Machine learning for industrial extension: Industrial chatbots

Varshni Devi B, Mounika V
Msc Software Systems(Third year)
Coimbatore Institute of technology
varshnibabu@gmail.com, mounikavenkatesan1998@gmail.com

Abstract

Machine learning and artificial intelligence has transformed almost every industry. With the huge amount of data and applying machine learning models, we are able to get valuable insights which help in improved production rate and sales. Machine learning and robotics plays an important role and has created a huge impact in our workplace. Automation is being applied in industry to reduce human effort. Breakthroughs in machine learning help in creating natural language processing model. Chatbots is a computer program that communicates with the human via text or auditory. We try to solve industrial issues using chatbots which plays the role of an advisor. It helps us to take decisions regarding stock maintenance, machinery expansion, demand detection, cost cutting, production rate etc. Chatbots analyse these problems and helps us to make right decisions. Thus human biases are removed from the process. In our paper we analyse the data to optimize our decision in the industrial field.The analysis of data is done using machine learning and is used for each prediction. Humans are being replaced by chatbots as they work efficiently, avoid errors, faster response time, increase in accuracy, cost effective, higher volume production and most importantly it saves human time.

Keywords:
Machine learning, data analysis, chatbots, artificial intelligence, robotics, industry.

1. Introduction

Human machine conversation is a new technology integrating different areas where the core is the language, and the computational methodologies, which aim to facilitate communication between users and computers via natural language. A related term to machine conversation is the chatbot, which is a conversational agent that interacts with users turn by turn using natural language; chatbots have been applied in different domains such as: linguistic research, language education, customer service, website help, and for fun. The purpose of a chatbot system is to simulate a human conversation; the chatbot architecture integrates a language model and computational algorithms to emulate informal chat communication between a human user and a computer using natural language. According to a new survey, 80% of businesses want to integrate chatbots in their business model by 2020. Industry 4.0 is a name for the current trend of automation and data exchange in manufacturing technologies. It includes cyber-physical systems, the Internet of things, cloud computing and cognitive computing. It creates what has been called a “smart factory”. Automation encompasses not only simple, repetitive tasks, but also many sophisticated functionalities. However, even in systems in which automation has replaced functionality previously performed by humans, the human is still a central player. As automation becomes “smarter” and more ubiquitous, it is paramount that the human interact with the controlled systems in a safe and efficient way, to help prevent problems in human-automation interaction. The aim is not to replace human with machine, is to create a new type of interaction between capability of machines and flexibility of human. The manufacturing industry is well suited to take the advantage of chatbots. Manufacturers can create a chatbot to help the employees get accurate information regarding various processes and production levels quickly, stock maintenance, machinery expansion, demand deduction, cost cutting, increased production rate, eliminating the need for ERP and CRM systems. By using chatbots it makes it easier to engage with vendors, perform HR-related tasks, helping maintenance crew to verify environmental or workplace conditions, aids the manufacturers assessing the need to recall products.

2. Early chatbots

There were numerous chatbots and chatbot technologies already before the first Loebner competition, mostly in games and focused domain expert systems. It is not known how well they performed and they were never compared against each other. The very first known chatbot was Eliza, which was developed in 1966. Its goal was to behave as a Rogerian psychologist. It used simple pattern matching and mostly returned users sentences in a form of questions. Its conversational ability was not very good, but it was enough to confuse people at a time when they were not used to interact with computers and to start the development of other chatbot systems. The very first online implementation of Eliza was done by the researches at Jozef Stefan Institute in Ljubljana, Slovenia and is still available1 for testing. The first such a system that was actually evaluated using some sort of Turing Test was PARRY [1]. Parry was designed to talk as a paranoid person. Its transcripts were given to psychiatrists together with transcripts from real paranoia patients for comparison. The psychiatrists were able to make the correct identification only 48% of the time.

3. BACKGROUND

3.1. Human-Computer Speech interaction

Speech recognition is one of the most natural and sought after techniques in computer and networked device interaction has only recently become possible (last two decades) with the advent of fast computing. Speech is a sophisticated signal and happens at different levels: “semantic, linguistic, articulator and acoustic” [2]. Speech is considered as the most natural among the aspects of human communication, owing to copious information implicitly existing beyond the meaning of the spoken words. One of the speech information extraction stages is converting speech to text via Automatic Speech Recognition (ASR) and mining speech information [3]; then, the resulting text can be treated to extract the meaning of the words. Speech recognition is widely accepted as the future of interaction with computers and mobile applications; there is no need to use traditional input devices such as the mouse, keyboard or touch sensitive screen and is especially useful for users who do not have the ability to use these traditional devices [4]. It can help disabled people with paralysis, for example, to interact with modern devices easily by voice only without moving their hands.

3.2. Natural Language Toolkit (NLTK)

In order to deal with and manipulate the text resulting from speech recognition and speech to text conversion, specific toolkits are needed to organise the text into sentences then split them into words, to facilitate semantic and meaning extraction. One of these toolkits is the widely used NLTK which is a free plugin for Python. The Natural Language ToolKit (NLTK) is a set of modules, tutorials and exercises which are open source and cover Natural Language Processing symbolically and statistically. NLTK was developed at the University of Pennsylvania in 2001allowing computational linguistics with three educational applications in mind: projects, assignments and demonstrations [5] [6]. It can be found within the Python Libraries for Graph manipulation GPL open license. NLTK is used to split words in a string of text and separate the text into parts of speech by tagging word labels according to their positions and functions in the sentence. The resulting tagged words are then processed to extract the meaning and produce a response as speech or action as required. Different grammar rules are used to categorise the tagged words in the text into groups or phrases relating to their neighbours and positions. This type of grouping is called chunking into phrases, such as noun phrases and verb phrases. Chatbot strategies to give suitable answers to keywords or phrases extracted from speech and to keep conversation continuous, there is a need to build a dialogue system (programme) called a Chatbot (Chatter-Bot). Chatbots can assist in human computer interaction and they have the ability to examine and influence the behaviour of the user [7] by asking questions and responding to the user's questions. The Chatbot is a computer programme that mimics intelligent conversation. The input to this programme is natural language text, and the application should give an answer that is the best intelligent response to the input sentence. This process is repeated as the conversation continues [8] and the response is either text or speech. Building a Chatbot needs highly professional programming skills, and experienced developers to achieve the basic level of realism. There is a complicated development platform behind any Chatbot which will only be as good as its knowledge base which maps a user’s words into the most appropriate response. The bot developer usually builds the knowledge base as well. However, there are some platforms which provide a learning environment. Writing a perfect Chatbot is very difficult because it needs a very large database and must give reasonable answers to all interactions. There are a number of approaches to create a knowledge base for a Chatbot and include writing by hand and learning from a corpus. Learning here means saving new phrases and then using them later to give appropriate answers for similar phrases [9]. Designing a Chatbot software package requires the identification of the constituent parts. A Chatbot can be divided into three parts: Responder, Classifier and Graphmaster (as shown in Figure. 1) [10], which are described as follows:

1) Responder: it is the part that plays the interfacing role between the bot’s main routines and the user. The tasks of the responder are: transferring the data from the user to the Classifier and controlling the input and output.

2) Classifier: it is the part between the Responder and the Graphmaster. This layer’s functions are: filtering and normalising the input, segmenting the input entered by the user into logical components, transferring the normalised sentence into the Graphmaster, processing the output from the Graphmaster, and handling the instructions of the database syntax (e.g. AIML).

3) Graphmaster: is the part for pattern matching that does the following tasks: organising the brain’s contents, storage and holding the pattern matching algorithms.

4. Technical approaches and algorithms

4.1. Pattern Matching

This is by far the most common approach and technique used in chatbots. Variations of some pattern matching algorithm exist in every existing chatbot system. The pattern matching approaches can vary in their complexity, but the basic idea is the same. The simplest patterns were used in earlier chatbots such as ELIZA and PC Therapist. For example: Pattern: “I need a ?X”

Response: “What would it mean to you if you got a ?X?”

4.2. Parsing

Textual Parsing is a method which takes the original text and converts it into a set of words (lexical parsing) with features, mostly to determine its grammatical structure (Figure 1). On top of that, the lexical structure can be then checked if it forms allowable expression (syntactical parsing).

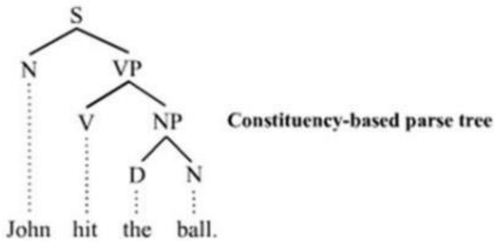

Figure 1: Example of a Parse Tree

The earlier parsers were very simple, looking for recognizable keywords in allowed order. Example of such parsing would be that sentences “please take the gold” and “can you get the gold” would be both parsed into “take gold”. With this approach the chatbot with a limited set of patterns can cover multiple input sentences.

The more complicated parsers used in latter chatbots do the complete grammatical parsing of the natural language sentences.

4.3. Markov Chain Models

The Idea behind Markov Chain Models is that each occurrence of a letter or a word in some textual dataset occurs with a fixed probability. The order of a model means how many consecutive occurrences the model takes into the account. For example if an input text is “agggcagcgggcg”, then the Markov model of order 0 predicts that letter „a‟ occurs with a probability 2/13. The model with order 1 would state that each letter still occurs with a fixed probability, but that probability depends on the letter before. In chatbots the Markov Chain Models were being used to construct responses which are probabilistically more viable and thus more correct. In some cases (HeX) these models were even used to generate a nonsense sentence that sounds right, as a failback method.

4.4. Ontologies (semantic nets)

Ontology or semantic network as it is called in some chatbot systems is a set of hierarchically and relationally interconnected concepts. These concepts can have natural language names and can be used directly in chatbots, to figure out hyponyms, synonyms and other relations between the concepts. Example of such an ontology which is often used or at least tried to be used in chatbots is OpenCyc3 [11].The advantage of the ontologies is that the concepts are interconnected into a graph, which enables computers to search through and using special reasoning rules even imply new statements (reasoning).

4.5. AIML

AIML‟s syntax is XML based (the main technology used in A.L.I.C.E bots) and consists mostly of input rules (categories) with appropriate output. The pattern must cover the entire input and is case insensitive. It is possible to use a wildcard (*) which binds to one or more words. The simplest example of it can be written like seen on Figure 2. Due to simple and effective explanation, this and as well the other examples were taken from the paper Beyond Façade: Pattern Matching for Natural Language Applications [12].

```
<Category>

<pattern> I NEED HELP * </pattern>

<template>Can you ask for help in the form of a question?

</template>

</category>
```

Figure 2: Simple AIML rule (pattern).

The real power of AIML lies in its ability to recursively call itself [12][13]. It can submit input to itself using the tag and the contents of * using . Example of such recursion can be seen on Figure 3, where the AIML engine forwards everything before the phrase “right now” to another pattern. The second pattern then forwards everything after the phrase “can you please”.

```
=> Can you please tell me what LINUX is right now?

<category>

<pattern> * RIGHT NOW <\pattern>

<template> <srai><star/></srai></template>

</category>

=> CAN YOU PLEASE TELL ME WHAT LINUX IS

and then

<category>

<pattern> CAN YOU PLEASE * <\pattern>

<template> <srai> Please <star/></srai></template> </category>
```

Figure 3: AIML recursion.

AIML allows chatbots to have topics which give it a way to prioritize the patterns. It has the pattern as well, which if it matches the output of the previous sentence it has priority over the other rules.

4.6. ChatScript

ChatScript aims to be a successor of the AIML language. It focuses on the better syntax which makes it easier to maintain. It fixes the zero word matching problems and introduces a bunch of additional functionalities such as concepts, continuations, logical and/or, variables, fact triples and functions. With these functionalities it tries to cover the need for ontologies inside the scrip itself. Example of a script defining a concept of meat and one pattern can be seen on figure 4.

```
concept: ~meat ( bacon ham beef meat flesh veal lamb

chicken pork steak cow pig )

s: ( I love ~meat ) Do you really? I am a vegan.
```

Figure 4: Chatbot concept definition and simple pattern

4.7. SQL and relational database

SQL and relational database is a technique used recently in Chatbot design in order to make the Chatbot remember previous conversations.

4.8. Markov Chain

Markov chainis used in Chatbots to build responses that are more applicable probabilistically and, consequently, are more correct. The idea of Markov Chains is that there is a fixed probability of occurrences for each letter or word in the same textual data set [14].

4.9. Language tricks

Language tricks these are sentences, phrases, or even paragraphs available in Chatbots in order to add variety to the knowledge base and make it more convincing. The types of language tricks are:

- Canned responses.
- Typing errors and simulating key strokes.
- Model of personal history.
- Non Sequitur (not a logical conclusion)

Each of these language tricks is used to satisfy a specific purpose and to provide alternative answers to questions.

5. Language approaches and tricks

5.1. Non Sequitur

Non sequitur (Latin) is an argument that has conclusions which does not imply from its premises. Example from everyday speech would be: “Life is life and fun is fun, but it’s all so quiet when the goldfish die.”

5.2. Simulating keystrokes and typing errors

The chat protocol that is used in Loebner Competitions works in a way that the judges see the sentences as they are being typed. This forces the chatbots to “pretend” they are tying word by word. Some of the bots even fake the spelling mistakes and backspacing. This protocol is one of the most debatable

5.3. Canned responses

Canned responses are predefined (hard coded) responses to questions. To some extent all of the chatbots patterns could be counted as canned responses if bot only uses these. This would vastly increase the number of patterns and would make them even more unmanageable, so these responses are usually used only for things which cannot be covered with the main chatbot technology.

5.4. Model of personal history

With the goal for a chatbot to appear more convincing, developers are inserting a personal story (imaginary or based on a real person) into chatbot responses. This includes memories from the past, childhood stories, parents, interests, political and religious views etc.

6. Industrial Chatbots

Industrial chatbots helps us to take decisions regarding stock maintainance, machinery expansion, demand detection, cost cutting, production rates etc. Chatbots analyse these problems and help us to take right decisions. Thus human biases are removed from the process. The analysis of the data is done using the artificial neural networks in machine learning.

7. Artificial neural networks

A neural network is a machine learning algorithm based on the model of a human neuron. The human brain consists of millions of neurons. It sends and process signals in the form of electrical and chemical signals. These neurons are connected with a special structure known as synapses. Synapses allow neurons to pass signals. From large numbers of simulated neurons neural networks forms.

An Artificial Neural Network is an information processing technique. It works like the way human brain processes information. ANN includes a large number of connected processing units that work together to process information. They also generate meaningful results from it.

We can apply neural network not only for classification. It can also apply for regression of continuous target attributes.

Neural networks find great application in data mining used in sectors. For example economics, forensics, etc and for pattern recognition. It can be also used for data classification in a large amount of data after careful training.

A neural network may contain the following 3 layers:

- Input layer – The activity of the input units represents the raw information that can feed into the network.

- Hidden layer – To determine the activity of each hidden unit. The activities of the input units and the weights on the connections between the input and the hidden units. There may be one or more hidden layers.

- Output layer – The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

8. Conclusion

In our paper we are using the machines condition, market sales of the product, and production rates in the artificial neural networks. The artificial neural networks helps in the accuracy of the prediction and chatbots are used to communicate to solve the customer queries.

9. References

[1] Colby K. M., 1975, Artificial Paranoia: A computer program for the study of natural language communication between man and machine, Communications of the ACM, vol. 9, pp. 36-45.

[2] J. P. Campbell Jr, “Speaker recognition: a tutorial,” Proceedings of the IEEE, vol. 85, no. 9, pp. 1437-1462, 1997.

[3] C.-H. Lee, "From knowledge-ignorant to knowledge-rich modeling: a new speech research paradigm for next generation automatic speech recognition", 2004.

[4] V. Bhargava, and N. Maheshwari, “An Intelligent Speech Recognition System for Education System,” 2009.

[5] E. Loper, and S. Bird, "NLTK: The natural language toolkit." pp. 63-70, 2002.

[6] S. Bird, "NLTK: the natural language toolkit." pp. 69-72, 2006.

[7] A. M. Galvao, F. A. Barros, A. M. Neves, and G. L. Ramalho, "Personaaiml: An architecture developing chatterbots with personality." pp. 1266-1267, 2004.

[8] J. Ratkiewicz, “Evolutionary Sentence Combination for Chatterbots Dana Vrajitoru Computer and Information Sciences Indiana University South Bend, 1700 Mishawaka Ave,” 2004.

[9] M. J. Pereira, and L. Coheur, “Just. Chat-a platform for processing information to be used in chatbots,” 2013.

[10] D. J. Stoner, L. Ford, and M. Ricci, “Simulating Military Radio Communications Using Speech Recognition and Chat-Bot Technology,” 2003.

[11] Lenat, D. B., 1995. Cyc: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM, 38(22).

[12] Wilcox, B., 2011. Beyond Façade: Pattern Matching for Natural Language Applications, http://www.gamasutra.com/view/feature/134675/beyond_fa%C3%A7ade_pattern_matching_.php

[13] Wallace R., 2003. The elements of AIML style. ALICE AI Foundation.

[14] D. Mladenić, and L. Bradeško, “A survey of chabot system through a Loebner prize competition,” 2012.