Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 3955–3965

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Integrating Experience-Based Knowledge Representation and Machine Learning for Efficient Virtual Engineering Object Performance

Syed Imran Shafiq[a]*, Cesar Sanin[b], Edward Szczebicki[c]

[a]*Faculty of Engineering and Technology, Aligarh Muslim University, Aligarh 202002, India*

[b]*Faculty of Engineering and Built Environment, University of Newcastle, Newcastle, Australia*

[c]*Faculty of Management and Economics Gdansk University of Technology, Gdansk, Poland*

Abstract

Machine learning and Artificial Intelligence have grown significant attention from industry and academia during the past decade. The key reason behind interest is such technologies capabilities to revolutionize human life since they seamlessly integrate classical networks, networked objects and people to create more efficient environments. In this paper, the Knowledge Representation technique of Set of Experience Knowledge Structure (SOEKS) and Decisional DNA (DDNA) is applied to facilitate Machine Learning. For effective and efficient decision-making in Machine Learning, the environment's own experience is captured, stored and reused using the DDNA technique. The proposed approach is implemented on practical test cases like a Chatbot. Decisional DNA gathers explicit experiential knowledge based on formal decision events and uses this knowledge to support decision-making processes. The experimental test and results of the presented implementation of Decisional DNA Chatbot case studies support it as a technology that can improve and be applied to the technology, enhancing intelligence by predicting capabilities and facilitating knowledge engineering processes.

Keywords: Knowledge Representation; Set of Experience Knowledge Structure (SOEKS); Decisional DNA (DDNA); Chatbot

* Corresponding author. Tel.: +918755286666;
E-mail address: sishafiq.bp@amu.ac.in

10.1016/j.procs.2021.09.170

1. Introduction

An enormous data transfer and decision-making is being accomplished on the internet on a daily basis [1]. Few examples to give an idea about this are: about 6 billion searches on google, 300 million pictures are uploaded on Facebook, 17 million orders are fulfilled by Amazon, 15 million cab rides completed by Uber, 4 trillion money moved by Citibank. Behind each of these transactions run powerful Machine Learning (ML) and AI Algorithms, using data to predict and optimize recommendations for the user, Just like the human brain.

Machine Learning is simply a process of using relevant/representative data to predict future actions. It is an iterative, methodical exploration of data to derive business insights and actions. Machine learning algorithms are estimated to substitute a significant percentage of jobs across the globe in the coming years. Machine learning applications are self-modifying and extremely automated, which continue to advance over time with minimal human interference as they learn with more data [2]. Few commonly used examples of ML use on an almost daily basis are:

- Virtual Personal Assistants- Smart Speakers: Amazon Echo and Google Home; Smartphones: Samsung Bixby on Samsung S8; Mobile Apps: Google Allo
- Predictions while Commuting- Traffic prediction, online transportation network
- Videos Surveillance-face recognition, behavior monitoring
- Social Media Services- People you may know, likes and dislikes recommendations
- Email Spam and Malware Filtering
- Online Customer Support

To overcome the complex nature of various data problems, specialized machine learning algorithms and models have to be developed to solve these problems perfectly. However, several limitations are encountered in the implementation of the machine learning approach; some of them are:

- *ML Algorithms require huge amount Training Data*

Artificial Intelligence (AI) systems are trained, not programmed. Thus, that they require huge amounts of data to perform complex jobs at the level of humans. Inspite of the fact that data is being created at a rapid pace and the robust computing power needed to competently process it is available, enormous data sets are not simple to create. Deep learning uses an algorithm called backpropagation that regulates the weights between nodes to confirm an input translates to the right output. Supervised learning happens when neural nets are trained to recognize photographs, for example, using millions of past labeled examples. And every variation in an assigned task calls for another large data set to conduct extra training. The significant limitation is that neural networks require excessive 'brute force' to function at a level similar to human intellect [3].

This challenge can be overcome by coupling deep learning with unsupervised learning techniques that do not rely on labeled training data. For example, deep reinforcement learning models learn by trial and error as opposed to via example.

- *Labeling Training Data Is a Laborious/Tedious Process*

Labeling is an essential stage of data processing in supervised learning. This model training style employs predefined target attributes from historical data. Data labeling is the process of cleaning up raw data and organizing it for cognitive systems to ingest. Deep learning requires lots of labeled data. If unlabeled data is fed into the AI, it is not going to get smart with time. An algorithm could develop the ability to make decisions, perceive, and behave in a way that is compatible with the environment within which it is required to steer in the future if a human mapped target attributes for it.

To set up what is in the data, a time-taking process of manually spotting and labeling items is required. However, new techniques are coming up, like in-stream supervision, where data is labeled during natural usage. High-quality data collection from users can be utilized to improve machine learning over time [4].

- *Machines Cannot Clarify or Explain Themselves*

A majority of AI-based models deployed are based on statistical machine learning that depends on large training data to build a statistical model. This is the main cause why the adoption of some AI tools is still less in areas where explainability is vital. Whether the decision is bad or good, having visibility into why/how it was made is pivotal so that the human expectation can be brought in line with how the algorithm actually acts. There are procedures that can be used to elucidate complicated machine learning models like neural networks. A nascent technique is Local Interpretable Model-Agnostic Explanations (LIME), which aims to pinpoint the sections of input data a trained ML model based on most to create predictions by feeding inputs similar to the initial ones and noting how these predictions vary [5].

- *There is Bias in the Data*

As machine learning and AI algorithms are deployed, there will be more instances in which potential bias finds its way into data sets and algorithms. In some cases, models that are apparently performing well maybe actually picking up noise in the data. Not only is transparency important, but unbiased decision-making builds trust. The accuracy of an AI solution is based on the quality of its inputs. For instance, facial recognition has had a huge impact on law-enforcement, social media, human resources and other applications. However, biases in the data sets provided by facial recognition models can result in inaccurate outcomes. If the training data is not unbiased/neutral, the results will amplify the bias and discrimination that lies in the data set. The ideal method to mitigate such risks is by collecting data from random and multiple sources. A heterogeneous dataset restricts the exposure to bias and results in better quality ML solutions [6].

- *AI Algorithms Do Not Collaborate*

Despite the advancements in deep learning and neural networks, AI models still lack the capability to generalize conditions that are different from the ones they experienced in training. AI models have problems transferring their experiences from one set of a situation to the other. Thus anything a model has achieved for a specific case will only be applicable to that use case. As a result, organizations are compelled to commit resources to train other models, even when the use cases are comparatively similar. A solution to this situation comes in the form of transfer learning [7].

Machine learning works on a concept: understanding with experiences. Also, SOEKS-DDNA is an experience-based knowledge representation technique. This technique obtains knowledge from one task can be used in domains/situations where little labeled data is available. Therefore, through this generalized mature approach, organizations will have the ability to build new applications more rapidly [8]. In this study, SOEKS-DDNA is applied on the various data classification model of machine learning to make more effective.

2. SET OF EXPERIENCE KNOWLEDGE STRUCTURE (SOEKS) AND DECISIONAL DNA (DDNA)

This section presents a description of the Set of Experience Knowledge Structure (SOEKS) and Decisional DNA (DDNA), argumentation for a knowledge representation, composition, configuration and metrics. SOEKS is a combination of filtered and amalgamated information obtained from formal decision events. It facilitates the effective explicit representation of decisional experience taken from different technologies. SOEKS comprises variables, functions, constraints and rules associated with a DNA shape, allowing the construction of enterprises' fingerprints called Decisional DNA. SOEKS possesses characteristics that potentialize it as a more precise knowledge representation in a world guided by sensitive dependence and uncertainty. That is, SOEKS is a suitable representation for explicit decisional knowledge that has been gifted with capabilities to manage uncertainty, preciseness and incompleteness. Furthermore, SOEKS extends into the so-called DDNA due to the characterization and aggrupation of SOEKS into different classes termed decisional chromosomes. Such decisional chromosomes simulate specialized genes that, when placed together, create the decisional experience of an enterprise, the Decisional DNA [9] [10] [11]; Using past data/patterns to predict future actions.

2.1 SOEKS or SOE

Four basic components surround decision-making events. These four components are variables, functions, constraints, and rules and constitute the proposed knowledge structure [9].

i) Variables

An action An is interpreted by the variables that define it, and a change of these variables means a change from state Sj to the state Si, through the action An. Thus, a function τ, which transforms variables through a set of actions Ai, is given. Subsequently, τ transforms a set of variables Vj, before an action is executed into Vi. Given a set of variables Vi = {xi1, xi2, . . . , xin} and Vj = _xj1, xj2, . . . , xjn_, it can be said in mathematical terms that:

$$\tau\_Vj\_ = Vi \tag{1}$$

Thus, initially, the set of variables involved in the process of decision making are included in the Set of Experience Ei, defining it as:

$$SOE\ Ei = (Vi) \tag{2}$$

Vi is a set of variables that contain the cause value of each variable from a state Sj, as well as the effect value of each variable from a state Si.

ii) Functions

In decision-making, the concept of an objective function is well known. An objective function is a relation among the variables that illustrates the method to reach an optimal state. Human Solving Paradigm (HSP) models accept that the stimulus-response paradigm (current and desired state) of decision-making demands that choices emanate from goals. Thus, in this order of ideas, objective functions arise out of the desired state because they describe it by means of defining how the cause values are changed into effect values. Hence, a state Si can be defined in terms of a given set of objective functions O1, …, Om:

$$Si = Fi = \{[min/max](O1,O2, . . . ,Om)\} \tag{3}$$

Each Oi is an objective function, which relates variables belonging to Vi. Functions now redefine the SOE Ei of Eq. (2) as:

$$SOE\ Ei = (Vi,\ Fi) \tag{4}$$

iii) Constraints

Constraints appear as a limitation in our decision-making process. They are part of our universe of experience and the psychological space. Hence, they should be included in the SOEKS. Given Ci as a set of constraints of the universe of constraints C, the SOE given in Eq. (4) is now redefined as:

$$SOE\ Ei = (Vi,\ Fi,Ci) \tag{5}$$

iv) Rules

Rules are relationships between a condition and a consequence connected by the statements IF-THEN-ELSE. They can operate two or more variables and one or many functions in both conditions and consequences, either the THEN and/or the ELSE. Rules, as well as constraints and functions, are included in the SOEKS. Given Ri, a set of rules of the universe of rules R, the SOE of Eq. (4) is redefined as:

$$SOE\ Ei = (Vi,\ Fi,Ci,Ri) \tag{6}$$

2.2 Decisional DNA

A group of Sets of Experience of the same category comprise a kind of chromosome, as DNA does with genes. These chromosomes or groups of Sets of Experience could make a "strategy" for a category, i.e., a decisional area of the organization. They are a group of ways to operate when making decisions. Each module of chromosomes forms an entire inference tool and provides a schematic view for knowledge.

Moreover, SOE allows the construction of companies' fingerprints. In other words, the Decisional DNA consists of stored experienced decision events (i.e., experiential knowledge) that can be grouped according to areas of decision or categories. Each SOE built after a formal decision event can be categorized and acts similarly to a gene in DNA. A gene guides hereditary responses in living organisms, as a SOE directs responses of certain areas of the organization. Furthermore, assembled genes create chromosomes and human DNA, as groups of categorized SOE create decisional chromosomes and Decisional DNA.

For instance, one SOE from a formal decision event represents a portion of an organization's Decisional DNA. It is a gene that guides decision-making. This portion of Decisional DNA belongs to a decisional chromosome of a certain type; it is a categorized SOE. Let's say it is a gene in a decisional chromosome of salary increment (that is, a decision associated with the increment of salary). Subsequently, many of these salary increment decisions, as portions of Decisional DNA, will comprise a whole chromosome, which is never complete because there will be new decisions added to this category from time to time. Multiple decisional chromosomes of different kinds, such as marketing decisions, production decisions, human resources, and many more kinds of decisions, will comprise the decisional genetic code of an organization. It is the Decisional DNA of an organization.

In conclusion, a SOEKS acts as a representation for explicit experiential knowledge according to the world it perceives from formal decision events. This SOEKS is composed of four components, which are uniquely combined, and can be collected, classified, organized, and even evolved according to their efficiency, grouping them into decisional chromosomes. Chromosomes are groups of Sets of Experience that can comprise a decisional strategy for a specific area, constructing the Decisional DNA of an organization [9] [10] [11].

3. DDNA AND MACHINE LEARNING IN CHATBOT

A Chatbot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent. The major challenge of Chatbot is its inability to understand the intend of the customer.

To overcome this problem DDNA powered Chatbot to be used in manufacturing environment is proposed in this study; consequently, the bots are trained with the actual logs of conversation data, which have to be stored in a structured format. The conversation logs are stored in the SOEKS format and Python programs are written to analyze what the customers are trying to ask. Machine learning models are combined with DDNA tools to reply to the best-suited answer to match the questions. For example: If an operator is asking "Where is a machine inspection receipt?" and "I have not received a machine inspection receipt", mean the same thing. Algorithm power is in training the models so that the Chatbot is able to connect both of those queries to correct intent and as output produces the correct answer. If there is no extensive data available, DDNA APIs data can be used to train the Chatbot. Figure 1 shows that the architecture of DDNA based Chatbot.

To fully exploit the capabilities of DDNA, Chat is modelled as a Virtual Engineering object (VEO). Through a VEO model the Chatbot experience is stored in SOEKS format in four different modules Characteristics, Requirement, Connections, PresentState and Experience [12] [13].

Equations and formulae should be typed in MathType, and numbered consecutively with Arabic numerals in parentheses on the right hand side of the page (if referred to explicitly in the text). They should also be separated from the surrounding text by one space.

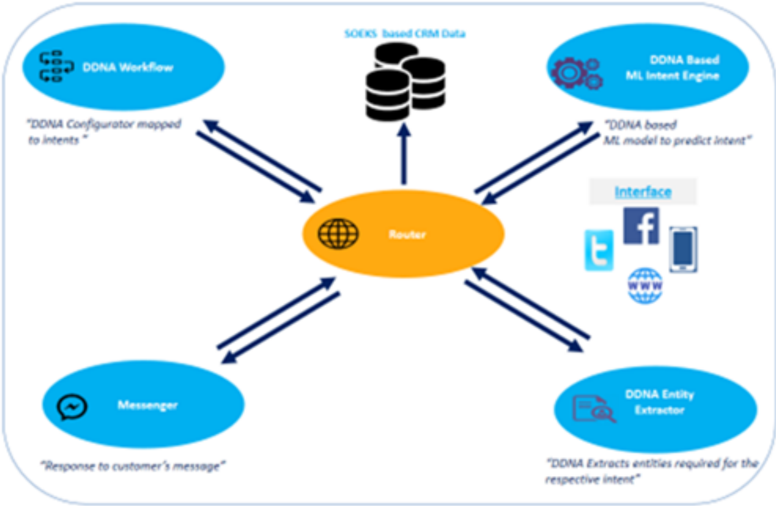

Fig.1. DDNA based VEO-Chat Bot Architecture

DDNA based training of a Chatbot happens at a larger scale and much faster; with thousands of machine conversation logs and from those logs, the Chatbot is able to understand what type of question requires what type of answers. Figure 2 demonstrate that DDNA based training of a Chatbot happens at a larger scale and much faster; with thousands of machine conversation logs and from those logs, the Chatbot is able to understand what type of question requires what type of answers.

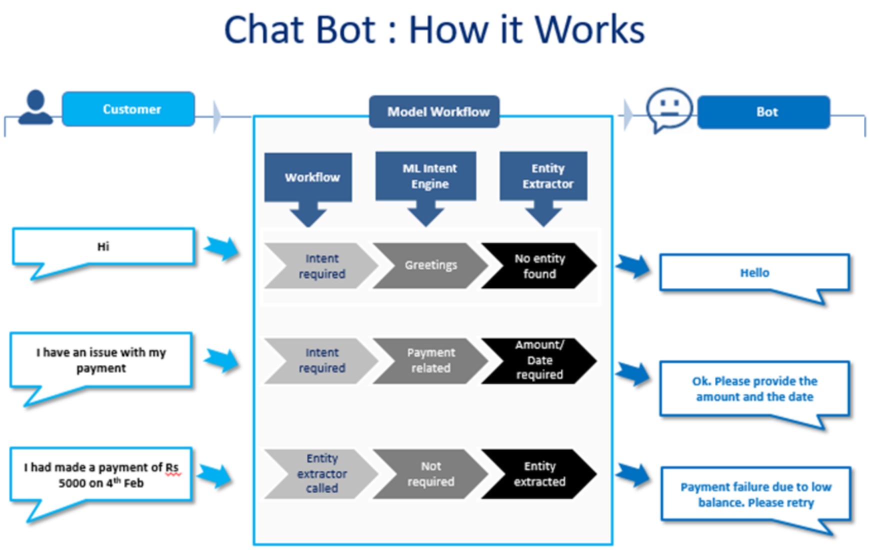

Fig. 2. DDNA based VEO-Chat Bot working

The proposed DDNA Chatbot model can facilitate in more meaningful conversation in all the following three classifications methods generally adopted:

3.1. Pattern Matchers

The text is classified using SOEKS structure based on the Artificial Markup Language (AIML) pattern matching and the suitable response is searched from the DDNA data.

A simple pattern matching example:

```
<aiml version = "1.0.1" encoding = "UTF-8"?>
<category>
<pattern> WHAT TOOL IS USED FOR KNURLING OPERATION</pattern>
<template>Knurling Tool is used for knurling operation</template>
</category>

<category>
<pattern> DO YOU KNOW WHAT TOOL IS USED *</pattern>
<template>
<srai> TOOL IS USED</star></srai>
</template>
</category>
</aiml>
```

The machine then gives and output:

```
Operator: What tool is used for Knurling Operation?
Robot: Knurling Tool is used for Knurling Operation.
```

Since the tool is in associated pattern with the operation in DDNA database, Chatbot can answer the question posed by the operator. Anything which is not in line with the associated pattern, Chatbot cannot answer; in that case, advance level algorithms are used.

3.2. Algorithms

In this approach, DDNA database provides a unique pattern for each question posed and suggests an appropriate response. The 'Reductionist' approach to obtain a simplified solution is adopted by creating a hierarchical SOEKS structural out of the available combination of patterns. For Natural Language Processing (NLP) and text classification, Multinational Naïve Bayes is the standard algorithm. Each word in a new input sentence is counted for its occurrence and is considered for its commonality and every class is assigned a score. The class with the highest value score is assigned to the input sentence. For example, the Sample Training set

```
class: greeting
"good morning"
"How you doing?"
"hi there"
```

Few sample Input sentence classification:

```
 input: "Hi good morning"
term: "hi" (no matches)
Term: "good" (class: greetings)
term: "morning" (class: greetings)
classification: greetings (score=2)
```

With the help of the equation, word matches are determined for given sentences for each class. Classification score associates the class with the highest term matches. The score represents which intent is most likely to the sentence but does not guarantee it is the best match. The highest score provides the relativity base only.

3.3. Artificial Neural Networks (ANN)

Neural Networks are a method of calculating the output from the input having weighted connections which are determined from repeated iterations while training the data. Every step in the training data amends the weights resulting in the output with precision.

Each input sentence is broken down into different words/parts and each word then is treated as an input for the neural networks. The weighted connections are calculated by different iterations through the training SOEKS data multiple times. Each iteration improving the weights to making them accurate and precise. The trained data of neural networks is less code and a more comparable algorithm. For the cases in which comparably small sample, where the training sentences have 250 different words and 25 classes, then that would be a matrix of 250×25. But when this matrix size is increased by n times more incrementally and can cause a large number of errors. In this kind of scenario, processing power and speed should be significantly high.

3.4. Natural Language Processing (NLP)

Natural Language processing Chatbot takes a combination of multiple steps to transform the customer's text or speech into structured SEOKS data that is utilized to select the related answer. Steps involved in Natural Language Processing are:

- Sentiment Analysis: Tries to determine if the user has a good experience or if after some point the chat should be transferred to human assistance.
- Tokenization: The NLP splits a string of words into tokens that are linguistically illustrative or are differently helpful for the application.
- Entity Recognition: The Chatbot program model searches with DDNA query mechanism, for categories of words, like the name of the product/operation, the operator's name, whichever data is required.
- Normalization: The program model processes the text in an attempt to find common typographical errors that might serve the user intent. This gives Chatbot users a more human-like effect.
- Dependency Parsing: The Chatbot model looks for the objects and subjects- verbs, nouns and common phrases in the user's text to find related phrases that users might be trying to convey.

Like most of the ML Applications, the Chatbot is also connected to the Database. The knowledge base/database of information is used to feed the Chatbot with the information required to give an appropriate response to the user. Data of user's past activities and whether the Chatbot was able to match their questions is stored in the database. NLP translates common human language into information with a mixture of patterns and text that can be mapped to find applicable responses in real-time.

There are NLP applications programming interfaces that are employed to build the Chatbots and make it feasible for all types of businesses, large, medium and small scale. The important point here is that Smart Bots not only have the potential to help increase customer base by improving the customer care/support services but also boosts sales as well as profits. They are an opportunity for many companies to reach a huge customer base.


```
{"intent": [
        {"tag": "greeting",
         "pattern": ["Hello", "Hi there", "How are you", "Is anyone
there?","Hey","Hola", "Good day"],
         "response": ["Hi there, how can I help?", "Hello, thanks for asking", "Good
to see you again"],
         "context": [""]
        },
        {"tag": "goodbye",
         "pattern": ["Goodbye", "Bye", "See you later", "Nice chatting to you, bye",
"Till next time"],
         "response": ["Have a nice day", "See you!", "Bye! Come back again soon."],
         "context": [""]
        },
```

```
        {"tag": "thanks",
         "pattern": ["Thanks", "Thank you", "That's helpful", "Awesome, thanks",
"Thanks for helping me"],
         "response": ["Happy to help!", "Any time!", "My pleasure"],
         "context": [""]
        },
        {"tag": "no answer",
         "pattern": [],
         "response": ["Sorry, can't understand you", "Please give me more info", "Not
sure I understand"],
         "context": [""]
        },
        {"tag": "options",
         "pattern": ["How you could help me?", "What you can do?", "What help you
provide?", "How you can be helpful?", "What support is offered"],
         "response": ["I can guide you through Machine Operations list, Machine
Operation tracking, Tools and Raw Materials", "Offering support for Adverse Cutting
parameters, Tools Settings and Work Pieces Material"],
         "context": [""]
        },
        {"tag": "cutting parameters",
         "pattern": ["How to check Adverse cutting parameters?", "Open adverse cutting
parameters module", "Give me range of parameters causing adverse behavior", "List all
tools setting suitable for with adverse cutting", "Which cutting operation setting
don’t have adverse machining ?" ],
         "response": ["Navigating to Adverse cutting operation module"],
         "context": [""]
        },
        {"tag": "Surface Roughness",
         "pattern": ["Open Surface Roughness module", "Machine Settings related to
Surface Roughness", "Surface Roughness required data entry", "I want to log Surface
Roughness results", "Surface Roughness data management" ],
         "response": ["Navigating to Surafce Roughness module"],
         "context": [""]
        },
        {"tag": "surface roughness search",
         "pattern": ["I want to search for Surface Roughness result history", "Surface
Roughness for machine", "Load machine Surface Roughness result", "Show blood pressure
results for patient", "Find Surface Roughness results by Machine_ID" ],
         "response": ["Please provide Machine ID", "Machine ID?"],
         "context": ["search_Surface Roughness_by_machine_id"]
        },
        {"tag": "search_surface roughness_by_machine_id",
         "pattern": [],
         "response": ["Loading Surface Roughness result for Machine"],
         "context": [""]
        },
        {"tag": "cuttingTool_search",
         "pattern": ["Find me a cuttingTool", "Find cuttingTool", "List of cuttingTool
available", "Locate cuttingTool", "Search cuttingTool" ],
         "response": ["Please provide cuttingTool name"],
         "context": ["search_ cuttingTool _by_name"]
        },
        {"tag": "search_ cuttingTool_by_name",
         "pattern": [],
         "response": ["Loading cuttingTool details"],
         "context": [""]
        },
        {"tag": "operator_search",
         "pattern": ["Lookup for Operator", "Searching for Operator for a job", "I
```

```
want to search Operator data", "Operator lookup for job", "Looking up operator
details" ],
          "response": ["Please provide operator name or location"],
          "context": ["search_operator_by_params"]
         },
         {"tag": "search_operator_by_params",
          "pattern": [],
          "response": ["Please provide operator type"],
          "context": ["search_operator_by_type"]
         },
         {"tag": "search_operator_by_type",
          "pattern": [],
          "response": ["Loading operator details"],
          "context": [""]
         }
    ]
}
```

4. Results and discussion

The proposed SOEKS-DDNA based chatbot architecture was implemented in Python programming language on a Windows 10 operating system. The SOEKS-DDNA consists of SOEs representing Characteristics, Functionality, Requirements, Present State, Connections, and Experience, each having variables, functions, and constraints. For testing purposes, we query the repository of 2500 SOEKS.

The parsing process of the VEO DDNA was executed, producing a parsing time of 703.0 ms. This is considered a very good time, and it gives the impression to the chatbot user that he is having interaction with a human in real-time.

The sample code shows the sample queries that were executed to find the most similar SOEKS. For example, a sample query to find out cutting parameters for a specific material. When this query is executed, DDNA returns the top 5 most similar SOEKS; in this particular case, it is VEO_Code no 1, 13, 5, 57, 20 having similarities 0.2791, 0.2809, 0.2810, 0.2821, 0.2832, respectively. On a similar pattern, other queries are formulated and evaluated.

5. Conclusion

Knowledge representation technique of DDNA and SOEKS is integrated with machine learning tools to enhance Chat Bot applications. A chat Bot for manufacturing environment is modelled as a VEO, which can capture, store and use the experiential knowledge for better future decision making. It is observed that through VEO, the experience of Chat Bot can be stored in a SOEKS structured format; this facilitates to recognize the intend and pattern of the queries, which results in better responses.

In the future, evaluation of the proposed system and its performance will be compared with the various other techniques used in the chatbot. Moreover, the SOEKS-DDNA chatbot model can be extended for the higher levels of manufacturing pyramid, i.e. computer integrated manufacturing (CIM) and Flexible Manufacturing System (FMS).

References

[1] Agarwal R., Dhar V. Editorial – big data, data science, and analytics: the opportunity and challenge for is research Information Systems Research, 25 (3) (2014), pp. 443-448

[2] Schapire, R. E. (1990). The strength of weak learnability. Machine Learning, 5, 197–227.

[3] Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M. et al. Deep learning applications and challenges in big data analytics. Journal of Big Data 2, 1 (2015). https://doi.org/10.1186/s40537-014-0007-7

[4] Ian H. Witten, Eibe Frank, Mark A. Hall, Christopher J. Pal, Chapter 11 - Beyond supervised and unsupervised learning, Editor(s): Ian H. Witten, Eibe Frank, Mark A. Hall, Christopher J. Pal,Data Mining (Fourth Edition),Morgan Kaufmann, 2017, Pages 467-478, ISBN 9780128042915, https://doi.org/10.1016/B978-0-12-804291-5.00011-8.

[5] Changhun Lee, Chiehyeon Lim, From technological development to social advance: A review of Industry 4.0 through machine learning, Technological Forecasting and Social Change, Volume 167, 2021, 120653, ISSN 0040-1625, https://doi.org/10.1016/j.techfore.2021.120653.

[6] Tomáš Kliegr, Štěpán Bahník, Johannes Fürnkranz, A review of possible effects of cognitive biases on interpretation of rule-based machine learning models, Artificial Intelligence, Volume 295, 2021, 103458, ISSN 0004-3702, https://doi.org/10.1016/j.artint.2021.103458.

[7] Donghee Shin, The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI, International Journal of Human-Computer Studies, Volume 146, 2021, 102551, ISSN 1071-5819, https://doi.org/10.1016/j.ijhcs.2020.102551.

[8] Cesar Sanin, Zhang Haoxi, Syed Imran Shafiq, Md Maqbool Waris, Caterine Silva de Oliveira, Edward Szczerbicki, Experience based knowledge representation for Internet of Things and Cyber Physical Systems with case studies, Future Generation Computer Systems (Elsevier), Available online 10 February 2018, ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.01.062.

[9] S. I. Shafiq, C. Sanin, and E. Szczerbicki, "Set of Experience Knowledge Structure (SOEKS) and Decisional DNA (DDNA): Past, Present and Future," Cybernetics and Systems (Taylor and Francis), vol. 45, pp. 200-215, 2014.

[10] Shafiq, Syed Imran, C. Sanín, E. Szczerbicki and C. Toro (2017). 'Towards an Experience Based Collective Computational Intelligence for Manufacturing.' Future Generation Computer Systems (Elsevier). Vol 66 (89-99). http://dx.doi.org/10.1016/j.future.2016.04.022

[11] S, Cesar, Shafiq Syed Imran, W. M. Maqbool, T, Carlos. S. Edward. (2017) 'Manufacturing collective intelligence by the means of Decisional DNA and virtual engineering objects, process and factory'. Journal of Intelligent & Fuzzy Systems (IOS Press), vol. 32, no. 2, pp. 1585-1599

[12] S. I. Shafiq, C. Sanin, C. Toro, and E. Szczerbicki, "Virtual Engineering Object (VEO): Toward ExperienceBased Design and Manufacturing for Industry 4.0," Cybernetics and Systems (Taylor and Francis), vol. 46, pp. 35-50, 2015.

[13] Shafiq, Syed Imran, C. Sanín, C. Toro and E. Szczerbicki (2015). 'Virtual Engineering Process (VEP): A Knowledge Representation Approach for Building Bio-Inspired Distributed Manufacturing DNA.' International Journal of Production Research (Taylor and Francis)) vol. 54, issue 23, pp. 7129-7142.