Scholar Publishing Group
Machine Learning Theory and Practice
https://doi.org/10.38007/ML.2020.010102
ISSN 2790-0983 Vol. 1, Issue 1: 10-21

# *In-Vehicle Speech Text Classification based on Multiple Machine Learning Algorithms*

**Fang Li***

*Department of Information Engineering, Heilongjiang International University, Harbin 150025, China*

*lifang@hiu.net.cn*

**corresponding author*

***Keywords:*** Machine Learning, In-Car Speech, Text Classification, Equipment Maintenance

***Abstract:*** With the rapid development of modern society, especially the popularity of the Internet and the rapid development of computer technology, people's lifestyles are undergoing fundamental changes. The rapid development of the Internet has led to an explosion of data and how people use this data has become one of the most popular research topics in modern society. In the past, the limitations of computer technology made it difficult to manage large amounts of data effectively, but the rapid development of computer technology has now made it possible to recognise that it is possible to manage and analyse such large amounts of data. The main objective of this paper is to develop a study of in-vehicle speech TC based on a variety of machine learning algorithms. In this paper, after acquiring the features of the text, a classification model is trained and this tagged data is learned by the model to obtain a classifier. Plain Bayesian classification, nearest-neighbour classification and decision trees are introduced and their advantages and disadvantages are analysed. The results of experiments on text from in-vehicle speech devices reveal that text classification (TC) using support vector machines has good results; natural language understanding can be achieved by combining a rule-based approach by first performing classification and information extraction operations on the text; and the feasibility of architectural modifications is demonstrated through functional verification.

## 1. Introduction

In recent years, the number of text documents in digital form has grown dramatically in size [1]. With the rapid development of the field of robotics and natural language processing, research into intelligent speech interaction has also entered a phase of rapid development. It is because speech has a level of convenience that other interaction methods cannot achieve that robots must have superb intelligent speech interaction capabilities if they are to further their intelligent role.

Copyright: © 2020 by the authors. This is an Open Access article distributed under the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (https://creativecommons.org/licenses/by/4.0/).

Importantly, it needs to be able to automatically aggregate and categorise documents based on their content. Machine learning is essentially a technology that allows computers to learn like humans and extract useful insights from large amounts of data [2-3].

In a related study, Vijayarani et al. automatically classified documents stored in a personal computer into relevant categories. In order to select the optimal features, a new algorithm, the Optimisation Technique for Feature Selection (OTFS) algorithm, was proposed [4]. Decisive results show that the proposed algorithm achieves better accuracy in optimising features and content-based classification of text documents.Andrea introduced the Interactive Classification System (ICS), a web-based application that supports manual TC activities [5]. The application uses machine learning to continuously adapt an automatic classification model, which in turn is used to actively support its users' classification suggestions. The 'complete freedom' requirement is met by designing an unobtrusive machine learning model, i.e. the machine learning component of ICS acts as an unobtrusive observer of the user, which never interrupts the user, constantly adapts and updates its model according to the user's behaviour, and is always available to perform automatic classification. muhammad et al. provide benchmark performance for Urdu text document classification [6], providing a publicly available benchmark dataset manually labelled for six classes; investigating the performance impact of traditional machine learning-based approaches to Urdu text document classification by embedding 10 filter-based feature selection algorithms that have been widely used in other languages; also investigating the performance impact of Urdu text document classification by using a bidirectional encoder representation of the Transformer performance impact of migration learning; and the completeness of a hybrid approach combining traditional machine learning-based feature engineering and deep learning-based automatic feature engineering was evaluated. Experimental results show that both the feature selection method, known as the normalised difference measure, and the support vector machine outperform the state-of-the-art on two closed-source benchmark datasets, CLE Urdu Digest 1000k and CLE Urtu Digest1 Million.

This paper focuses on in-car speech TC based on a variety of machine learning algorithms. Based on the text data, this paper firstly conducts a preliminary sub-analysis on the framework of auxiliary maintenance of in-vehicle equipment, then cites five examples of in-vehicle equipment maintenance texts based on the characteristics of in-vehicle equipment fault maintenance texts, and addresses the problems and challenges in fault TC; then analyses the steps of TC, the analysis of intention recognition techniques, and the detailed analysis of feature extraction steps; finally conducts an experimental analysis of in-vehicle equipment maintenance texts and in-vehicle speech texts. Finally, the experimental analysis of in-vehicle equipment maintenance text and in-vehicle voice text is carried out.

## 2. Design Research

### 2.1. Auxiliary Maintenance Framework for In-Vehicle Equipment Based on Text Data

When a fault occurs in on-board equipment, maintenance personnel who are professionally trained and familiar with the working principles of on-board equipment observe the fault phenomena and data through their fault diagnosis efficiency, which directly depends on the technical level of the maintenance personnel and their understanding of the equipment. For inexperienced on-board equipment maintenance personnel, the problem of long diagnosis time and low diagnosis efficiency may occur in the fault diagnosis process [7-8]. In the event of a fault in the on-board equipment, the on-board maintenance personnel will use natural language to record the state and phenomenon of the on-board equipment when it fails [9-10]. In this paper, through the analysis of these fault maintenance texts, it is found that these fault maintenance texts imply effective fault category information, as shown in Figure 1, the on-board fault records contain information related to the composition structure of the on-board equipment, the state of each composition structure, and this information can effectively reflect the cause of failure of the column-controlled on-board equipment, i.e. the fault category [11-12]. By representing the fault record text, feature extraction, and mining the pattern relationship between the fault record and its corresponding fault category, a fault record text classifier is obtained, which can predict the fault type for new fault records. As an auxiliary maintenance method, the fault record TC system constructed in this paper aims to assist in on-board equipment fault diagnosis and improve the efficiency of fault diagnosis [13-14]. With the accumulation of in-vehicle fault records, the fault information contained in fault text data becomes richer and richer, and fault diagnosis based on fault data reflects the relationship between system structure principles and fault categories from another perspective, and the connection between fault phenomena and fault causes (categories) is obtained from the data level. As more fault data becomes available, the more significant the assistance that an on-board maintenance system based on maintenance texts can provide for fault diagnosis [15-16].

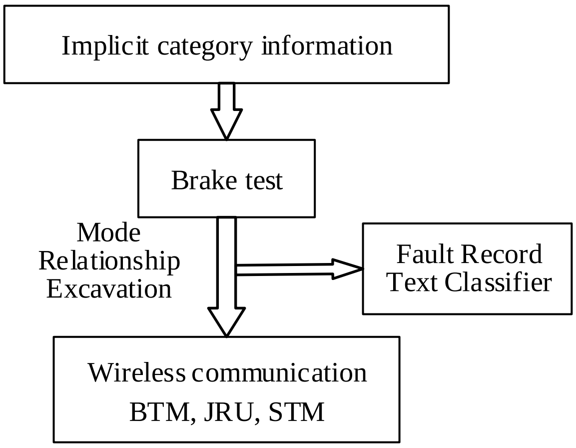


*Figure 1. Text classification and on-board device maintenance assistance*

## 2.2. Text Analysis

Based on the analysis of the characteristics of on-board device maintenance texts, five examples of on-board device maintenance texts were listed to address the problems and challenges in fault TC [17-18]. The details are shown in Table 1

*Table 1. Examples of maintenance texts for on-board equipment*

| Marking | Description of the on-board fault phenomenon | Fault Type |
|---|---|---|
| 1 | Wireless interruption timeout, train downgraded to CTCS-2 level of travel | Wireless communication faults |
| 2 | ATP cannot connect to RBC | Wireless communication faults |
| 3 | Brake test failed, system restart normal, on time | Train Interface |
| 4 | Section dispatch notification: G1025 reported transponder link error at 11:10 | BTM related |
| 5 | G1089 failed to activate the driver's cab several times at Wuhan Station | Train interface |

Combining the above five examples of in-vehicle equipment maintenance texts, the characteristics of in-vehicle equipment maintenance texts are analysed as follows.

1) The text is short and contains few units of information. The text is short in length and contains few word features, as many as a few dozens or a few, while the document set corresponds to a lexicon of up to several thousand dimensions. When using the bag-of-words model representation, the bag-of-words model representation of the text exhibits high and sparse characteristics, which is not conducive to the construction of subsequent classifiers [19-20].

2) There is little overlap of word feature units on the bag-of-words model. Due to the short length of the text, which contains few word units, and the use of natural language by on-board maintenance personnel to describe fault phenomena, there are multiple fault phenomena for the same fault type and multiple ways to describe the same fault phenomena. The complex mapping relationship between fault types and fault phenomena due to the complexity of the system and operating environment, as well as the unstructured text description approach result in two semantically similar texts with little or no overlap of word feature units in the bag-of-words model representation [21-22]. As in the above-mentioned l and 2 maintenance texts, from the semantic point of view, it is easy to determine that these are two fault phenomena caused by wireless communication faults, but these two documents do not have any overlapping word feature units on the bag-of-words model, i.e. the similarity of the bag-of-words model representation vectors of the two documents is 0, which is obviously unreasonable.

3) Uneven distribution of text categories. The number of texts in different categories varies greatly, which poses a challenge to the classification task.

## 3. Experimental Study

### 3.1. TC

After pre-processing the corpus with word separation, feature selection and feature weight calculation, the text needs to be classified for the purpose of intent recognition, i.e. to identify whether the user input sentence wants path planning, or wants a song to be played, etc.

LibSVM is an open source support vector machine library that provides automated tuning tools that make it easy to do classification or regression on data. The library is based on the C++ language and also provides a JNI-based Java API, which is the most used library for SVMs in China today.

The steps to implement TC based on LibSVM are as follows:

1) Organize the results after corpus feature extraction and weight calculation into the format required by LibSVM, as follows:

Labell 1: valuel 2:value2⋯.1abel21: valuel 2:value2⋯.

Where labell and label2 represent the category of the classification, specifically for this thesis

1 represents "route planning";

2 represents "real-time traffic search";

3 stands for "Peripheral search";

4 represents "music playback";

5 for "Listen to radio";

6 for "Make a call";

7 stands for "Other".

The "other" class is provided to avoid misclassification. Since SVM always provides the highest probability result in the end, when the user says a sentence that does not belong to any of the valid categories, an "other" class is provided to identify cases that the semantic recognition engine cannot recognize. For example, if the user says "Have you eaten yet", the classification result is unpredictable if no "other" class is provided, but when the "other" class is provided, it can be classified as "other", at which point it can be prompted, "Sorry, I didn't understand, please say it again, for example, you can ask how to get to Tiananmen Square".

The sequence of key-value pairs (1: valuel 2: value2, ...) after the label represents the feature word number and the corresponding weight (the TF-IDF value calculated in the previous section).

2) For text training, LibSVM provides an automated training script (easy.PY) that automates operations from normalisation to parameter selection. The script finds the optimal parameters by cross-checking.

3) Feature normalisation of the test feature vector

4) Perform prediction on the test text.

5) Call libsvm in the java project to perform the prediction. Firstly, we need to include the libsvmjar, package, and use this package to call the prediction function provided by libsvm. Secondly, we need to load the model file and normalization rule file trained using the corpus, and finally call the interface provided in the phase libsvmjar package to implement the classification of user input sentences.

### 3.2. Intent Recognition Techniques

In intelligent speech interaction systems, when the user issues a dialogue command, the process of the machine understanding and giving feedback is called intent recognition. It is also a TC process that selects the most similar categories among known texts to determine the current speaker's intention. The main components include the following.

1) Pre-processing: Pre-processing the text before extracting features can attenuate the interference of noise. Firstly, it needs to be word-separated; secondly, it removes words that are not useful for expressing the intention of the text, i.e. deactivated words, which can affect the processing speed of the text.

2) Text vectorisation: vectorising text allows it to be directly involved in mathematical operations.

3) Text feature extraction: Once the text has been vectorised, useful information is then extracted from it. At this point a weighting factor is introduced to evaluate the importance of each word. Assuming that all words are equally useful and then all words have a weight of 1. However, each word has a different level of importance to the sentence, so it is necessary to differentiate the level of importance of words. The following are some common algorithms.

(1) Word frequency (TF): the number of times a word appears in a document is counted and compared to a set threshold to configure the weights accordingly. This method is simple, but some low word frequencies may be subject words, thus missing important information.

(2) Word Frequency-Inverse Document (TF-IDF): The importance of a word is measured in terms of both word frequency and inverse document frequency. The higher the word frequency and the lower the inverse document frequency, the more important the word is to the whole article and the more representative it is of the whole article, and the TF is defined by equation (1).

$$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

Where sub ni, j is the number of occurrences and Σ is the sum of the counts; TF alone does not provide a more complete description of the relevance of a document, as there will be words that recur in order to link statements. It is in this context that the inverse text frequency (IDF) was created. The formula is shown in equation (2).

$$idf_{ij} = \log \frac{|D|}{|\{j : t_i \in d_j\}|} \tag{2}$$

Where D is the total number of documents. j is the number of documents. The high-weighted TF-IDF is calculated as follows.

$$TF - IDF = TF * IDF \tag{3}$$

(3) Mutual information: It is used to measure the degree of association between two pieces of information. The principle of mutual information is shown in equation (4).

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log(\frac{p(x,y)}{p(x)p(y)}) \tag{4}$$

In the case of continuous random variables, the summation is replaced by a dual definite integral:

$$I(X;Y) = \int_Y \int_X p(x,y) \log(\frac{p(x,y)}{p(x)p(y)}) dxdy \tag{5}$$

This formula is known as the mutual information of two discrete random variables, X and Y.

(4) TC algorithms: After acquiring the features of a text, a classification model is trained and the model learns that tagged data to obtain a classifier. The following is a description of several common algorithms and an analysis of their advantages and disadvantages.

(1) Plain Bayesian classification: The classification to which a location data belongs is mainly determined by the probability of the known data. The posterior probability is obtained as shown in the following equation.

$$P(B \mid A) = \frac{P(B \mid A) P(B)}{P(A)} \tag{6}$$

Where P(A | B) is the known prior probability, and then the posterior probability P( B | A) is found using the Bayesian formula. Plain Bayesian classification can be predicted in real time and is effective in making classification predictions, with good performance for high-dimensional data. However, if the training data does not represent the whole well because the features are not fully independent in most cases, the classification effect is not satisfactory in this case, and smoothing techniques are needed to solve this problem.

(2) Nearest Neighbour Classification: or K-Nearest Neighbour Classification, is to find the K data that are most similar to the sample, and to which category the majority of these K data belong.

Nearest-neighbour classification does not require assumptions about the data and can solve multi-class problems, but it is no longer applicable when the data set is large and slow and inefficient.

(3) Decision tree: As the name suggests it is a tree structure. It goes down to the leaf nodes, at which point each child node corresponds to a value taken for that feature, and finally the category stored in the leaf node is used as the decision result.

The advantage of a decision tree is that it has high classification accuracy, can automatically perform feature selection and can effectively reduce the probability of overfitting the model.

However, it is easy to over-fit and cannot be used for some decisions that cannot be expressed in quantitative terms.

### 3.3. Feature Extraction Steps

The basic principle of the chi-square test is as follows: in order to select the best features, the correlation between each feature and the category is calculated, and the features with the highest correlation are selected as classification features. For example, suppose the relevance of the feature word "navigate to" to the category "path planning" is calculated, and the statistics in the corpus are shown in Table 2:

*Table 2. Example of a chi-square test*

| Feature selection | Part of "path planning | Not "path planning" | Total |
|---|---|---|---|
| Include "Navigate to" | A | B | A+B |
| Does not include "navigate to" | C | D | C+D |
| | A+C | B+D | N |

1) Classification of the corpus means that the corpus is stored in different text files according to categories, and within the text files, each sentence occupies one line (equivalent to a document). For this thesis, the corpus is divided into different categories according to the corresponding operations, such as path planning, surrounding retrieval, real-time traffic query, playing music, listening to the radio, etc. The corpus is stored in the corresponding different text files, and noise text files are added in order to improve the robustness of the classification. In order to avoid the problems caused by the unbalanced corpus, the number of sentences in each text is kept as close as possible. A total of 1000 sentences were included in the original corpus, 900 samples were randomly selected as training samples and the remaining 100 sentences were used as test samples. As the number of tests increased, a total of 8000 sentences were included in the final training sample.

2) Calculate the chi-square test values for each word and wordiness in each classification. The number of sentences for each word and lexeme in each classification file (corresponding to A in the formula); the number of sentences for each word and lexeme in all classification files (corresponding to A+B in the formula); the number of sentences included in each category (corresponding to A+c in the formula); and the total number of sentences in all classified corpus (corresponding to N in the formula) were counted. Finally, the chi-square test values for each word and lexical category in each category were calculated according to the chi-square test calculation formula. Since A, A+B, A+C and N are already known, the number of sentences containing a word but not belonging to a category can be found as the number of sentences corresponding to that word in all categories minus the number of sentences corresponding to that word in that category (i.e. B=A+B.A); the number of sentences not containing a word but belonging to a category is the total number of sentences contained in that category minus the number of sentences corresponding to that word in that category (i.e. C= A+C.A); the number of sentences that neither contain a word nor belong to a given category is the total number of sentences contained in all documents in the category minus the number of sentences that contain a word and belong to a category, contain a word but do not belong to a category, and do not contain a word but belong to a category (i.e. D=N-A-B-C). .

3) Sort all words and lexemes in the document by CHI value, and select the top n words and lexemes with the highest CHI value as TC features. It should be noted that the size of n also affects the classification accuracy. After several adjustments, the current n value for words is 40 and for lexicalities is 14.

## 4. Experiment Analysis

### 4.1. Analysis of in-Vehicle Equipment Maintenance Texts

Table 3 gives the percentage of sample sizes for each category of maintenance text.

*Table 3. Percentage of samples in each category of maintenance text*

| Type of fault | BTM faults | Wireless communication failure | ATPCU faults | Train interface fault | TCR fault | SDU fault | JRU fault |
|---|---|---|---|---|---|---|---|
| Percentage of samples | 12.6% | 45.3% | 22.6% | 7.2% | 4% | 5.2% | 3.2% |

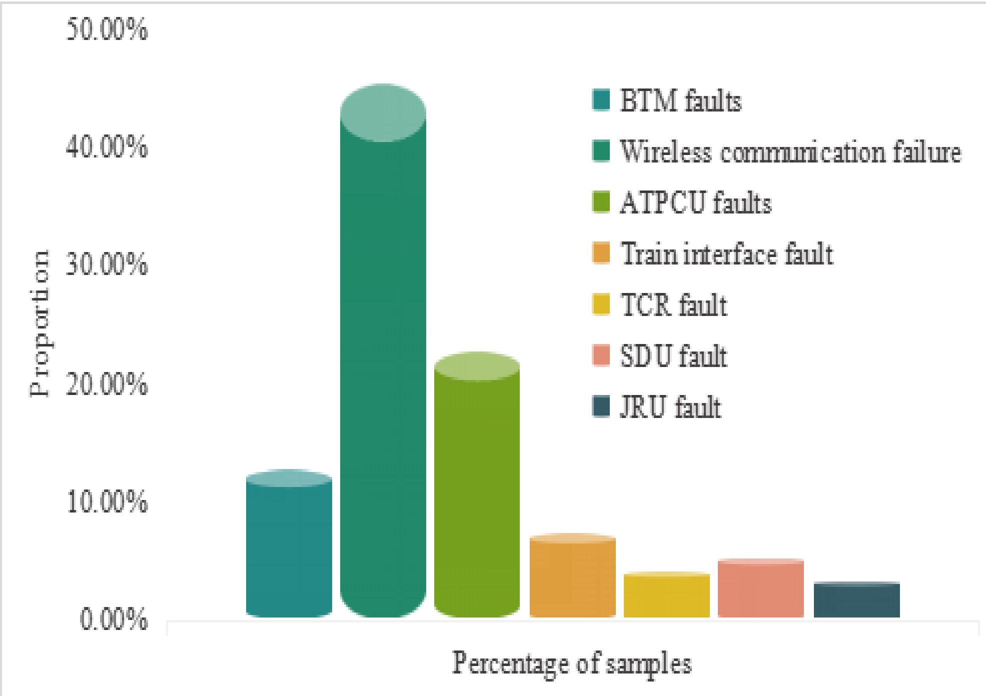


*Figure 2. Analysis of the percentage of samples in each category of maintenance text*

As can be seen in Figure 2, the seven categories of faults have a majority category, such as wireless communication faults, which account for 45.3%, a minority category, such as TCR faults, which account for only 4%, and JRU faults which account for only 3.2%. This severe class distribution imbalance poses a challenge to the construction of the classifier. Traditional classification models (e.g. support vector machines) that aim to minimise misclassification loss will favour the majority class at the expense of the minority class when dealing with class imbalance data, which leads to extremely poor classification of samples on the minority class, resulting in impractical classifiers.

### 4.2. Experimental Analysis of In-Vehicle Speech Text

(1) Speech recognition engine measurement

The integrated Baidu speech recognition engine was tested in Mandarin in a quiet environment, and the results are shown in Table 4.

*Table 4. Speech recognition engine evaluation results*

| | Accuracy | Recall |
|---|---|---|
| Navigation | 0.87 | 0.86 |
| Music | 0.79 | 0.79 |
| Telephone | 0.74 | 0.74 |
| Broadcast | 0.85 | 0.85 |

(2) TC test results

The text was classified into seven categories: route planning, surrounding search, real-time traffic playing music, listening to the radio, making a phone call and other categories. A running database was used to validate them, with a total of 8277 pieces of data. The test results are shown in Table 5.

*Table 5. TC results*

| | Accuracy | Recall |
|---|---|---|
| Route Planning | 0.98 | 0.98 |
| Neighbourhood search | 0.98 | 0.98 |
| Real-time traffic | 0.97 | 0.98 |
| Calling | 0.99 | 0.98 |
| Music playback | 0.97 | 0.97 |
| Listen to the radio | 0.97 | 0.98 |

(3) Information extraction test results

Semantic information extraction for applications such as route planning, peripheral search, playing music, listening to radio, body control and making phone calls was implemented. A running database was applied to validate them and the test results are shown in Table 6.

*Table 6. Semantic recognition results*

| | Accuracy | Recall |
|---|---|---|
| Route Planning | 0.95 | 0.95 |
| Neighbourhood search | 0.94 | 0.93 |
| Real-time traffic | 0.95 | 0.95 |
| Calling | 0.96 | 0.95 |
| Music playback | 0.96 | 0.96 |
| Listen to the radio | 0.93 | 0.93 |

(4) Overall test results

The integrated Baidu speech recognition engine was tested in Mandarin in a quiet environment with a total of 8,277 data. The overall test results are shown in Table 7.

*Table 7. Semantic recognition results*

| | Accuracy | Recall |
|---|---|---|
| Route Planning | 0.77 | 0.77 |
| Neighbourhood search | 0.75 | 0.75 |
| Real-time traffic | 0.78 | 0.75 |
| Calling | 0.70 | 0.69 |
| Music playback | 0.74 | 0.73 |
| Listen to the radio | 0.71 | 0.72 |
| Total | 0.77 | 0.76 |

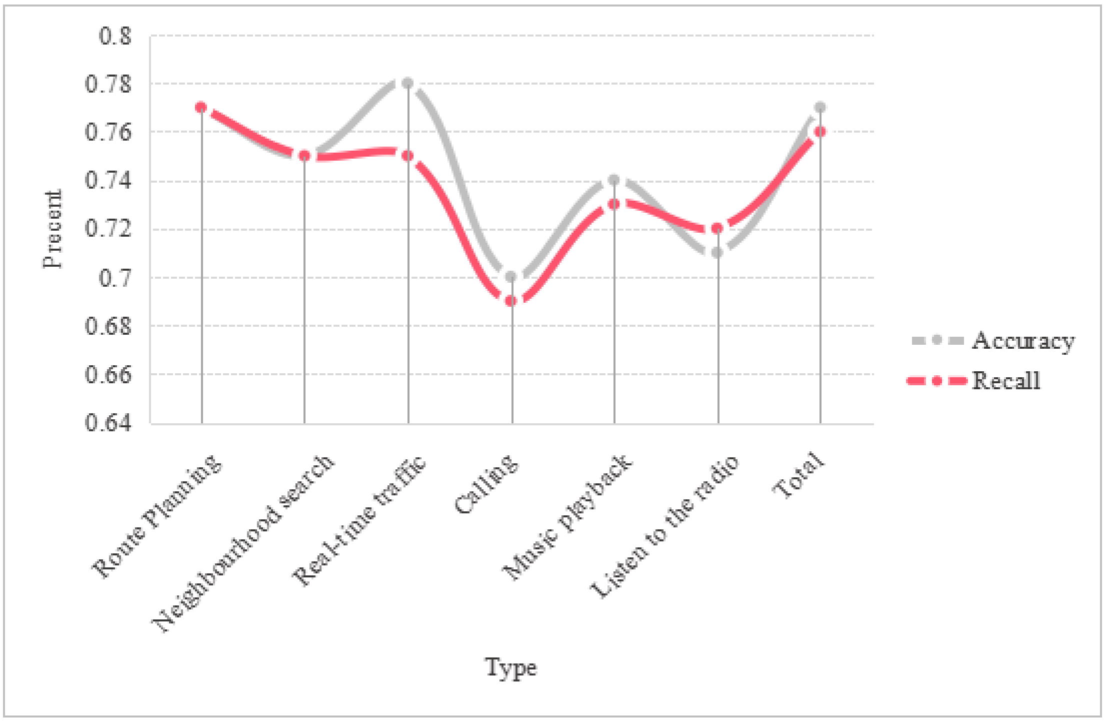


*Figure 3. Semantic recognition results analysis diagram*

From the test results in Figure 3 above, it can be seen that the use of conditional random fields for named entity recognition and semantic recognition has good results; the use of support vector machines for TC has good results; the natural language understanding function can be achieved by combining the rule-based approach with the classification and information extraction operations on the text first; the feasibility of the architectural modification is proved through functional verification.

## 5. Conclusion

With the rapid development of artificial intelligence in recent years, human-computer interaction has become a popular research area in computer science. Human-computer interaction has to solve the problem of accurate recognition and fast response on the one hand, and to continuously improve the level of understanding of the machine on the other. Improving the machine's speech and text processing capabilities is one of the main factors in improving the level of understanding. This paper investigates the application of various machine learning algorithms to in-vehicle speech and TC. The framework of in-vehicle equipment maintenance assistance is analysed and examples of in-vehicle equipment maintenance texts are presented; after pre-processing the corpus with word separation, feature selection and feature weight calculation, the texts are classified for the purpose of intent recognition; then the intent recognition techniques are analysed and the steps of feature extraction are described in detail; during the experimental analysis of in-vehicle speech texts, it is found that the use of conditional random fields In the course of the experimental analysis of in-vehicle speech text, it is found that the use of conditional random fields for named entity recognition and semantic recognition has good results; the use of support vector machines for TC has good results; and the feasibility of the architectural modification is demonstrated by performing functional verification by first classifying the text and information extraction operations.

## Funding

This article is not supported by any foundation.

## Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

## Conflict of Interest

The author states that this article has no conflict of interest.

## References

*[1] Mohammed M J, Mohammed E A, Jarjees M S. Recognition of Multifont English Electronic Prescribing Based on Convolution Neural Network Algorithm. Bio-Algorithms and Med-Systems, 2020, 16(3):182-90. https://doi.org/10.1515/bams-2020-0021*

*[2] Kingsley Okoye, Arturo Arrona-Palacios, Claudia Camacho-Zuñiga, Joaquín Alejandro Guerra Achem, José Escamilla, Samira Hosseini: Towards Teaching Analytics: A Contextual Model for Analysis of Students' Evaluation of Teaching through Text Mining and Machine Learning Classification. Educ. Inf. Technol. 27(3): 3891-3933 (2020).*

*[3] Razaul Karim, Ashef Shahrior, Mohammad Motiur Rahman: Machine learning-based tri-stage classification of Alzheimer's progressive neurodegenerative disease using PCA and mRMR administered textural, orientational, and spatial features. Int. J. Imaging Syst. Technol. 31(4): 2060-2074 (2020). https://doi.org/10.1002/ima.22622*

*[4] R. Janani, S. Vijayarani: Automatic TC Using Machine Learning and Optimization Algorithms. Soft Comput. 25(2): 1129-1145 (2020). https://doi.org/10.1007/s00500-020-05209-8*

*[5] Andrea Esuli: ICS: Total Freedom in Manual TC Supported by Unobtrusive Machine Learning. IEEE Access 10: 64741-64760 (2020).*

*[6] Muhammad Nabeel Asim, Muhammad Usman Ghani Khan, Muhammad Ali Ibrahim, Waqar Mahmood, Andreas Dengel, Sheraz Ahmed: Benchmarking Performance of Machine and Deep Learning-Based Methodologies for Urdu Text Document Classification. Neural Comput. Appl. 33(11): 5437-5469 (2020). https://doi.org/10.1007/s00521-020-05321-8*

*[7] Rasmus Kær Jørgensen, Christian Igel: Machine Learning for Financial Transaction Classification across Companies Using Character-Level Word Embeddings of Text Fields. Intell. Syst. Account. Finance Manag. 28(3): 159-172 (2020). https://doi.org/10.1002/isaf.1500*

*[8] Shyla Raj, B. S. Mahanand, D. S. Vinod: Diffuse Lung Disease Classification based on Texture Features and Weighted Extreme Learning Machine. Multim. Tools Appl. 80(28-29): 35467-35479 (2020) https://doi.org/10.1007/s11042-020-10469-5*

*[9] Joshua Eykens, Raf Guns, Tim C. E. Engels: Fine-grained classification of social science journal articles using textual data: A Comparison of Supervised Machine Learning Approaches. Quant. Sci. Stud. 2(1): 89-110 (2020). https://doi.org/10.1162/qss_a_00106*

*[10] Shalini Ramanathan, Mohan Ramasundaram: Accurate Computation: COVID-19 rRT-PCR Positive Test Dataset Using Stages Classification through Textual Big Data Mining with Machine Learning. J. Supercomput. 77(7): 7074-7088 (2020). https://doi.org/10.1007/s11227-020-03586-3*

*[11] Renu Balyan, Kathryn S. McCarthy, Danielle S. McNamara: Applying Natural Language Processing and Hierarchical Machine Learning Approaches to Text Difficulty Classification. Int. J. Artif. Intell. Educ. 30(3): 337-370 (2020). https://doi.org/10.1007/s40593-020-00201-7*

*[12] Umid Suleymanov, Behnam Kiani Kalejahi, Elkhan Amrahov, Rashid Badirkhanli: TC for Azerbaijani Language Using Machine Learning. Comput. Syst. Sci. Eng. 35(6): 467-475 (2020). https://doi.org/10.32604/csse.2020.35.467*

*[13] Venkatanareshbabu Kuppili, Mainak Biswas, Damodar Reddy Edla, K. J. Ravi Prasad, Jasjit S. Suri: A Mechanics-Based Similarity Measure for TC in Machine Learning Paradigm. IEEE Trans. Emerg. Top. Comput. Intell. 4(2): 180-200 (2018). https://doi.org/10.1109/TETCI.2018.2863728*

*[14] Murat Okkalioglu, Burcu Demirelli Okkalioglu: AFE-MERT: Imbalanced TC With Abstract Feature Extraction. Appl. Intell. 52(9): 10352-10368 (2020).*

*[15] Kingsley Okoye, Arturo Arrona-Palacios, Claudia Camacho-Zuñiga, Joaquín Alejandro Guerra Achem, José Escamilla, Samira Hosseini: Towards Teaching Analytics: A Contextual Model for Analysis of Students' Evaluation of Teaching through Text Mining and Machine Learning Classification. Educ. Inf. Technol. 27(3): 3891-3933 (2020).*

*[16] Muhammad Pervez Akhter, Jiangbin Zheng, Irfan Raza Naqvi, Mohammed Abdelmajeed, Muhammad Fayyaz: Exploring Deep Learning Approaches for Urdu TC in Product Manufacturing. Enterp. Inf. Syst. 16(2): 223-248 (2020). https://doi.org/10.1080/17517575.2020.1755455*

*[17] Santosh Kumar Behera, Rajashree Dash: A Novel Feature Selection Technique for Enhancing Performance of Unbalanced TC Problem. Intell. Decis. Technol. 16(1): 51-69 (2020). https://doi.org/10.3233/IDT-210057*

*[18] Imen Ferjani, Minyar Sassi Hidri, Ali Frihida: SiNoptiC: Swarm Intelligence Optimisation of Convolutional Neural Network Architectures for TC. Int. J. Comput. Appl. Technol. 68(1): 82-100 (2020).*

*[19] Liriam Enamoto, Alan R. Santos, Ricardo Maia, Weigang Li, Geraldo P. Rocha Filho: Multi-label legal TC with BiLSTM and attention. Int. J. Comput. Appl. Technol. 68(4): 369-378 (2020).*

*[20] Mohd Firoz Warsi, Ruqaiya Khanam, Usha Chauhan, Suraj Kamya: Melanoma Classification by 3D Colour-Texture Feature and Neural Network with Improved Computational Complexity Using PCA. Int. J. Medical Eng. Informatics 14(4): 369-378 (2020).*

*[21] Palaiahnakote Shivakumara, Alloy Das, K. S. Raghunandan, Umapada Pal, Michael Blumenstein: New Deep Spatio-Structural Features of Handwritten Text Lines for Document Age Classification. Int. J. Pattern Recognit. Artif. Intell. 36(9): 2252013:1-2252013:34 (2020). https://doi.org/10.1142/S0218001422520139*

*[22] Nasrin Amini, Ahmad Shalbaf: Automatic classification of severity of COVID-19 patients using texture feature and random forest based on computed tomography images. Int. J. Imaging Syst. Technol. 32(1): 102-110 (2020). https://doi.org/10.1002/ima.22679*