# ABSTRACT

Letters dominated the sooner era in communication. Then with emergence of Telephones and later on mobile phones, voice conversations dominated the communication. However, currently, with the emergence of web and much of social media, chat conversations are ruling the planet. consider your highest friend and raise yourself, have you ever talked additional or chatted more? therefore, with quality of chat in today’s world, several technologists visualized that chat couldn’t simply be a mode of communication between humans however additionally between an individual's and a pc. That’s what chat-bot is. In some cases, it's high-powered by machine learning (the additional you act with the chat-bot the smarter it gets). Or, additional normally, it is driven victimization intelligent rules (i.e. if an individual says this, respond with that).

### CONTENTS

### CHAPTER 1

### 1. INTRODUCTION

### 1.1 Existing system 02

### 1.2 Proposed system 02

### 1.3 System architecture 03

### 1.4 Hardware and Software tools 04

### CHAPTER 2

### 2. LITERATURE SURVEY

### 2.1 Literature survey 06

### CHAPTER 3

### 3. MODULES

### 3.1 User interface 11

### 3.2 API interface 11

### 3.3 Data flow diagram 12

### CHAPTER 4

### 4. PYTHON OVERVIEW AND ENVIRONMENT

### 4.1 Python overview 14

### 4.1.1 Python overview 14

### 4.1.2 History of Python 14

### 4.1.3 Python features 15

### 4.2 Python environment 16

### 4.2.1 Software requirements 16

### CHAPTER 5

### 5 DOMAIN SPECIFICATION

### 5.1 Machine learning 18

### 5.1.1 Applications 26

### 5.2 Deep learning 28

### 5.3 Reinforcement learning 28

### CHAPTER 6

### 6. ARTIFICIAL INTELLIGENCE AND TENSOR FLOW

### 6.1 Artificial intelligence 30

### 6.2 Tensorflow 34

### 6.2.1 Tensorflow architecture 36

### CHAPTER 7

### 7. ANACONDA NAVIGATOR AND NUMPY

### 7.1 Anaconda navigator 39

### 7.2 Numpy 41

### CHAPTER 8

### 8. RESULT AND CONCLUSION

### 8.1 Code 45

### 8.2 Result 50

### 8.3 Conclusion 51

### LIST OF FIGURES

### S.No. List of figures Pg.No

### 1. System architecture 08

### 2. Simplified architecture 08

### 3. Data flow diagram 12

### 4. Machine learning 19

### 5. Learning phase 20

### 6. Inference from model 20

### 7. Machine learning algorithms 21

### 8. Artificial intelligence 32

### 9. Average graph of AI, ML, DL 34

### 10. Result 50

**CHAPTER-1**

**INTRODUCRION**

# 1.1 EXISTING SYSTEM:

In Existing System, we have a chat bot that is in static type only. For example, we create a one corpus. Corpus is nothing but it’s a data set. in that dataset we have to add all the question and answer related to the which field we need and train the machine and after it give the reply answer according to the predication algorithm in the data set so, it’s a static type it gives reply according to the dataset answer.

**Disadvantage**

* Static Type
* User have to give question and answer
* Accuracy value is not very high
* Reply according to the dataset query’s

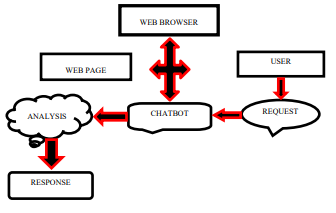
**1.2 PROPOSED SYSTEM:**

In proposed system we are connecting to a Wikipedia data base and we get the answer from the online whatever input given by user so, it’s a dynamic type reply no need of database answer will have retrieved form internet.

**Advantage**

* Dynamic type
* No need to give dataset
* Accuracy of answer is high

**1.3 SYSTEM ARCHITECTURE:**



**Fig.1**

**Simplified architecture**:

Wikipedia database

Question

Input

Answer

**Fig.2**

**1.4 HARDWARE AND SOFTWARE SPECIFICATION**

**Hardware:**

1. Windows 10 64 bit
2. RAM 4GB

**Software :**

1. Packages
2. Python

**CHAPTER-2**

**LITERATURE SURVEY**

* 1. **LITERATURE SURVEY:**

1. **M. D. Smucker and C. L. Clarke, “Time-based calibration of effectiveness measures,” in Proc. 35th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2012, pp. 95– 104.**

This paper presents a very-low-drift 0.18μ m CMOS time-based resistive-bridge sensor interface. It exhibits only 3.8 ppm/° C gain drift and 0.3 ppm/°C offset drift for the entire -40°C to 175°C temperature range using a single-temperature calibration scheme and no external accurate references nor components. The interface provides a 15 ENOB for a 100ms conversion time, consuming 3.41mW of power and 0.26mm 2 of active area. The holistic drift-resilience strategy combines time-based chopping and VCO tuning to remove the DC and low-frequency errors introduced by VCO non idealities and drift.

1. **T. Nguyen, et al., “MS MARCO: A human generated machine reading comprehension dataset,” in Proc. 30th Conf. Neural Inf. Process. Syst. Workshop, 2016, p. 10.**

During this analytical green engineering methods series, we are going to review green engineering system and process odelling, system analysis and design following sustainable lean six sigma principles. These principles are well established in quality circles and they apply in a very positive way to our sustainable green engineering effort. This tutorial offers and explains customizable and downloadable graphic templates that you can use and modify for process odelling, system odelling and system designing green engineering processes and systems. The tutorial focuses on the actual use of the templates and provides some examples. The purpose of this tutorial is to enable you to model complex green engineering systems, process inputs, outputs, resources and controls. You will see critical compliance aspects of products, processes and service systems by using customizable templates. The customizable templates provide you a set of architecture, format, syntax, some semantics, and a good head start when odelling. It provides guidance, some structure, as well as time saving since you are given templates and do not have to start from scratch.

1. **M. Keikha, J. H. Park, W. B. Croft, and M. Sanderson, “Retrieving passages and finding answers,” in Proc. Australasian Document Compute. Symp., 2014, pp. 81–84.**

Passage retrieval of Question Answering (QA) systems aims to find the text segments or sentences that may contain the exact answers for the given question. Previous studies on passage retrieval are mostly utilized a single function to calculate the relevance scores of passages. However, some research has proved that the relations between passages can be utilized to improve the accuracy of relevance evaluation. Hence, a passage retrieval method based on passage-passage graph model is proposed. A KNN-based question expansion method is proposed and then the candidate answer passages are retrieved based on the expanded question model. The passage graph is constructed based on the similarities between the candidate answer passages. Finally, a graph-based ranking model is utilized to re-calculate the relevance scores of the answer passages and the ranking parameter is trained using the learning method. Experiment results show that our method can significantly increase the MRR and TRDR performances compared to the baseline methods.

1. **K. Williams, J. Kiseleva, A. C. Crook, I. Zitouni, A. H. Awadallah, and M. Khabsa, “Detecting good abandonment in mobile search,” in Proc. 25th Int. Conf. World Wide Web, 2016, pp. 495–505.**

A key element differentiating traditional systems from systems of systems is governance. While systems are characterized by belonging to a single governing authority, systems within a system of systems are often independently governed or governed by fully empowered entities. Such independence is a necessary condition for the autonomy of each constituent system and for enabling the concept of belonging. At the same time, the capability to be autonomous and the voluntary nature of belonging, enables a constituent system to voluntarily abandon the system of systems it belongs to as well. Yet, research has not addressed so far the implications and odelling of intended abandonment into the operational effectiveness of a system of systems. This paper presents the concept of system abandonment as a philosophical necessity in the definition of systems of systems, it discusses some visions to measure the risk of abandonment, and proposes a way forward to explore mitigation techniques.

1. **O. Kolomiyets and M.-F. Moens, “A survey on question answering technology from an information retrieval perspective,” Inf. Sci., vol. 181, no. 24, pp. 5412 –5434, Dec. 2011.**

The move to Cloud services introduces many new and complex issues related to data security. This video course addresses the threats to data security as they relate to the Cloud, and offers a review of the technologies that work together to create a robust Data Loss Prevention (DLP) architecture. Core concepts such as Encryption and Key Management are discussed, as well as strategies for the development of an enterprise-wide DLP policy.

1. **C. Shah and J. Pomerantz, “Evaluating and predicting answer quality in community QA,” in Proc. 33rd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 2010, pp. 411 –418.**

The flooding of low-quality user generated contents (UGC) in online social network (OSN) has been a threat to web knowledge management systems. Recently several domain-specific systems have been developed addressing this problem, for example, predict correct answer in QA community; recognize reliable comment in products review forums etc. Major drawback of most research efforts is the lack of a general framework applicable to all OSNs. In this study, the authors start by analysing the effects of distinguishing features on UGC quality in different types of OSNs. Extensive statistical analysis leads to the discovery of existence of diverse patterns of human information sharing activity in dissimilar OSNs. This discovery is employed as prior knowledge in the classification framework, which decompose the original highly imbalanced problem into several balanced sub-problems. Ensemble classifiers are adopted in samples from clusters generated by incompact features. Experiments show the proposed framework is both effective and efficient for several OSNs. Contributions of this study are two-fold:

1. model posting activity in different types of OSNs;
2. propose novel classification framework to identify UGC quality.

**CHAPTER-3**

**MODULES**

**3.1 USER INTERFACE:-**

It is a point of human-computer interaction and communication on a device, webpage and app etc.

It helps in defining the inputs from the user .and also helps interfacing functionality at the output terminal of the system. Thus it plays a vital role in input and output terminals of a system.

It is important as it enables user effectively controls the computer or device they are interacting with.

**3.2 API INTERFACE:-**

It is abbreviated as application programming interface.

The system gets inputs from the user about what are the requirements of the user and browses the data and finally gives the required information.

APIs is a set of commands, functions, protocols and objects, that programmers used to create software or interact with the external systems.

Web application uses APIs to connect user facing front ends with all important back end data**.**

**3.3 DATA FLOW DIAGRAM**

**Fig.3**

**CHAPTER-4**

**PYTHON OVERVIEW & ENVIRONMENT**

**4.1.1 PYTHON OVERVIEW:**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner’s Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**4.1.2 HISTORY OF PYTHON:**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**4.1.3 PYTHON FEATURES:**

Python’s features include:

* **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain:** Python’s source code is fairly easy-to-maintain.
* **A broad standard library:** Python’s bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases:** Python provides interfaces to all major commercial databases.
* **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable:** Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

* IT supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**4.2 PYTHON ENVIRONMENT:**

Python is available on a wide variety of platforms including Linux and Mac OS X. Let’s understand how to set up our Python environment.

**4.2.1 SOFTWARE REQUIREMENTS:**

* Python
* Anaconda Navigator
* Python built-in modules
* Numpy
* Pandas
* Matplotlib
* Sklearn
* Seaborm

**CHAPTER-5**

**DOMAIN SPECIFICATION**

**5.1 MACHINE LEARNING:**

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive odelling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user’s historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

**Machine Learning vs. Traditional Programming**

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

**Machine Learning**

**COMPUTER**

**DATA RULES**

**OUTPUT**

**Fig.4**

## How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if it’s feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data that is used to tackle a problem.

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.

[](https://www.guru99.com/images/tensorflow/082918_1102_WhatisMachi3.png)

**Fig.5**

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

#### Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

[](https://www.guru99.com/images/tensorflow/082918_1102_WhatisMachi4.png)

**Fig.6**

The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

## Machine learning Algorithms and where they are used?

[](https://www.guru99.com/images/tensorflow/082918_1102_WhatisMachi5.png)

**Fig.7**

Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

#### Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

1. Classification task
2. Regression task

**Classification Task:**

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

#### **Regression Task:**

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

|  |  |  |
| --- | --- | --- |
| **Algorithm Name** | **Description** | **Type** |
| **Linear regression** | Finds a way to correlate each feature to the output to help predict future values. | Regression |
| **Logistic regression** | Extension of linear regression that’s used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colours) | Classification |
| **Decision tree** | Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a colours, each possible colours becomes a new branch) until a final decision output is made | Regression Classification |
| **Naive Bayes** | The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event. | Regression Classification |
| **Support vector machine** | Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver. | Regression (not very common) Classification |
| **Random forest** | The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the ‘majority vote’ method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction. | Regression Classification |
| **AdaBoost** | Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome | Regression Classification |
| **Gradient-boosting trees** | Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it. | Regression Classification |

#### Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you.

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Description** | **Type** |
| **K-means clustering** | Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans) | Clustering |
| **Gaussian mixture model** | A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters) | Clustering |
| **Hierarchical clustering** | Splits clusters along a hierarchical tree to form a classification system.  Can be used for Cluster loyalty-card customer | Clustering |
| **Recommender system** | Help to define the relevant data for making a recommendation. | Clustering |
| **PCA/T-SNE** | Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances. | Dimension Reduction |

**5.1.1 APPLICATION OF MACHINE LEARNING:**

**Augmentation**:

* Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

**Automation**:

* Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

**Finance Industry**

* Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

**Government organization**

* The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

**Healthcare industry**

* Healthcare was one of the first industry to use machine learning with image detection.

**Marketing**

* Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

**Example of application of Machine Learning in Supply Chain**

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM’s Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

**Example of Machine Learning Google Car**

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What’s impressive is that the car is processing almost a gigabyte a second of data.

**5.2 DEEP LEARNING:**

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

**5.3 REINFORCEMENT LEARNING:**

Reinforcement learningis a subfield of machine learning in which systems are trained by receiving virtual “rewards” or “punishments,” essentially learning by trial and error. Google’s DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

* Q-learning
* Deep Q network
* State-Action-Reward-State-Action (SARSA)
* Deep Deterministic Policy Gradient (DDPG)

**Applications/ Examples of deep learning applications**

* **AI in Finance:**

The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants.

Underwrite is a Fintech company providing an AI solution for credit maker’s company. Underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

* **AI in HR:**

Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour’s ability to have their retail stores fully staffed, ramped and ready to operate.

At that time, Under Armour had all of the ‘must have’ HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren’t useful enough. Under armour choose HireVue, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

* **AI in Marketing:**

I is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers.

For example, deep-learning analysis of audio allows systems to assess a customer’s emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue.

Apart from the three examples above, AI is widely used in other sectors/industries.

**CHAPTER-6**

**ARTIFICIAL INTELLIGENCE & TENSOR FLOW**

**6.1 ARTIFICIAL INTELLIGENCE:**

ML

**Machine Learning**

**Deep Learning**

**Fig.8**

## Difference between Machine Learning and Deep Learning:

|  |  |  |
| --- | --- | --- |
|  | **Machine Learning** | **Deep Learning** |
| **Data Dependencies** | Excellent performances on a small/medium dataset | Excellent performance on a big dataset |
| **Hardware dependencies** | Work on a low-end machine. | Requires powerful machine, preferably with GPU: DL performs a significant amount of matrix multiplication |
| **Feature engineering** | Need to understand the features that represent the data | No need to understand the best feature that represents the data |
| **Execution time** | From few minutes to hours | Up to weeks. Neural Network needs to compute a significant number of weights |
| **Interpretability** | Some algorithms are easy to interpret (logistic, decision tree), some are almost impossible (SVM, XGBoost) | Difficult to impossible |

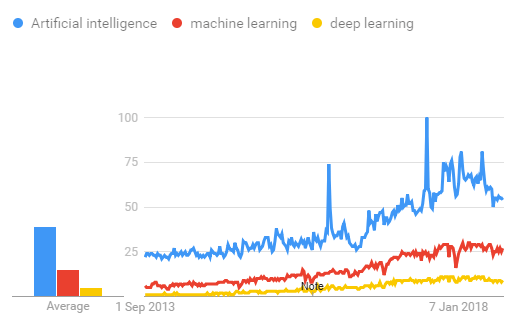
## When to use ML or DL?

In the table below, we summarize the difference between machine learning and deep learning.

|  |  |  |
| --- | --- | --- |
|  | **Machine learning** | **Deep learning** |
| **Training dataset** | Small | Large |
| **Choose features** | Yes | No |
| **Number of algorithms** | Many | Few |
| **Training time** | Short | Long |

With machine learning, you need fewer data to train the algorithm than deep learning. Deep learning requires an extensive and diverse set of data to identify the underlying structure. Besides, machine learning provides a faster-trained model. Most advanced deep learning architecture can take days to a week to train. The advantage of deep learning over machine learning is it is highly accurate. You do not need to understand what features are the best representation of the data; the neural network learned how to select critical features. In machine learning, you need to choose for yourself what features to include in the model.

**Average graph of AI, ML, DL.**

[](https://www.guru99.com/images/tensorflow/083018_0454_MachineLear6.png)

**Fig.9**

## 6.2 TENSORFLOW:

The most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword and the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

* Researchers
* Data scientists
* Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so TensorFlow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

In this tutorial, you will learn

**6.2.1 TENSORFLOW ARCHITECTURE:**

TensorFlow architecture works in three parts:

* Pre-processing the data
* Build the model
* Train and estimate the model

It is called TensorFlow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

**Where can TensorFlow run?**

TensorFlow can hardware, and software requirements can be classified into:

Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.

Run Phase or Inference Phase: Once training is done TensorFlow can be run on many different platforms. You can run it on

* Desktop running Windows, macOS or Linux
* Cloud as a web service
* Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python.

Finally, a significant feature of TensorFlow is the TensorBoard. The TensorBoard enables to monitor graphically and visually what TensorFlow is doing.

**List of Prominent Algorithms supported by TensorFlow**

* Linear regression: tf.estimator.LinearRegressor
* Classification: tf.estimator.LinearClassifier
* Deep learning classification: tf.estimator.DNNClassifier
* Deep learning wipe and deep: tf.estimate.DNNLinearCombinedClassifier
* Booster tree regression: tf.estimator.BoostedTreesRegressor
* Boosted tree classification: tf.estimator.BoostedTreesClassifier

**CHAPTER-7**

**ANACONDA NAVIGATOR & NUMPY**

**7.1 ANACONDA NAVIGATOR:**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

## Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

## WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR?

The following applications are available by default in Navigator:

* Jupyter Lab
* Jupyter Notebook
* QTConsole
* Spyder
* VSCode
* Glueviz
* Orange 3 App
* Rodeo
* RStudio

Advanced conda users can also build your own Navigator applications

## How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

## What’s new in 1.9?

* Add support for **Offline Mode** for all environment related actions.
* Add support for custom configuration of main windows links.
* Numerous bug fixes and performance enhancements.

**7.2 NUMPY:**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

* NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an ndarray will create a new array and delete the original.
* The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
* NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python’s built-in sequences.
* A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today’s scientific/mathematical Python-based software, just knowing how to use Python’s built-in sequence types is insufficient - one also needs to know how to use NumPy arrays. The points about sequence size and speed are particularly important in scientific computing. As a simple example, consider the case of multiplying each element in a 1-D sequence with the corresponding element in another sequence of the same length. If the data are stored in two Python lists, a and b, we could iterate over each element:

The Numeric Python extensions (NumPy henceforth) is a set of extensions to the Python programming language which allows Python programmers to efficiently manipulate large sets of objects organized in grid-like fashion. These sets of objects are called arrays, and they can have any number of dimensions: one dimensional array’s are similar to standard Python sequences, two-dimensional arrays are similar to matrices from linear algebra. Note that one-dimensional arrays are also different from any other Python sequence, and that two-dimensional matrices are also different from the matrices of linear algebra, in ways which we will mention later in this text. Why are these extensions needed? The core reason is a very prosaic one, and that is that manipulating a set of a million numbers in Python with the standard data structures such as lists, tuples or classes is much too slow and uses too much space. Anything which we can do in NumPy we can do in standard Python – we just may not be alive to see the program finish. A more subtle reason for these extensions however is that the kinds of operations that programmers typically want to do on arrays, while sometimes very complex, can often be decomposed into a set of fairly standard operations. This decomposition has been developed similarly in many array languages. In some ways, NumPy is simply the application of this experience to the Python language – thus many of the operations described in NumPy work the way they do because experience has shown that way to be a good one, in a variety of contexts. The languages which were used to guide the development of NumPy include the infamous APL family of languages, Basis, MATLAB, FORTRAN, S and S+, and others. This heritage will be obvious to users of NumPy who already have experience with these other languages. This tutorial, however, does not assume any such background, and all that is expected of the reader is a reasonable working knowledge of the standard Python language. This document is the “official” documentation for NumPy. It is both a tutorial and the most authoritative source of information about NumPy with the exception of the source code. The tutorial material will walk you through a set of manipulations of simple, small, arrays of numbers, as well as image files.

This choice was made because:

* + - * A concrete data set makes explaining the behaviour of some functions much easier to motivate than simply talking about abstract operations on abstract data sets.
      * Every reader will at least an intuition as to the meaning of the data and organization of image files, and
      * The result of various manipulations can be displayed simply since the data set has a natural graphical representation. All users of NumPy, whether interested in image processing or not, are encouraged to follow the tutorial with a working NumPy installation at their side, testing the examples, and, more importantly, transferring the understanding gained by working on images to their specific domain. The best way to learn is by doing – the aim of this tutorial is to guide you along this “doing”.

**CHAPTER-8**

**CODE & RESULT & CONCLUSION**

**8.1 CODE:**

# -\*- coding: utf-8 -\*-

# Form implementation generated from reading ui file 'BeaverBot.ui'

# Created by: PyQt5 UI code generator 5.9.2

# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets

from APIRequest import APIRequest

import os

from subprocess import call

from gtts import gTTS

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(915, 721)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

#layout

self.gridLayout = QtWidgets.QGridLayout(self.centralwidget)

self.gridLayout.setObjectName("gridLayout")

spacerItem = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem, 2, 9, 1, 1)

spacerItem1 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem1, 2, 11, 1, 1)

spacerItem2 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem2, 2, 3, 1, 1)

#beaverLabel

self.label = QtWidgets.QLabel(self.centralwidget)

self.label.setObjectName("label")

self.gridLayout.addWidget(self.label, 0, 7, 1, 1)

#responseField

self.textEdit = QtWidgets.QTextEdit(self.centralwidget)

self.textEdit.setObjectName("textEdit")

# Sets response box to read only

self.textEdit.setReadOnly(True)

self.gridLayout.addWidget(self.textEdit, 2, 7, 1, 1)

spacerItem3 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem3, 2, 5, 1, 1)

spacerItem4 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem4, 2, 12, 1, 1)

#textEditField

self.lineEdit = QtWidgets.QLineEdit(self.centralwidget)

self.lineEdit.setMinimumSize(QtCore.QSize(0, 30))

self.lineEdit.setText("")

self.lineEdit.setClearButtonEnabled(True)

self.lineEdit.setObjectName("lineEdit")

self.gridLayout.addWidget(self.lineEdit, 5, 7, 1, 1)

spacerItem5 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)

self.gridLayout.addItem(spacerItem5, 4, 7, 1, 1)

spacerItem6 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem6, 2, 0, 1, 1)

spacerItem7 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem7, 2, 4, 1, 1)

spacerItem8 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem8, 2, 14, 1, 1)

spacerItem9 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem9, 2, 13, 1, 1)

spacerItem10 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem10, 2, 6, 1, 1)

spacerItem11 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem11, 2, 8, 1, 1)

spacerItem12 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem12, 2, 1, 1, 1)

#button

self.pushButton = QtWidgets.QPushButton(self.centralwidget)

self.pushButton.setMaximumSize(QtCore.QSize(120, 32))

self.pushButton.setObjectName("pushButton")

# Connects button's clicked signal to function openClick

self.pushButton.clicked.connect(self.beaverPowerClicked)

self.gridLayout.addWidget(self.pushButton, 6, 7, 1, 1, QtCore.Qt.AlignHCenter)

spacerItem13 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem13, 2, 10, 1, 1)

spacerItem14 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)

self.gridLayout.addItem(spacerItem14, 3, 7, 1, 1)

spacerItem15 = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)

self.gridLayout.addItem(spacerItem15, 2, 2, 1, 1)

spacerItem16 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)

self.gridLayout.addItem(spacerItem16, 8, 7, 1, 1)

spacerItem17 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)

self.gridLayout.addItem(spacerItem17, 9, 7, 1, 1)

spacerItem18 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)

self.gridLayout.addItem(spacerItem18, 7, 7, 1, 1)

spacerItem19 = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)

self.gridLayout.addItem(spacerItem19, 1, 7, 1, 1)

MainWindow.setCentralWidget(self.centralwidget)

self.menubar = QtWidgets.QMenuBar(MainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 915, 26))

self.menubar.setObjectName("menubar")

MainWindow.setMenuBar(self.menubar)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "Beaver Bot v 1.0"))

self.label.setToolTip(\_translate("MainWindow", "<html><head/><body><p>Hello! I\'m Beaver Bot!</p></body></html>"))

self.label.setText(\_translate("MainWindow", "<html><head/><body><p align=\"center\"><img src=\":/BeaverImage/beaver.png\"/></p></body></html>"))

self.lineEdit.setToolTip(\_translate("MainWindow", "<html><head/><body><p>I am Beaver Bot. How may I help?</p></body></html>"))

self.lineEdit.setPlaceholderText(\_translate("MainWindow", "I am Beaver Bot. How may I help?"))

self.pushButton.setToolTip(\_translate("MainWindow", "<html><head/><body><p>Beaver Bot to the Rescue!</p></body></html>"))

self.pushButton.setText(\_translate("MainWindow", "Beaver Power!"))

hello = "hello my name is Beaver Bot, how can I help?"

self.beaverSpeech(hello)

def updateResponseField(self, text):

self.textEdit.setText(text)

def beaverPowerClicked(self):

input = self.lineEdit.text()

# When button is clicked, calls getResponse function from APIRequest

api = APIRequest()

response = api.getResponse(input)

self.updateResponseField(response)

#update lineEdit

self.lineEdit.clear()

def beaverSpeech(self, text):

tts = gTTS(text=text, lang='en')

tts.save("voice.mp3")

print("should be saying ")

if sys.platform == 'linux2':

call(["xdg-open", "voice.mp3"])

elif sys.platform == 'darwin':

call(["afplay", "voice.mp3"])

print("in mac")

#tts = gTTS(text= text, lang='en')

#tts.save("beaver.mp3")

#os.system("afplay beaver.mp3 &")

import BeaverSource\_rc

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

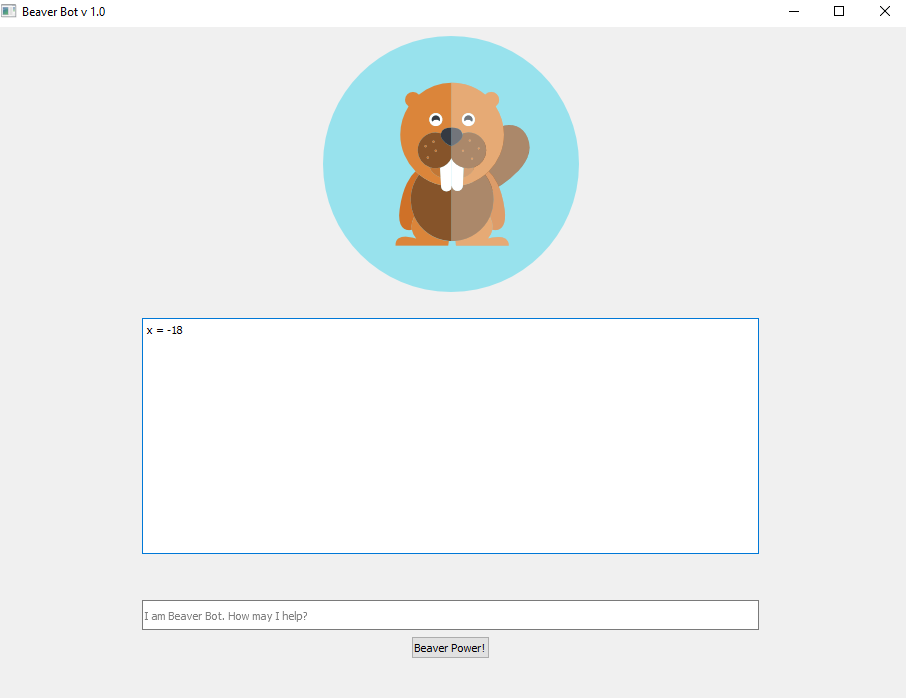
ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec\_())

**8.2 RESULT**:



**Fig.10: The output for the given input to the beaver bot is shown**

**8.3 CONCLUSION:**

The goal of the project is to reduce man-power and to respond to user query at faster rate. Early days, the user’s use to send a query mail to the particular site administrator and it would take few days for the site administrator to reply to the mails. Chatbots can overcome this delay, chatbot satisfies the user request or query immediately with relevant responses. These days many websites of banks, educational institutions, business sectors have developed their chatbots to satisfy user request in a faster time. Chatbots are user-friendly artificial machines. This paper presents the overall development and working of the chatbot.

**REFERENCES:**

[1] K. Cho, B. Van Merri¨enboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y, Bengio, ”Learning phrase representations using RNN encoder-decoder for statistical machine translation,” arXiv:1406.1078, 2014.

[2] C. Lim, H. Choi, ”Efﬁcient Temporal Information Extraction from Korean Documents,” in the 2nd International Workshop on Spatial/Temporal Information Extraction from Unstructured Texts(WSTIE 2017), Daejeon, Korea, May 29, 2017 [The 18th IEEE International Conference on Mobile Data Management (MDM 2017)]

[3] K. Oh, D. Lee, S. Kown, S. Hong, H. Choi, ”Machine learning Based Boundary Detection and Dictionary Expansion for Named Entity Recognition in Dialogues,” in Korea Computer Congress 2017 (KCC 2017), Jeju, Korea, Jun. 2017.

[4] K. Oh, D. Lee, H. Choi, J. Heo, ”Morpheme embedding for paraphrase extraction,” in KIISE KCC2016, Jeju, Korea, S5-3, Jun. 2016 [Korea Computer Congress 2016].

[5] K. Oh, D. Lee, C. Park, H. Choi, Y. Jeong, S. Kwon, S. Hong, ”Out-ofDomain Detection Based on Sentence Distance for Dialogue System,” in The 1st International Workshop on Dialog Systems (IWDS 2018), pp. 673-676, Shanghai, China, January 15, 2018. [The 5th International Conference on Big Data and Smart Computing (BigComp2018)