

Bachelors Essay in Computer Science
Royal Institute of Technology

Automatic Twitter Response Generator

Christoffer Gunning
cgunning@kth.se

Daniel Forslund
dforsl@kth.se

Abstract

Since the beginning of chatbots, the most common techniques used for simulating conversation has been different pattern matching-techniques. This is a study of possible alternative methods and an attempt of implementation. As the research of natural language processing is rapidly progressing, the aim of this report is to clarify current techniques and what could possibly be achieved by adapting recent findings on the area.

A combination of part-of-speech tagging, methods for information retrieval and known methods for generating language are covered and described of how it could be of use. An implementation is reviewed and compared to traditional programs. The final conclusion states that the implementation is not as satisfying as chatbots using elaborated pattern matching-techniques. However, it is concluded that the new methods covered are more likely to be part of future chatbots considering the progress of natural language processing.

Contents

1	Introduction	1
	1.1 Purpose.....	1
	1.2 Objective.....	2
	1.3 Document Overview.....	2
2	Background	3
	2.1 Existing Chatbots.....	3
	2.2 Pattern Matching.....	3
	2.3 Learning Agents.....	5
3	Problems	6
	3.1 Understanding Natural Language.....	6
	3.2 Information Retrieval and Analysis.....	7
	3.3 Generation of Natural Language.....	7
	3.4 Knowledge Base.....	8
4	Methods	9
	4.1 Research.....	9
	4.2 Development.....	9
	4.3 Evaluation.....	10
5	Results	11
	5.1 Possible Implementation.....	11
	5.2 Conversational Differences from Other Chatbots.....	11
6	Discussion	12
	6.1 Response Content and Language.....	12
	6.2 Source of Knowledge.....	12
	6.3 Creation of Natural Language.....	12
	6.4 Learning Possibilities.....	13
	6.5 Market Perspective.....	13
	6.6 Conclusions.....	13
7	Bibliography	15
8	Appendix A	16

Chapter 1

Introduction

It has been over four decades since we encountered one of the first chatbots, ELIZA. ELIZA is an example of a computer program able to process natural language in a very primitive way. By using simple pattern matching techniques, the program could generate a response in context to the user's input in natural language.

In the everyday life many of us interact with these kinds of dialog systems. An educational scenario is one possibility, where this type of program is used for direct interaction. Chatbots for the matter of support is another one increasing in popularity among many companies.

Since ELIZA, many other chatbots have evolved and are using different techniques for processing natural language. One of the most widely methods used is AIML, Artificial Intelligence Markup Language. It takes use of XML as a knowledge base, consisting of defined categories of patterns and response templates. An infinite number of categories can be added to the knowledge base, and many already exist as open source. However, as a result most programs are not able to recall previous conversations, or even parts of what has been said, as the generated response is mainly based on the user's latest statement or question.

As the title of this report indicates, creating a program for automatic twitter response generation is the main objective of this project. Twitter is a widely used service for microblogging being used by millions of people every day. An implementation of a chatbot, more commonly regarded to as dialog systems, is the method of choice for achieving this. While pattern matching as described might seem as a straightforward way of implementing this, the use of alternative and more sophisticated techniques are processed in this report to achieve something different.

1.1 Purpose

Intelligent conversational agents is a topic within artificial intelligence under heavy research. The area of use is extensive and they are becoming more common everyday. The early chatbots used pattern matching for generating their responses, which has been refined over the years. The technique is simple, predictive and the behaviour completely predefined.

The purpose of this report is to research different techniques, evaluate whether it exists more sophisticated and dynamic methods than simple pattern matching, and develop a conversational agent from this.

1.2 Objective

The objective was to create a sophisticated, self-learning dialog system. A system with the ability to interpret natural language, decide on important keys in sentences, extract appropriate data out a knowledge base to form a response in natural language. The mentioned properties and taking use of Twitter for source of knowledge as well place for interaction is all part of the project.

We hypothesized that the processing of natural language and a dynamic knowledge base, for learning and data extraction, will result in a dialog system with the ability to take in user input, learn from it and form a response of relevance.

1.3 Document Overview

Chapter X one finds information regarding existing chatbots and techniques. The main focus is pattern matching, the currently most common way of implementing chatbots, and how it is implemented.

In Chapter Y, problems in theories regarding alternative techniques focusing on natural language processing is presented. Possible solutions and implementations are explained described lightly.

Methods for the study are described in Chapter Z. Results can be found in Chapter Å and discussion as well as conclusions are presented in Chapter Ä.

Chapter 2

Background

Different techniques used in chatbots has been mentioned already. With the purpose of gaining a deeper understanding of how these are used, what follows are closer evaluations and how they are commonly implemented.

2.1 Existing Chatbots

The first chatbots, such as ELIZA, PARRY and RACTER, used rule based pattern matching-techniques[1], a method still commonly used today by many chatbots to simulate conversation. One of the main reasons for this is it being an easy method to create chatbots with human-like responses. There are now many different kinds of chatbots using different techniques, but the majority still make use of pattern matching to generate their response.[2]

2.2 Pattern Matching

Chatbots based on pattern matching-techniques contain rules for how to respond to different input. By using pattern matching it is much easier to make the answers appear more human as the author has full control over the responses. However, to be able to provide an answer for a wide variety of input many rules are required. One of the most famous chatbots (A.L.I.C.E) has 120,000 rules.[3] Due to that two sentences meaning of the same meaning can look very differently, and in order for the chatbot to be able to handle all possible variations, it would need one rule for every possibility, though some differences can often be handled by the use of wildcards.

The weakness of these kinds of techniques is the number of rules needed to be able to generate answers for a wide variety of input and the chatbot is limited to handling only the questions specified in the rules.

AIML

AIML (Artificial Intelligence Markup Language) is a commonly used pattern matching-technique developed by Richard Wallace.[4] It is mainly famous for being used by the chatbot A.L.I.C.E., winner of the Loebner prize 2000, 2001 and 2004.

AIML is a XML-compliant language with support for rules based on input from user and, optionally, the chatbots latest statement. The four most important tags in AIML is <aiml>, <category>, <pattern> and <template>.

Tag	Description
<aiml>	Marks the beginning and end of an AIML document.
<category>	The tag marking a rule in the knowledge base.
<pattern>	Contains the pattern to match user input against.
<template>	Contains the answer to user input matching the corresponding <pattern>-tag.

There are also other tags, such as <that>, which for instance matches against the last thing the chatbot said.[5]

An example rule is the following, specifying what the chatbot will answer if the input is “Do you love me?”.

```
<category>
  <pattern> DO YOU LOVE ME </pattern>
  <template>I love you </template>
</category>
```

The next rule specifies the answer if the input is “You are just saying that” and the last thing the chatbot said was “I love you”.

```
<category>
  <pattern> YOU ARE JUST SAYING THAT </pattern>
  <that> I LOVE YOU </that>
  <template> No, I mean it </template>
</category>
```

These two rules combined enables the following conversation:

u - The user

b - The chatbot

u: Do you love me?

b: I love you

u: You are just saying that

b: No, I really mean it.

If any word in this conversation would differ additional rules would be needed to cover those possibilities too. The possibility of specifying rules taking more into account than just the last user input goes a long way when trying to imitate real understanding.

2.3 Learning Agents

Chatbots learning from user input are more versatile as it will add data to its knowledge base as the time progress, thus resulting in a behavior less predictable than when using pattern matching-techniques. Consequently, these kind of techniques put a much higher demand on complex and effective algorithms for generating suitable answers.

The knowledge database grows by continually storing new information retrieved from its conversations with different users. This way the knowledge base grows more and more extensive with every conversation and the chatbot more and more versatile.

CleverBot

CleverBot is a typical learning agent. It learns by storing input from users in it's knowledge base. When a user is chatting with CleverBot, it searches its database of previously recorded input for a sentence that could serve as a suitable answer.[6]

Chapter 3

Problems

The chatbot covered in this paper is an attempt to create a chatbot that is more versatile than the ones using pattern matching and explore how this can be achieved. This involves many different problems which needs to be solved.

3.1 Understanding Natural Language

To be able to respond to input, the program must first be able to understand it. Instead of pattern matching, a rather robust way of doing this is by finding the different nouns in a sentence and assume that these are the subjects being spoken of. One way of doing this is by taking use of a technique called part-of-speech tagging, also called grammatical tagging. Part-of-speech tagging is a process of marking a word as corresponding to a particular part of speech. The marking is based on both its definition as well as its context, such as the relationship with adjacent and related words in a sentence.

The task of POS tagging is rather complicated due to ambiguity.[7] Consider the sentences below.

- “*Plants[noun] need light and water*”
- “*I will plant[verb] some flowers*”

Humans immediately understand the difference between the word plant in each sentence thanks to the human brain. However, making a program understand this is much more complex.

There are different methods for POS tagging, such as rule-based and stochastic tagging. Simply, a rule-based tagger is made of a large collection of constraints on how sentences can be tagged. The more refined stochastic method of tagging is based on probability of a certain tag to occur, given various possibilities. For this, a training corpus is used holding a collection of sentences already tagged from which the program can learn.

Different taggers can use different tags of different meaning. By using a functioning POS tagger implemented with the Penn Treebank Tags, the process would generate the following:

Sentence: *Many planets rotate around the sun.*

POS tagged: *JJ_Many NNS_planets VBP_rotate IN_around DT_the NN_sun*

From the tagging, it is possible to distinguish the type of each word and a program could possibly use it for extracting information of relevance. For instance, In the above example it is clear, with the use of a tag dictionary, that the words “planets” and “sun” are two nouns, and likely to be subjects of the sentence.

3.2 Information Retrieval and Analysis

In order to respond to input, assuming the program has an empty knowledge base, the ability to retrieve relevant information is required. When performing a search, it is essential that the keywords match what is being looked for and that all possible subsets of information are retrieved, and possibly ranked.

For a given sentence, a set, X , of keywords are found. A simple way of finding all possible information of relevance would be to commit one search for each keyword in X . Though, a search engine such as Google is ranking results differently depending on the order of the search keywords. Due to this, for retrieving all possible information concerning the keywords, there should be a search for each permutation of the relevant keywords. It means

$$\alpha = \sum_0^n n!$$

number of searches for being able to retrieve all of the possible information. Even with a low number of keywords, α will grow quickly and the process could possibly be very time and resource consuming.

The result sets of data can be of varying relevance. To determine data’s relevance an analysis of the keywords’ importance could be made if the results were ranked the same. One possible method for doing this is by ranking each set of data from the number of different keywords it contains. Highly ranked data is then likely to be the same as the highest matches. The result from ranking can then be used when generating natural language in response to input.

3.3 Generation of Natural Language

The simplest way of “generating” responses is simply to steal existing sentences matching the input. It would result in (depending on the quality of the text stolen) actual natural language without further computations or the need for complex algorithms.

If allowing the program to form sentences on its own, an optional strategy would be n-grams, used for many purposes in connection to natural language. It is a probabilistic model and can be used for predicting the following item in a sequence, such as words in sentences. An n-gram typically consists of n items and a set of possible items to follow, each with a corresponding

probabilistic value of its occurrence. By creating n-grams from the data found during information retrieval and analysis, and possibly ranking the data accordingly as previously explained, natural language could be created in different forms.[8]

By adjusting size of the n-gram model varying result can be achieved. Using a bigram model, n-gram of size 2, there would basically be a bigram for each word found in the data retrieved and all possible words to follow along with its probability to occur. Bigrams are however likely make a sentence fall out of context, as it bases following words on a too small sequence. The words will in short be selected with no regard to the context. Due to this, a higher n-gram will result in better coherence. However, the use of higher n-grams will require more time and processing. Also, with the use of too high n-grams the probability theory behind will not perform very well due to insufficient amount of data, as more data will be needed for finding unique sequences to be added to the n-grams. The number of different words found after the same previous 2 words are many more than those after the previous 7 or 8. It is a trade-off which needs to be considered if the strategy is chosen.[9]

3.4 Knowledge Base

For creating a knowledge base with the possibility of being extended, representation and actual storage are two areas to consider. If taking use of a n-gram model, storing them in a corresponding data structure is an option. Keeping them in memory is not advisable as large amounts of data is likely to be processed and the total memory needed is probably not available. Creating a SQL database is an option, though a perhaps simpler way would be direct storage in files. This can be achieved by using a trie implementation pointing to the possible items and their probability.[10]

Chapter 4

Methods

This study has been divided into three phases concerning different areas of the study; research, development and evaluation.

4.1 Research

Initially, existing methods and techniques were researched and evaluated with the purpose of gathering deeper information concerning chatbots, techniques and how the progress has been over time.

Chatbots of interest were mainly ones with good results in Turing tests and Loebner prize winners. Our objective has been to find ways of achieving less predictable and more sophisticated and dynamic behaviour than many of those you can find today. For doing this, the main research area have been natural language processing, with the aspects of how to interpret, analyse and generate natural language.

Existing libraries for above purposes have also been researched and tested during this phase. As many of the tasks involved the use of advanced algorithms techniques, were the implementations have not been the focus of this study, this has been of high priority for an implementation of the research to be possible.

The results are presented in this report in the forms of background, problems and result. Analysis of what has been found and as well as conclusions can be found in the final discussion.

4.2 Development

The development phase has persisted of much trial and error as the object was to explore ways to create a chatbot that was more versatile than the ones already existing. It consisted of three different phases itself; interpretation, information retrieval/storage and language generation.

Interpreting

POS tagging was the main area and a few different libraries for this were tested. The phase involved the actual tagging of text and analysis of the result.

Information Retrieval and Storage

Different techniques were tried based on previous research on how searching procedures for information could be made. It also involved some analysis of the results and how it was to be ranked.

Language Generation

Has concerned different ways of how language could be generated. This stage was partly done during while finding ways of analysing search results, as the data structure of information was connected to presentation.

4.3 Evaluation

For evaluating the resulting program's behaviour, comparisons mainly on interaction has been done to existing chatbots where they have been presented with the same input.

Chapter 5

Result

This section will account for the results of this study, concerning possible implementation with the use of previous research and the differences from other chatbots as well as the human experience from conversations.

5.1 Possible Implementation

The final implementation was a chatbot using POS tagging and a trigram (n-gram of size 3) as primary techniques. POS tagging was used to find keywords in the input, later used as parameters to search for relevant tweets. It uses an algorithm specifically created for the case to find the main subject/subjects of the input.

When keywords are extracted, it proceeds by searching on Twitter for matching tweets. Different kinds of keyword phrases are used, however due to time limitations not all as advised in section 3.2 for the largest set of data.

A trigram is then created from the search results to be used for language generation. To generate a sentence using a trigram a minimum of two words are needed to start the process. These words are chosen as the most common word preceding or following the subject noun in the search results and the subject noun.

5.2 Conversational Differences From Other Chatbots

As pattern matching-techniques are most commonly used in chatbot implementations, these are the kinds of chatbots being compared.

The major difference is that chatbots using pattern matching only can answer sentences covered in their rules. If a matching rule can't be found, the answer will often be something similar to "I don't understand" or "How interesting". On the other hand the developed program will always answer with something, though sometimes it may be with incomplete sentences or irrelevant content. Examples responses can be found in appendix A.

Chapter 6

Discussion

6.1 Response Content and Language

The main problem with using the approach used for implementation is that there is a chance that the generated sentence mostly or completely consists of one or many already existing tweet. The sentence could then become very hard to understand as it does not make sense at all. To prevent this, more data than the amount currently being processed for each response is needed. The current implementation is limited to 100 tweets per subject found. This could be increased a lot, which would hopefully result in improved responses without modifying the actual natural language processing.

6.2 Source of Knowledge

Since tweets are used as the only source of data, it will not consist exclusively of facts and sentences spelled correctly and grammatically valid. This opens up the risk that the answer generated could be based on other persons opinions or experiences, which is to think of before use.

The amount of information retrieved is an area for discussion. The chosen approach requires a lot information to be retrieved and processed which both consumes time and memory. The memory issue could be easily avoided by modifications to the implementation, of more interest is the choice between more or less data. Less data, such as 100 tweets per sentence “subject”, is obviously more likely to result in sentences not making sense, compared to using 1000 for each. Also, if the program would have been searching for all possible permutations of the keywords, it would not only have retrieved more data but it would also in a way have served as a way of ranking common results, due to multiple occurrences.

Compared to pattern matching there are many more factors that needs to be considered which opens up for a lot more of uncertainty. As stated, the result is improved by processing more data but the implementations can definitely be improved.

6.3 Creation of Natural Language

Since trigrams builds a sentence by adding the word most likely to follow the previous two words to the sentence, there is no guarantee that the sentence will follow the same line of argument. However, since tweets made by actual humans are used as source for the trigrams, the created sentences will appear relatively human-like. It will, however, not be able to match the

output of a well implemented chatbot using pattern matching but it will be much more versatile and capable of creating answers for a wide variety of sentences.

The major flaw of using a trigram for creation of natural language is the lack of control and in some sense the predictability. The sentences generated will depend entirely on the data and subjects extracted from the input. There is no controlling entity that assures that the sentence generated is a correct sentence and of relevance. Because of this it is important to choose the data for the trigram and the first words wisely in order to increase the chance of creating a relevant response.

6.4 Learning Possibilities

The chatbot we created can easily be implemented with the ability for learning, simply made out of allowing it to store data which has already been processed as a knowledge base. A strategy for how this could be implemented is explained within section 3.4. The result of storing already processed data as knowledge would in the long run mean that the information processed for creating responses is a larger set than the current implementation. This means more data to form its probability models out of and more accurate responses.

6.5 Market Perspective

The chatbots of today are ruled by chatbots based on pattern matching-techniques. In comparison to dialog systems taking use of natural language processing, they are easy to develop with satisfying results regarding to the cost. Also for purposes such as support, they are also easily managed and controlled.

The major weakness is the fact of these chatbots being restricted to rules. Even though there are advancements in the area, this fact remains and for purposes requiring a more dynamic dialog system it might not be enough. Chatbots for educational purposes is an example of this, where the system has the purpose of teaching, simply being an entity which can answer questions in any form. Though our implementation is not near working as a “sophisticated” and dynamic program for similar purposes, it is a step in what we conclude is the right way to go. Natural language processing is an area within artificial intelligence under heavy research and rapid development. Even if a program were to add new rules to itself, it is probably more likely to be a program full of functionality for understanding natural language already for it even to be possible, why the traditional kind of rules would be redundant.

6.6 Conclusions

Despite the fact of our implementation forming responses far away from satisfying, we believe it to be in the direction of how future dialog systems will function. Programs made of pattern matching has been used for over 40 years and there are more sophisticated ways under research

in the area of natural language processing. The study has resulted in an implementation which does take user input and can form responses, though of varying relevance to the input. Reasons for varying result and suggestions of improvements has been made, as well as review of how learning functionality could be implemented.

Bibliography

- [1] Güven Güzeldere and Stefano Franchi, 1995, Dialogues with Colorful Personalities of Early AI, retrieved 19 February 2013. <<http://www.stanford.edu/group/SHR/4-2/text/dialogues.html>>
- [2] AI4US Ltd., [chatbots.org](http://www.chatbots.org), retrieved March, 2013. <<http://www.chatbots.org>>
- [3] Bruce Wilcox, 4 January 2012, Fresh Perspectives - A Google talk on Natural Language Processing. <http://gamasutra.com/blogs/BruceWilcox/20120104/9179/Fresh_Perspectives_A_Google_talk_on_Natural_Language_Processing.php>
- [4] A.L.I.C.E AI Foundation, Bio: Richard Wallace, retrieved on 19 February 2013. <<http://www.alicebot.org/bios/richardwallace.html>>
- [5] A.L.I.C.E AI Foundation, AIML - The Artificial Intelligence Markup Language, retrieved on 27 February 2013. <<http://www.alicebot.org/aiml.html>>
- [6] NewScientist - Jacob Aron, 6 September 2011, Software tricks people into thinking its human, retrieved on 28 February 2013. <<http://www.newscientist.com/article/dn20865-software-tricks-people-into-thinking-it-is-human.html>>
- [7] University of Maryland - Department of Computer Science, Part-of-Speech Tagging, retrieved on 16 March 2013. <<http://www.cs.umd.edu/~nau/cmssc421/part-of-speech-tagging.pdf>>
- [8] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, Jenifer C. Lai, Class-Based n-gram Models of Natural Language, retrieved on 26 March 2013. <<http://acl.ldc.upenn.edu/J/J92/J92-4003.pdf>>
- [9] Pedro Paulo Balage, 18 January 2011, Experiments with Statistical Language Generation, retrieved on 28 March 2013. <<http://nlpb.blogspot.se/2011/01/experiments-with-statistical-language.html>>
- [10] Adam Pauls, Dan Klein, 2011, Faster and Smaller N-Gram Language Models, retrieved on 2 April 2013. <http://nlp.cs.berkeley.edu/pubs/Pauls-Klein_2011_LM_paper.pdf>

Appendix A

This appendix contains tests of the developed bot AILA. Five tweets has been chosen and each tweet has been used as input to AILA as well as two other bots, CleverBot and ALICE. ALICE is based on pattern matching and CleverBot is a learning agent. The chatbots answers for each tweet is shown in the tables below.

Instead of doing my homework I like to stay on the computer and worry about how much homework I have to do.

ChatBot	Answer
ALICE	That is something I haven't heard of.
CleverBot	Yes, but you do have to worry about children pouring goo into your keyboard.
AILA	more classes of homework to do my homework and my computer has been through so much homework to do my homework

If you can prove to the Nando's staff that you've been to all of their restaurants, you can eat free for life there.

ChatBot	Answer
ALICE	The proof cannot fit in this margin.
CleverBot	If you can't intelligently answer what I'm saying, then you obviously make no sense.
AILA	i bet you cant wait and im excited for the staff in the old quarter of san roque don benito is organising a <link>

Late night thoughts before falling asleep are always the most intriguing

ChatBot	Answer
ALICE	All of them?
CleverBot	No I will remember it, and I do not date males. I am a male myself.
AILA	its hard to fall asleep at night when so many thoughts are running through my head how will i ever fall back asleep because of thoughts and prayers are with you both

Music always helps, no matter what you're going through.

ChatBot	Answer
ALICE	Can you think of one instance where it never happens?
CleverBot	I don't even know what to say.
AILA	modern rock legends the cure are booked to headline the lollapalooza for the first time alongside mumford and sons organ

Does anyone else wake up in the middle of the night thirsty as hell, and water tastes like it came from God's personal river bank?

ChatBot	Answer
ALICE	Suppose they do.
CleverBot	Your grammar seems good enough. Sometimes you just answer questions that aren't asked.
AILA	that even though ive become super rich i feel the exact same i hate waking up in the middle of the night for me bringing me treats helping me in her suitcase were supposed to feel this good waking up after a night out with my boys is like being pregnant