKHOLM, SWEDEN 2015

Chatbot with common-sense database

MATTIAS AMILON

KTH ROYAL INSTITUTE OF TECHNOLOGY

CSC SCHOOL

Mattias Amilon

Royal Institute of Technology
DD143X, Bachelor’s Thesis in Computer Science
Supervisor: Pawel Herman

May 10, 2015

Abstract

In 1950 Alan Turing introduced the famous “Turing test” which tests if a machine can be as intelligent as a human by testing if it can communicate with a person in a “human” way. Inspired by this test, numerous so called chatbots, in the form of computer programs, that manage a written dialogue have been created.

A so called common-sense database consists of data that most humans would know and consider as common knowledge, something that computers generally do not know very much about.

This report describes the process of an attempt to implement a simple chatbot using the common-sense database ConceptNet. The behaviour, or the human-likeness, of this chatbot was then compared to that of the classic chatbot ELIZA and the 2008 Loebner prize winning chatbot Elbot, through a series of user tests.

The results indicate that using a common-sense database for a chatbot shows some promise for further investigation.

Referat

Redan 1950 publicerade Alan Turing det berömda “turingtestet”, testet går ut på att undersöka om en maskin är lika intelligent som en människa genom att testa om den kan kommunicera med en person på ett mänskligt sätt. Som ett resultat av detta test har en hel rad olika chatbotar, i form av dataprogram som kan föra en skriven dialog, skapats.

En så kallad “sunt förnuft”-databas innehåller sådan information som de flesta människor känner till och anser vara vardagligt vetande, något som datorer normalt sett inte vet någonting om.

Genom den här rapporten beskrivs implementationen av en enkel chatbot som använder sig av “sunt förnuft”-databasen ConceptNet. Hurpass människolikt denna chatbot sedan kan föra en konversation jämfört med den klassiska chatboten ELIZA och 2008 års Loebner-pris vinnare Elbot undersöks genom en serie användertester.

Resultatet visar positiva tendenser för fortsatt undersökning av området.

Table of contents

1. Introduction **1**

1.1 Objectives and scope 2

1.2 Outline 2

2. Background **3**

2.1 Chatbots - history and examples 3

2.2 NLP - interpreting natural language 4

2.3 The Stanford Parser - parsing natural language 5

2.4 Common-sense database - ConceptNet 7

3. Methods **10**

3.1 Implementing the chatbot - a simple approach 10

3.2 User tests - setup 11

4. Results **13**

5. Discussion **15**

5.1 Future research 15

6. Conclusions **16**

References **17**

1. Introduction

A so called chatbot is a computer program which can engage in a written dialogue with a user. The practical use of this today is, for example, in applications on companies' web pages offering support to users of their different services or products. A chatbot typically works as such that it reads and analyzes a user's natural language input, and then uses some kind of algorithm, more or less advanced, to give a suitable answer to the given input. One of the first examples of such a program is "ELIZA", created in 1966 by Joseph Weizenbaum. The basic idea of ELIZA is to recognize a keyword in the user input and then produce an answer containing this keyword[1]. A typical example of ELIZA's behaviour is that it on the user input: "I have a problem with my mother" would answer: "Tell me more about your mother."

In 1950 Alan Turing published the famous Turing test, which is a test on how well - or how human-like - a machine, or in this case a computer program, can communicate and express itself in a dialogue with a real person. For the machine to pass the test, the person chatting with it should not be able to determine if it is another person or a machine[2]. Every year there is a competition held for chatbots where the so called Loebner prize is awarded to the chatbot that is considered as the most human-like.

A common-sense database is a database that contains information that most humans would consider to be common knowledge, something that computers generally are not very good at, an example of such a common-sense database is ConceptNet which was created as project at the Massachusetts Institute of Technology[3].

Throughout this report it is investigated how convincingly, or how human-like, a simple chatbot using the common-sense database ConceptNet can engage in a written dialogue with a user. How to measure this is not trivial since the definition of human-like behaviour can always be considered to be subjective. Here, this was measured through user testing, a number of testers tried to chat with the chatbot and were then asked to grade how human-like on a scale from 0 to 100, where 0 was "not at all like a human" and 100 was "like chatting with a human", she or he considered the chatbot. In addition to this, each tester was asked to answer a small questionnaire where she or he, on the same scale from 0 to 100, graded the chatbot regarding a number of other categories that could be considered as human-like properties. To put the results into some perspective each tester also tested and answered the same questions for the chatbot ELIZA, and the 2008 Loebner prize winning chatbot Elbot, created by Fred Roberts and the company Artificial Solutions[4]. The test results for each chatbot were then compared as to draw some conclusions from this.

1.1 Objectives and scope

The problem that this report focuses on could be formulated as:

Could using a common-sense database improve the behaviour of a chatbot when considering human-likeness?

This is tested through the implementation of a simple chatbot using the common sense database ConceptNet, the behaviour is then evaluated and compared to two other chatbots, ELIZA and Elbot, through user testing. The scope of this is to determine whether applications that rely on user interaction could benefit from using a common-sense database to better understand a user input, and to be more natural in its response.

1.2 Outline

The report consists of four bigger parts, first follows a shorter section with the objective- and formulation of the problem that this project focuses on and tries to answer. Thereafter follows a background section, where some earlier chatbots are studied and analyzed, other tools and knowledge necessary are also described in this section. Then follows a section on the implementation of the actual chatbot. Finally, the results and conclusions are presented in the last section.

2. Background

This section is focused on analyzing how some existing chatbots are implemented and what their key concepts are. Moreover the field of natural language processing (NLP) is looked into as it is an important feature in how to parse and analyze natural language. The structure of ConceptNet is also described, as well as The Stanford Parser, which is a powerful tool to actually parse natural language, which will be needed to parse the user input.

2.1 Chatbots - history and examples

In the field of computer science the implementation of intelligent machines that can communicate using natural language in a human way has been studied for a relatively long time. As early as 1950 Alan Turing published the famous Turing test, which is a test on how well - or rather how human-like - a machine can communicate with a real person, and as such gives some measure of the machine's intelligence. A machine is said to pass the test if a person talking to it can not determine whether she or he is talking to a another human being or a machine[2]. An implementation of such a machine could of course be what is today known as a chatbot. Here follows a short description of three different chatbots, that in different ways have added new ideas to the field, and their basic structures.

An early example is the chatbot ELIZA created by Joseph Weizenbaum in 1966. ELIZA answers some user input by looking for a keyword in this input and then constructing an answer that would generally be fitting for this keyword. This simple idea works quite well is instructed to talk to ELIZA as if it was a psychiatrist[1].

In the middle of the 1990s the American engineer Richard Wallace wrote a chatbot that he called Alice (Artificial Linguistic Internet Computer Entity). Alice is reminiscent of ELIZA in its basic structure. Although Wallace noticed that a great part of the language that is commonly used in everyday speech is in fact a quite small part of the language as a whole, this made it possible to hardcode answers to a great deal of the most common user input[5].

A later example is the 2008 Loebner prize winning chatbot called Elbot, created by Fred Roberts and the company Artificial Solutions to, for what it seems, promote their technology in communication and artificial intelligence[4]. Given that this is a competitive company it is hard to find information that in detail describes how Elbot is implemented, but some interesting facts about the program's structure can be found on the company's web page. Where ELIZA only looks at keyword in a given user input, Elbot also looks for synonyms and understands that the meaning of two synonyms is the same. Elbot also looks for certain common expressions, word combinations and other patterns in the language of the user input, and then gives an answer based on this[6].

These are just three examples of many chatbots through history, but it is possible to see that they follow some kind of evolution, where the newer is more intelligent than the older in that it adds more features to try to improve the behaviour to be more natural. The main difference seems to lie in the way that the user input is processed and interpreted. How a computer program process and interprets natural language is a study that goes under the name NLP and is described more closely in the next section.

2.2 NLP - interpreting natural language

This field of of study is in short concerned with task of how natural language can be processed in such a way that it's semantics can be understood or interpreted by a computer program, to then act based on these interpretations. This is an essential part of any chatbot since it is important to try to understand what a user wants to say in order to produce a suiting answer. Although, this is far from a trivial problem, on the contrary it is very complex since natural language often is very abstract.

As a branch to NLP there is a field of study called Natural Language Understanding (NLU). While NLP is concerned with processing natural language to be interpreted, NLU focuses on the actual interpretation. Bolinda G. Chowdhury mentions three main problems within NLU: The first concerns the human thought process, the second the semantics of a given input and the third knowledge outside the program, or common knowledge. Moreover she suggests a number of steps to go through in order to get a good interpretation of natural language, the steps suggested are as follows[7]:

- phonetic or phonological level that deals with pronunciation

- morphological level that deals with the smallest parts of words, that carry a meaning, and suffixes and prefixes
- lexical level that deals with lexical meaning of words and parts of speech analyses
- syntactic level that deals with grammar and structure of sentences
- semantic level that deals with the meaning of words and sentences
- discourse level that deals with the structure of different kinds of text using document structures
- pragmatic level that deals with the knowledge that comes from the outside world, i.e., from outside the contents of the document.

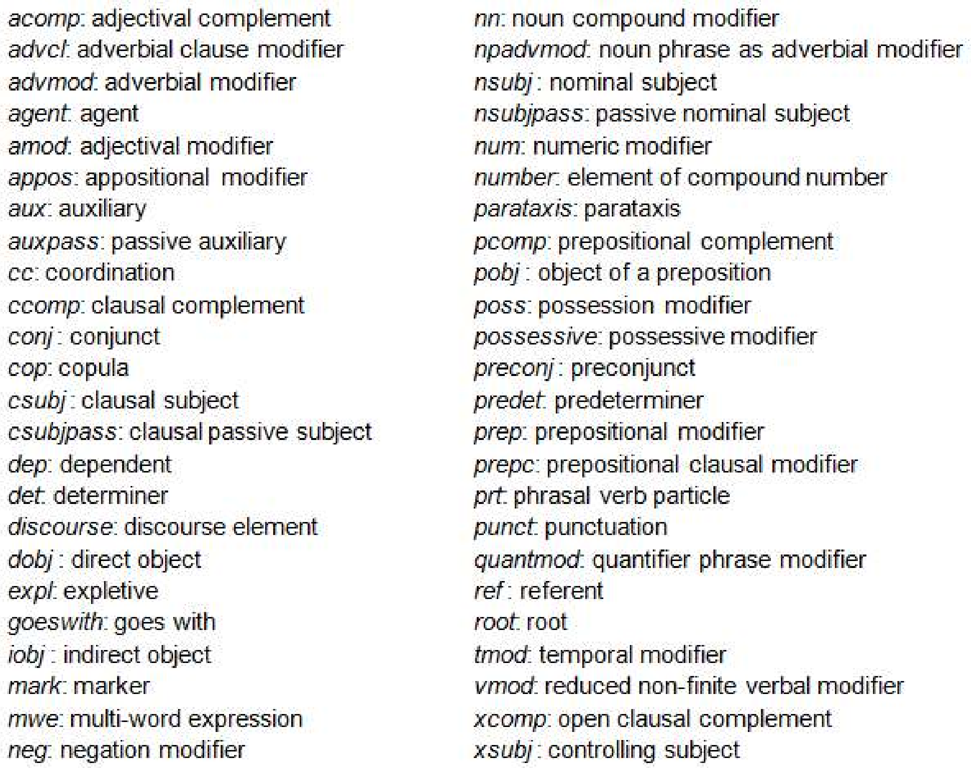
The first and sixth step did not apply to this project since they concern spoken language and longer texts respectively. The main idea however was more to investigate how well the seventh step, the pragmatic level, could be tackled using a common-sense database.

In order to apply the ideas of NLU, the natural language input must first processed or parsed, a tool for this is the so called Stanford Parser, which was used in the implementation of the chatbot and is described in the next section.

2.3 The Stanford Parser - parsing natural language

The Stanford Parser was mainly created by Dan Klein as a project at the Stanford university, and it is “a parser of natural language that can find the grammatical structure of a sentence”. Moreover it can produce parse trees for natural language sentences and identify different parts of a sentence such as subject and object of a verb[8]. This last feature to identify certain parts of a given user input was used in the implementation of the chatbot to identify keywords on which to base the answer.

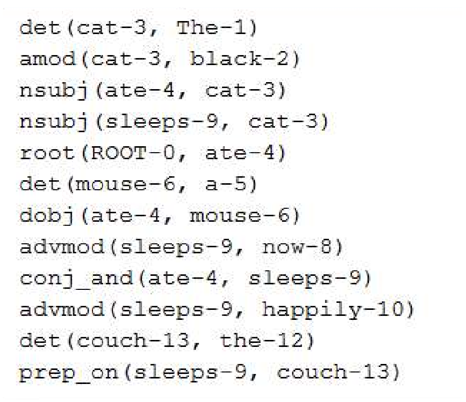
The parser identifies these parts as binary dependencies between two words in any given sentence, a list of the dependencies that are recognized by the parser is presented below in Figure 1[9].



acomp:	adjectival complement
advcl:	adverbial clause modifier
advmod:	adverbial modifier
agent:	agent
amod:	adjectival modifier
appos:	appositional modifier
aux:	auxiliary
auxpass:	passive auxiliary
cc:	coordination
ccomp:	clausal complement
conj :	conjunct
cop:	copula
csubj :	clausal subject
csubjpass:	clausal passive subject
dep:	dependent
det:	determiner
discourse:	discourse element
dobj :	direct object
expl:	expletive
goeswith:	goes with
iobj :	indirect object
mark:	marker
mwe:	multi-word expression
neg:	negation modifier
nn:	noun compound modifier
npadvmod:	noun phrase as adverbial modifier
nsubj :	nominal subject
nsubjpass:	passive nominal subject
num:	numeric modifier
number:	element of compound number
parataxis:	parataxis
pcomp:	prepositional complement
pobj :	object of a preposition
poss:	possession modifier
possessive:	possessive modifier
preconj :	preconjunct
predet:	predeterminer
prep:	prepositional modifier
prepc:	prepositional clausal modifier
prt:	phrasal verb particle
punct:	punctuation
quantmod:	quantifier phrase modifier
ref :	referent
root:	root
tmod:	temporal modifier
vmod:	reduced non-finite verbal modifier
xcomp:	open clausal complement
xsubj :	controlling subject

Figure 1; list over the dependencies that are used to identify different parts of a sentence, such as subject and object of a verb[9]*.*

In this project's simple implementation of a chatbot not many of these dependencies were actually used, keywords were mostly identified as nominal subjects, *nsubj*, or direct objects, *dobj*, of a verb. In Figure 2 below it is shown how the parser identifies the parts and dependencies of the sentence: "The black cat ate a mouse and now sleeps happily on the couch."



```
det(cat-3, The-1)
amod(cat-3, black-2)
nsubj(ate-4, cat-3)
nsubj(sleeps-9, cat-3)
root(ROOT-0, ate-4)
det(mouse-6, a-5)
dobj(ate-4, mouse-6)
advmod(sleeps-9, now-8)
conj_and(ate-4, sleeps-9)
advmod(sleeps-9, happily-10)
det(couch-13, the-12)
prep_on(sleeps-9, couch-13)
```

Figure 2; output from The Stanford Parser of the sentence "The black cat ate a mouse and now sleeps happily on the couch", as the parts of the sentence recognized as dependencies.

From the example in Figure 2 it can be seen that the *cat* is the subject of the verbs *ate* and *sleeps*, i.e. the cat ate and sleeps. The *mouse* on the other hand is the object of the verb *ate*, i.e. the mouse was eaten.

This makes a good base on how to process and interpret the user input, the next step was to consult the common-sense database, more on ConceptNet follows in the next section.

2.4 Common-sense database - ConceptNet

ConceptNet started out as project called Open Mind Common Sense at the MIT university to create a database over common knowledge, something that computers generally are not very good at. The data was originally gathered through crowd sourcing where people could log on to the project's web page and type in sentences of common knowledge to the database, for example: "Apple is a kind of fruit."[10] This project later evolved into ConceptNet where the natural language data was restructured to binary relations between so called concepts, for example "Apple is a kind of fruit" became: IsA(apple, fruit), where IsA is the relation between the concepts "apple" and "fruit", and "An apple is green" became: HasProperty(apple, green) where HasProperty is the relation between the concepts apple and green[11]. Natural language data was also gathered from internet resources such as Wikipedia and WordNet and structured into these kinds of relations.

ConceptNet can be seen as a giant graph, where the nodes consist of the so called concepts and the edges of the relations between concepts, a small part of the ConceptNet graph is shown as an example in Figure 3 below[12].

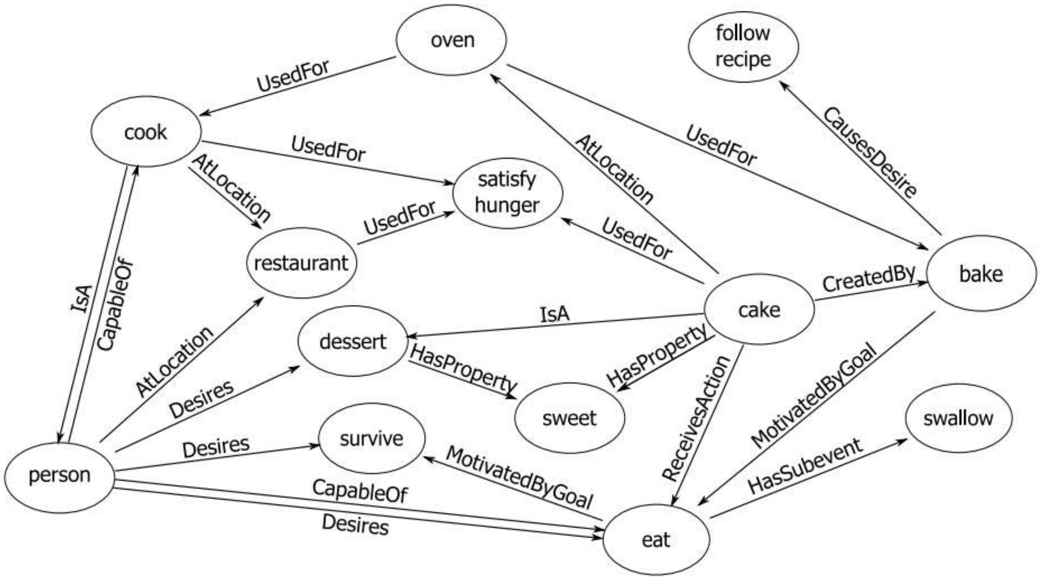
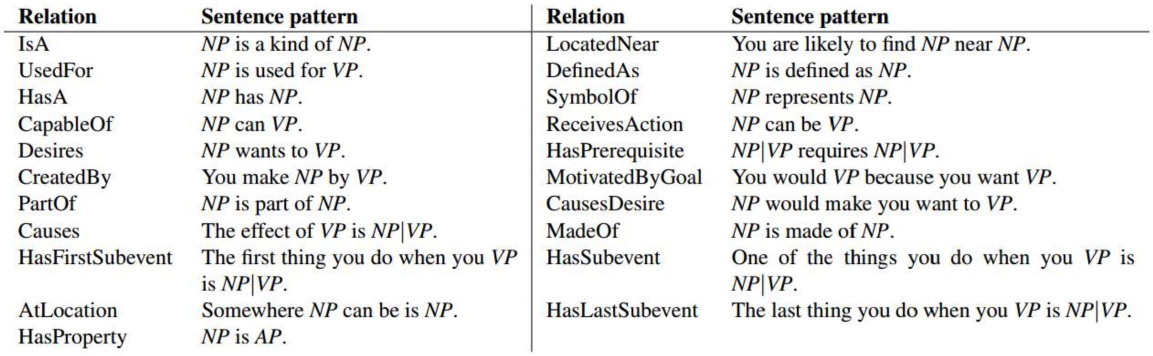


Figure 3, a small cut of ConceptNet seen as a graph, the nodes are the concepts and the edges the relations between different concepts[12].

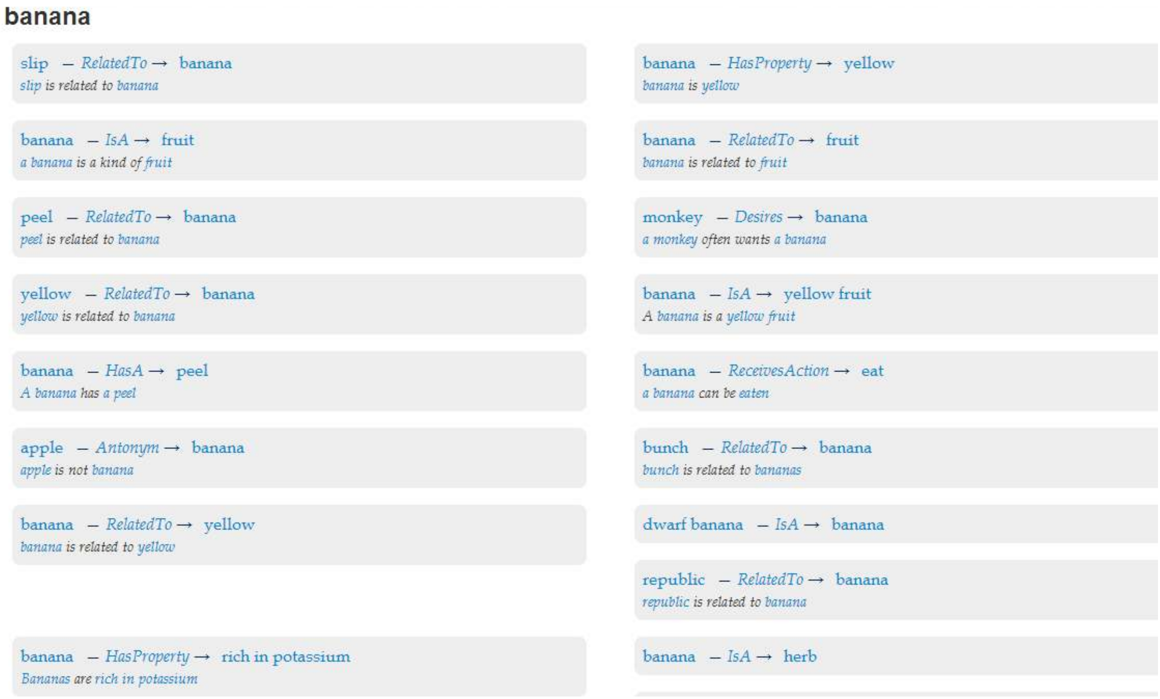
As can be seen in Figure 3, there is a number of different relations that may connect concepts. A selection of the most common relations is shown as a table in Figure 4 below.



Relation	Sentence pattern	Relation	Sentence pattern
IsA	*NP* is a kind of *NP*.	LocatedNear	You are likely to find *NP* near *NP*.
UsedFor	*NP* is used for *VP*.	DefinedAs	*NP* is defined as *NP*.
HasA	*NP* has *NP*.	SymbolOf	*NP* represents *NP*.
CapableOf	*NP* can *VP*.	ReceivesAction	*NP* can be *VP*.
Desires	*NP* wants to *VP*.	HasPrerequisite	*NP*|*VP* requires *NP*|*VP*.
CreatedBy	You make *NP* by *VP*.	MotivatedByGoal	You would *VP* because you want *VP*.
PartOf	*NP* is part of *NP*.	CausesDesire	*NP* would make you want to *VP*.
Causes	The effect of *VP* is *NP*|*VP*.	MadeOf	*NP* is made of *NP*.
HasFirstSubevent	The first thing you do when you *VP* is *NP*|*VP*.	HasSubevent	One of the things you do when you *VP* is *NP*|*VP*.
AtLocation	Somewhere *NP* can be is *NP*.	HasLastSubevent	The last thing you do when you *VP* is *NP*|*VP*.
HasProperty	*NP* is *AP*.		

Figure 4; table over a selection of the most common relations in ConceptNet, the sentence pattern indicates how a typical natural language sentence of the relations looks. NP stands for Noun Phrase, VP for Verb Phrase and AP for Adjective Phrase[12].

This is how ConceptNet is structured and built in theory, it also provides a web API that was used here, when implementing the chatbot, to put ConceptNet into use in practice. There is also a web interface on the web page to search for concepts in a web browser. Figure 5 below shows the result when searching for “banana” using the web interface, the data returned from the web API would be the same[13].



banana

slip – *RelatedTo* → banana
slip is related to banana

banana – *IsA* → fruit
a banana is a kind of fruit

peel – *RelatedTo* → banana
peel is related to banana

yellow – *RelatedTo* → banana
yellow is related to banana

banana – *HasA* → peel
A banana has a peel

apple – *Antonym* → banana
apple is not banana

banana – *RelatedTo* → yellow
banana is related to yellow

banana – *HasProperty* → rich in potassium
Bananas are rich in potassium

banana – *HasProperty* → yellow
banana is yellow

banana – *RelatedTo* → fruit
banana is related to fruit

monkey – *Desires* → banana
a monkey often wants a banana

banana – *IsA* → yellow fruit
A banana is a yellow fruit

banana – *ReceivesAction* → eat
a banana can be eaten

bunch – *RelatedTo* → banana
bunch is related to bananas

dwarf banana – *IsA* → banana

republic – *RelatedTo* → banana
republic is related to banana

banana – *IsA* → herb

Figure 5; the result when searching for “banana” using ConceptNet’s web interface.

This was a short history of ConceptNet and about how it is structured, in the next section it is described more closely how this project's chatbot was implemented and how the user tests then were chosen and designed.

3. Methods

In this section follows a description of how the chatbot was implemented. Since the main problem was to determine whether a common-sense database could improve the behaviour of a chatbot, the idea was to keep the chatbot simple and focus was put on using ConceptNet as much as possible.

3.1 Implementing the chatbot - a simple approach

To keep it simple, all user input was divided into one of three categories, namely:

- General questions
- Yes or No questions
- Statements

General questions start with (or contain) one of the words: "when", "what", "where", "why", "which" or "how" (called wh-words from here on), and finish with a question mark. Yes or no questions are all other sentences that finish with a question mark. Everything else was categorized as a statement.

Furthermore an approach not unlike the one that ELIZA uses was adopted where a keyword was recognized in the user's input, this keyword was then used to search ConceptNet and the response given by the chatbot was built around the same word. The keyword was either an object or, if no object was identified, a subject in the sentence given as input. These were identified using the dependencies of the Stanford parser.

If the input was categorized as a general question, ConceptNet was searched for the identified keyword and the information found in a relation in the search result were formatted into an answer, the relation was chosen randomly among the top search results to avoid too much of a repeating behaviour. If no keyword or search result were found an answer was given as "I don't know".

For a yes or no question; yes or no, randomly, were put first in the response to try to mimic a convincing behaviour. If searching ConceptNet for the keyword gave some hits, the information in any of the random top hits was put in the response to try to continue the conversation on the same topic.
For any input categorized as a statement, the response could either just ask the user to develop and talk more on the same topic. If searching ConceptNet for the keyword gave any results, the response could be given as a question as to whether the user was aware of this piece of information.

In addition to this a small database with hard coded answers, inspired by Alice, was added to handle input of hellos and goodbyes.

3.2 User tests - setup

How to measure performance of a chatbot with respect to naturalness and human-like behaviour is a difficult task since the opinion of these attributes is highly subjective[14]. To get a qualitative measure of a chatbot's performance user surveys are typically used.

In order to get a measure of whether a common-sense database in fact could improve on a chatbot, the chatbot was setup to face the Turing test along ELIZA and Elbot. In addition to the Turing test a small questionnaire was designed, the questions were taken from the suggested metrics for questionnaires on chatbot evaluation by Victor Hung et al[14]. They were as follows:

- Ease of usage
- Clarity
- Naturalness
- Friendliness
- Robustness regarding misunderstandings
- Willingness to use system again

The Turing test and each of these attributes were then graded on a scale from 0 to 100 by each user for all three chatbots. There was a total of ten users who participated in the tests.

The test environment was a standard laptop and the users conducted the test one by one, without having seen any use of- or tried to use any of the chatbots before. ELIZA was first tried for five minutes, then the chatbot implemented here for five minutes and finally Elbot for five minutes. The tests were not blind, the test subjects knew at all times which chatbot they were using. After trying each chatbot for five minutes, the test subject graded each category in the questionnaire for all three chatbots. The grading was made in a shorter discussion where all categories were discussed and the test subject could ask questions, air opinions and finally give a grade between 0 and 100 for each category.

The demographic profile of the user subjects was mainly computer science students in the age group 20 - 30 years. 30% of the users was of another age group, both older and younger, and background.

The results of these tests for each chatbot could then be compared to reach a final result for the project, as presented in the following section.

4. Results

The results in numbers from the user tests performed, as described in the previous section “User tests - setup”, for this project’s chatbot, ELIZA and Elbot are presented as a table in Table 1 below. The numbers in the table are the average results for each category that was evaluated through the questionnaire. The minimum possible value for each category was 0, and the maximum value was 100.

Table 1; average result from the user tests for each each category of the three evaluated chatbots, the minimum value for each category is 0, and the maximum value 100.

Category \ Chatbot	**This project**	**ELIZA**	**Elbot**
Turing test (human-like)	21	17	28
Ease of usage	64	46	65
Clarity	25	28	22
Naturalness	24	26	38
Friendliness	45	49	42
Robustness regarding misunderstandings	22	21	32
Willingness to use system again	35	12	27
Total average	**34**	**28**	**36**

No single chatbot among the three was a clear standout and superior to the others in all categories. Even the very simple implementation of ELIZA got the highest score in the categories clarity and friendliness. In the category inspired by the Turing test (human-likeness) this project’s chatbot could not really match the score of Elbot, but it could improve on the score of ELIZA.

In general, however, Elbot scored the highest, quite closely followed by the chatbot implemented here, as can be seen in total averages. ELIZA generally got the lowest score when looking at the total averages.

The corresponding unbiased sample variance, s^2, for each chatbot and category is presented in Table 2 below, calculated as:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

Where y_i is sample i from some category, $\bar{y}$ is the sample mean of the same category (Table 1) and n is the total sample size, i.e. 10 in this case.

Table 2; the unbiased sample variance of the user tests for each category and chatbot.

Category \ Chatbot	**This project**	**ELIZA**	**Elbot**
Turing test (human-like)	64.0	61.2	130.3
Ease of usage	336.4	339.8	323.1
Clarity	330.7	312.8	296.7
Naturalness	157.3	262.7	402.4
Friendliness	471.7	520.6	504.8
Robustness regarding misunderstandings	134.7	118.2	258.7
Willingness to use system again	368.9	65.4	264.7
Total average	**266.2**	**240.1**	**311.5**

The grading from different users could vary a lot within a category as suggested by the rather high values of variance.

Further interpretation and discussion of these results follows in the next section.

5. Discussion

As suggested by the high values of variance, the rather small number test subjects is problematic and makes it difficult to conclude anything at all from the results. Finding test subjects and conducting the tests turned out to be a rather time consuming task, which made it difficult to find a bigger number of testers, regarding the limited time frame for the project. Another obvious drawback was that the user tests were not blind, that the test subject knew at all times which chatbot they were currently testing, this opens up for possible bias towards any of the chatbots tested.

However, the tendency shown in the results is promising in that a simple implementation of a chatbot, with the addition of a common-sense database, could improve on, or at least match ELIZA in most of the categories that were evaluated. In comparison to Elbot it did not stray too far behind in the average score. It was interesting that the chatbot implemented here scored highest in the category “willingness to use system again”, this could indicate that the addition of a common-sense database makes a chatbot more interesting or funnier to chat with.

Another drawback is that the different categories used for the questionnaire might vary in relevance considering the original problem statement and human-likeness, this is not considered at all and all categories have the same weight on the final result.

5.1 Future research

A suggestion for future research on the same topic would be to look more on which categories that are most relevant to evaluate regarding human-like behaviour. Furthermore it would be a good next step to make the tests blind, i.e. not letting the test subjects know which chatbot they are talking to.

It would also be interesting to look more on how the information in the common-sense database could be used in a more efficient way to mimic human behaviour.

6. Conclusions

As stated in the discussion in the previous section it is hard to draw any conclusions at all considering that the amount of user tests conducted was small.
However, the tendencies of the results are promising in that a common-sense database would be a good addition to a chatbot as to make it more human-like. Although, further research would be required to make any real conclusions.

References

[1] Joseph Weizenbaum (1966). “ELIZA - A Computer Program For the Study of Natural Language Communication Between Man and Machine”
Communications of the ACM, Volume 9, Number 1. pp. 36-35.

[2] A. M. Turing (1950). “Computing Machinery and Intelligence”

[3] Web page. “ConceptNet 5 - About”, <http://conceptnet5.media.mit.edu/>

[4] Web page. “Who is Elbot?”, <http://www.elbot.com/chatterbot-elbot/>

[5] Richard S. Wallace (2009). “The Anatomy of A.L.I.C.E.”
Parsing the Turing test. pp. 181-210.

[6] Artificial Solutions. “What is NLI?”
<http://www.artificial-solutions.com/wp-content/uploads/protected/td-what-is-nli.pdf>

[7] Gobinda G. Chowdhury (2003). “Natural Language Processing”
Annual Review of Information Science and Technology, *37*. pp. 51-89.

[8] Web page. “The Stanford NLP Group” <http://nlp.stanford.edu/software/lex-parser.shtml>

[9] Marie-Catherine de Marneffe, Christopher D. Manning (2008). “Stanford typed dependencies manual”
<http://nlp.stanford.edu/software/dependencies_manual.pdf>

[10] Push Singh, Grace Lim, Thomas Lin, Erik T. Mueller, Travell Perkins, Mark Tompkins, Wan Li Zhu. “Open Mind Common Sense: Knowledge Acquisition from the General Public”
<http://web.media.mit.edu/~push/OMCS-2.pdf>

[11] Hugo Liu, Push Singh (2004). “ConceptNet — a practical commonsense reasoning tool-kit” *BT Technology Journal, Volume, 22, Number 4*.

[12] Robert Speer, Catherine Havasi. “Representing General Relational Knowledge in ConceptNet 5”
<http://www.lrec-conf.org/proceedings/lrec2012/pdf/1072_Paper.pdf>

[13] Web page. “ConceptNet 5” <http://conceptnet5.media.mit.edu/>

[14] Victor Hung, Miguel Elvir, Avelino Gonzalez, Ronald DeMara (2009). “Towards a Method For Evaluating Naturalness in Conversational Dialog Systems”
IEEE International Conference on Systems, Man and Cybernetics. pp. 1236-1241.

www.kth.se