ABSTRACT

Natural language processing (NLP), as a theory motivated computational technique, has extensive applications. Automated test case generation based on path coverage, which is a popular structural testing activity, can automatically reveal logic defects that exist in NLP programs and can save testing consumption. The high complexity behind SQL language and database schemas has made database querying a challenging task to human programmers. In this paper, we present our new natural language based database querying system as an alternative solution, by designing new translation models smoothly fusing deep learning and traditional database parsing techniques. We develop new techniques to enable the augmented neural network to reject queries irrelevant to the contents of the target database and recommend candidate queries reversely transformed into natural language.

TABLE OF CONTENTSS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

MACHINE LEARNING

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future. Data is the core backbone of machine learning algorithms. With the help of the historical data, we are able to create more data by training these machine learning algorithms. For example, Generative Adversarial Networks are an advanced concept of Machine Learning that learns from the historical images through which they are capable of generating more images. This is also applied towards speech and text synthesis. Therefore, Machine Learning has opened up a vast potential for data science applications.

Machine Learning combines computer science, mathematics, and statistics. Statistics is essential for drawing inferences from the data. Mathematics is useful for developing machine learning models and finally, computer science is used for implementing algorithms. However, simply building models is not enough. You must also optimize and tune the model appropriately so that it provides you with accurate results. Optimization techniques involve tuning the hyperparameters to reach an optimum result. The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways. Machine Learning has revolutionized industries like medicine, healthcare, manufacturing, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry.

Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms,with an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it.

These machine learning algorithms use the patterns contained in the training data to perform classification and future predictions. Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions. Based on the final accuracy, one can optimize their models using various standardized approaches. In this way, Machine Learning model learns to adapt to new examples and produce better results.

SUPERVISED LEARNING

In the majority of supervised learning applications, the ultimate goal is to develop a finely tuned predictor function h(x) (sometimes called the “hypothesis”). “Learning” consists of using sophisticated mathematical algorithms to optimize this function so that, given input data x about a certain domain (say, square footage of a house), it will accurately predict some interesting value h(x) (say, market price for said house).

$$h(x_1, x_2, x_3, x_4) = \theta_0 + \theta_1 x_1 + \theta_2 x_3^2 + \theta_3 x_3 x_4 + \theta_4 x_1^3 x_2^2 + \theta_5 x_2 x_3^4 x_4^2$$

This function takes input in four dimensions and has a variety of polynomial terms. Deriving a normal equation for this function is a significant challenge. Many modern machine learning problems take thousands or even millions of dimensions of data to build predictions using hundreds of coefficients. Predicting how an organism’s genome will be expressed, or what the climate will be like in fifty years, are examples of such complex problems.

Under supervised ML, two major subcategories are:

- Regression machine learning systems: Systems where the value being predicted falls somewhere on a continuous spectrum. These systems help us with questions of “How much?” or “How many?”.
- Classification machine learning systems: Systems where we seek a yes-or-no prediction, such as “Is this tumer cancerous?”, “Does this cookie meet our quality standards?”, and so on.

In practice, x almost always represents multiple data points. So, for example, a housing price predictor might take not only square-footage (x1) but also number of bedrooms (x2), number of bathrooms (x3), number of floors (x4), year built (x5), zip code (x6), and so forth. Determining which inputs to use is an important part of ML design. However, for the sake of explanation, it is easiest to assume a single input value is used.

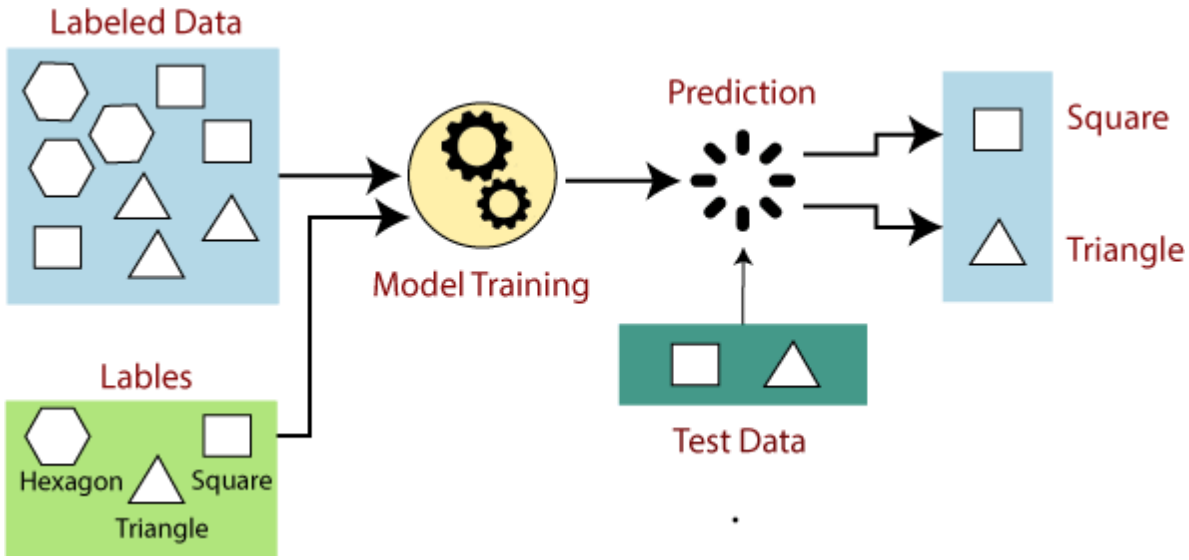

Fig 1.1 Supervised prediction model

Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough

knowledge so that the model can accurately predict the output.

- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

1. Linear Regression
2. Regression Trees
3. Non-Linear Regression
4. Bayesian Linear Regression
5. Polynomial Regression

Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Spam Filtering,
- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

EXISTING SYSTEM

The NEURAL ENQUIRER — a neural network architecture for answering natural language (NL) questions based on a knowledge base (KB) table. Unlike existing work

on end-to-end training of semantic parsers [Pasupat and Liang, 2015; Neelakantan et al., 2015], NEURAL ENQUIRER is fully “Neuralised”: it finds distributed representations of queries and KB tables, and executes queries through a series of neural network components called “executors”. Executors model query operations and compute intermediate execution results in the form of table annotations at different levels. NEURAL ENQUIRER can be trained with gradient descent, with which the representations of queries and the KB table are jointly optimized with the query execution logic. The training can be done in an end-to-end fashion, and it can also be carried out with stronger guidance, e.g., step-by-step supervision for complex queries. NEURAL ENQUIRER is one step towards building neural network systems that can understand natural language in real-world tasks. As a proof-of-concept, we conduct experiments on a synthetic QA task, and demonstrate that the model can learn to execute reasonably complex NL queries on small-scale KB tables.

Disadvantages

- This takes more time to process.
- The efficiency is less that compare to proposed one.

PROPOSED SYSTEM

NADAQ consists of three major components, including data storage module, model management module and user interface module. Data storage module includes MySQL as the underlying database engine, which extracts meta-data from the tables for translation model training and executes the SQL queries to return search results to human users.

Advantages

- It simplify the process.
- Minimum Processing time.

CHAPTER 2
LITERATURE SURVEY

Title : Driver Distraction Detection Using Semi-Supervised Machine Learning.

Author : Tianchi Liu ; Yan Yang ; Guang-Bin Huang ; Yong Kiang Yeo ; Zhiping Lin

Explanation

Real-time driver distraction detection is the core to many distraction countermeasures and fundamental for constructing a driver-centered driver assistance system. While data-driven methods demonstrate promising detection performance, a particular challenge is how to reduce the considerable cost for collecting labeled data. This paper explored semi-supervised methods for driver distraction detection in real driving conditions to alleviate the cost of labeling training data. Laplacian support vector machine and semi-supervised extreme learning machine were evaluated using eye and head movements to classify two driver states: attentive and cognitively distracted. With the additional unlabeled data, the semi-supervised learning methods improved the detection performance (G-mean) by 0.0245, on average, over all subjects, as compared with the traditional supervised methods. As unlabeled training data can be collected from drivers' naturalistic driving records with little extra resource, semi-supervised methods, which utilize both labeled and unlabeled data, can enhance the efficiency of model development in terms of time and cost.

Title : Detection of Driver Cognitive Distraction: A Comparison Study of Stop-Controlled Intersection and Speed-Limited Highway.

Author : Yuan Liao ; Shengbo Eben Li ; Wenjun Wang ; Ying Wang ; Guofa Li ; Bo Cheng

Explanation

Driver distraction has been identified as one major cause of unsafe driving. The existing studies on cognitive distraction detection mainly focused on high-speed driving situations, but less on low-speed traffic in urban driving. This paper presents a method for the detection of driver cognitive distraction at stop-controlled intersections and compares its feature subsets and classification accuracy with that on a speed-limited

highway. In the simulator study, 27 subjects were recruited to participate. Driver cognitive distraction is induced by the clock task that taxes visuospatial working memory. The support vector machine (SVM) recursive feature elimination algorithm is used to extract an optimal feature subset out of features constructed from driving performance and eye movement. After feature extraction, the SVM classifier is trained and cross-validated within subjects. On average, the classifier based on the fusion of driving performance and eye movement yields the best correct rate and F-measure (correctrate = 95.8 ± 4.4%; for stop-controlled intersections and correct rate = 93.7 ± 5.0%; for a speed-limited highway) among four types of the SVM model based on different candidate features. The comparisons of extracted optimal feature subsets and the SVM performance between two typical driving scenarios are presented.

Title : Online Driver Distraction Detection Using Long Short-Term Memory.

Author : Martin Wollmer ; Christoph Blaschke ; Thomas Schindl ; Björn Schuller ; Berthold Farber ; Stefan Mayer ; Benjamin Trefflich

Explanation

This paper presents a new computational framework for early detection of driver distractions (map viewing) using brain activity measured by electroencephalographic (EEG) signals. Compared with most studies in the literature, which are mainly focused on the classification of distracted and nondistracted periods, this study proposes a new framework to prospectively predict the start and end of a distraction period, defined by map viewing. The proposed prediction algorithm was tested on a data set of continuous EEG signals recorded from 24 subjects. During the EEG recordings, the subjects were asked to drive from an initial position to a destination using a city map in a simulated driving environment. The overall accuracy values for the prediction of the start and the end of map viewing were 81% and 70%, respectively. The experimental results demonstrated that the proposed algorithm can predict the start and end of map viewing with relatively high accuracy and can be generalized to individual subjects. The outcome of this study has a high potential to improve the design of future intelligent navigation systems. Prediction of the start of map viewing can be used to provide route information based on a driver's needs and consequently avoid map-viewing activities.