International Journal of Intelligent Information Systems
2015; 4(3): 58-70
Published online July 8, 2015 (http://www.sciencepublishinggroup.com/j/ijiis)
doi: 10.11648/j.ijiis.20150403.12
ISSN: 2328-7675 (Print); ISSN: 2328-7683 (Online)

Vehicle Fault Diagnostics Using Text Mining, Vehicle Engineering Structure and Machine Learning

Yi Lu Murphey, Liping Huang, Hao Xing Wang, Yinghao Huang

Department of Electrical and Computer Engineering, University of Michigan-Dearborn, Dearborn, USA

Email address:
yilu@umich.edu (Y. L. Murphey)

To cite this article:
Yi Lu Murphey, Liping Huang, HaoXing Wang, Yinghao Huang. Vehicle Fault Diagnostics Using Text Mining, Vehicle Engineering Structure and Machine Learning. *International Journal of Intelligent Information Systems*. Vol. 4, No. 3, 2015, pp. 58-70.
doi: 10.11648/j.ijiis.20150403.12

Abstract: This paper presents an intelligent vehicle fault diagnostics system, SeaProSel(Search-Prompt-Select). SeaProSel takes a casual description of vehicle problems as input and searches for a diagnostic code that accurately matches the problem description. SeaProSel was developed using automatic text classification and machine learning techniques combined with a prompt-and-select technique based on the vehicle diagnostic engineering structure to provide robust classification of the diagnostic code that accurately matches the problem description. Machine learning algorithms are developed to automatically learn words and terms, and their variations commonly used in verbal descriptions of vehicle problems, and to build a TCW(Term-Code-Weight) matrix that is used for measuring similarity between a document vector and a diagnostic code class vector. When no exactly matched diagnostic code is found based on the direct search using the TCW matrix, the SeaProSel system will search the vehicle fault diagnostic structure for the proper questions to pose to the user in order to obtain more details about the problem. A LSI (Latent Semantic Indexing) model is also presented and analyzed in the paper. The performances of the LSI model and TCW models are presented and discussed. An in-depth study of different term weight functions and their performances are presented. All experiments are conducted on real-world vehicle diagnostic data, and the results show that the proposed SeaProSel system generates accurate results efficiently for vehicle fault diagnostics.

Keywords: Vehicle Fault Diagnostics, Text Data Mining, Machine Learning, Vehicle Diagnostic Engineering Structure, TCW, LSI

1. Introduction

As computers and networks grow more powerful and data storage devices become more plentiful and less costly, the amount of information in digital form is exploded. The majority of such digital data are in text form. Text data mining has many applications including text document search and categorization, website search, customer services, and automatic diagnostic systems [1~4].

In this research we focus on text documents that are casually typed or recorded. Many text mining applications require processing casual text data, which often are in semi-structured or unstructured text, such as clinical document analysis [3, 5], emails, instant messages, free-text of medical records, operational notes, emails, instant messages, etc., and the application of this research is in automotive diagnostic text mining.

In automotive industry there are abundant information available in casual natural language description form that contain valuable vehicle fault diagnostic knowledge, marketing information, consumer evaluation or satisfaction of certain vehicle models, styles, accessories, etc [6]. For example, several thousands of vehicle problems are reported daily to various auto service shops. It is important to find root-cause of a vehicle problem quickly and accurately. In a typical vehicle fault diagnostic process, vehicle problems are first described by customers in casual words and terms. A service advisor records the customer's complaints or description of symptoms verbatim on the repair order, and then searches for a diagnostic code that matches the description. The diagnostic code is used to guide the diagnosis and repairing processes. Due to the complexity of modern vehicles, the number of diagnostic codes can be in hundreds, which makes manual searching of correct diagnostic code difficult and may lead to a lengthy and less accurate diagnosis and repair process, and, possibly, unnecessary part replacements. In order to improve the

accuracy and efficiency of vehicle fault diagnostics, it is important to develop an automated system that can help customers to report problems described in casual words and terms, and technicians to quickly find the correct diagnostic code, i.e., the root cause of the problems. There are several challenges involved in this problem.

The descriptions of vehicle problems provided by customers are often ill-structured. Most of such descriptions do not follow the English grammar, and contain many misspelled words, self-invented acronyms and shorthand descriptions.

The descriptions of a problem by different people vary based on the education and/or cultural background of the customers, and their familiarity of vehicle terminologies and knowledge of automotive engineering. For example, the term "trunk" used United States means the same thing as the term "boot" used in UK. One faulty symptom, for example, "a noise is heard from the engine and the engine runs rough" can be described by customers in various ways, such as engine knocks, hood squeak, engine misses idle, engine lopes, etc. The following are examples of customer descriptions of the same vehicle problem:

Customer 1: "WENT ON A SALES ROAD TEST WITH CUST, VEHICLE WOULDNOT START,"

Customer 2: "CHECK CAR WONT START,"

Customer 3: "CK BATTERY HARD TO START."

High dimensions of terms and document classes. Since there are typos and self-invented acronyms and abbreviations frequently occurring in customer descriptions, the number of distinct terms used in these documents is several times more than formally printed documents. For example, the word "engine" has more than 20 different spellings in customers' descriptions in our data collections. Since the output vector represents all the diagnostic codes used by a car manufacturing company, there can easily be several hundreds of different document classes. These two high dimension issues pose challenges for generating efficient and effective response for a given problem description.

In this paper we present an intelligent vehicle fault diagnostics system, Search-Prompt-Select(SeaProSel), which is developed by combining automatic text categorization techniques with vehicle engineering structure and machine learning to provide effective search functions for the diagnostic code that accurately matches a given problem description. The SeaProSel system uses machine learning techniques to automatically learn words, terms and their variations commonly used in verbal description of vehicle problems from training data, and incorporate a vehicle fault diagnostic engineering structure into the search process to provide accurate diagnostic code that matches the problem description. This paper is organized as follows. Section 2 presents a brief overview of the state-of-art technologies for text mining and document categorization, Section 3 presents the proposed system, SeaProSel, Section 4 presents the experiment results generated from real-world vehicle diagnostic data, and Section 5 concludes the paper.

2. Research in Text Mining and Document Classification

Until the late '80s, the most popular approach to text categorization are based on knowledge engineering [7~9]. These approaches usually consist of a set of predefined rules that are encoded with expert knowledge. Each rule is represented as a disjunctive normal form (DNF formula) followed by a category name. A document is classified under a specific category if it satisfies the DNF formula of the category. This DNF expression is mostly defined by domain experts. If categories are updated or ported to a different domain, domain experts need to intervene to redefine DNF expressions for new categories from scratch. In recent years most techniques used in text document classification and categorization are developed based on machine learning technologies, which automatically build document classifiers by learning the characteristic of document categories from a set of training documents. The advantage of machine learning is that it does not heavily rely on manual labors during the model construction stage. Its effectiveness level, in many cases, is superior to that of professional human work. Consequently, automatic text categorization has become a major research area of machine learning. Many text categorization systems have been developed using different machine learning algorithms, including k-nearest neighbor (K-NN), neural networks, latent semantic indexing, probabilistic models, support vector machine, and etc.

The initial application of k-Nearest Neighbor (K-NN) to text document categorization and classification was introduced by Masand and his colleagues [7, 10], and later it became a widely used method in text classification [11~13]. In text document classification and categorization, a document is often represented as a vector composed of a series of selected words called as feature vector. A K-NN based text categorization system is to find the K documents in the training data that are most similar to an input unknown document. The category contains the majority among the K best matched documents is considered as the category to which the unknown document belongs. The similarity between the unknown document and each training document is measured by a similarity function, which is critical in generating accurate results [11~13].

Neural networks (NNs) have been popular in text categorization and document retrieval [14~16]. The most popular neural network architecture for text classification is a multilayer neural network trained with the well-known backpropagation algorithm using supervised learning [17~22].

Self-Organized Map(SOM), also known as the Kohonen network [23], is a popular unsupervised neural network used in text classification. A SOM network attempts to cluster the training data while preserving the topological properties of the input space. During the training process, it builds the network, i.e. the map, by applying a competitive process to input examples. A fully trained SOM network can be used as a pattern classifier [3]. SOM has also been used in feature

selection for text categorization [24] and text clustering [25, 26].

Probabilistic Modeling has been used in text document classification. A widely used framework of probabilistic model for text document classification is derived from the Bayesian theorem of conditional probability [8, 27]:

$$P(c_i \mid \overrightarrow{d_j}) = \frac{P(c_i)P(\overrightarrow{d_j} \mid c_i)}{P(\overrightarrow{d_j})},$$

where dj is the input document, $\overline{d}_j$ is the feature vector that represents dj, ci is the ith document class, $P(\overline{d}_j)$ is the probability that a document dj represented by vector $\overline{d}_j$ occurs randomly, $P(c_i)$ the probability of a randomly picked document belongs to category ci, $P(\overline{d}_j \mid c_i)$ is the probability of document $\overline{d}_j$ occurring given that document dj is in document class ci, and $P(c_i \mid \overline{d}_j)$ is the probability of dj , represented by vector $\overline{d}_j$ belonging to document class ci. To simplify the calculation of the conditional probability, Naïve Bayes (NB) classifier has been applied to document classification [9]. A Naïve Bayes classifier assumes that the conditional probability of each term in the feature vector for a given class is independent of the conditional probability of other terms in the feature vector for a given class. This assumption is called class conditional independence. It makes the computation of the NB classifier far more efficient than the exponential complexity of a non-naïve Bayes approach since it does not require the term combination as predictors. Studies comparing different classification models have shown the performance of Naïve Bayes classifier is comparable with neural network classifiers and batch linear classifier [28, 29].

Support vector machine (SVM) approach was developed based on the structural risk minimization theories in statistical learning [30]. A SVM maps the input feature space to a high dimensional space through a kernel function. It then chooses the hyperplane with the maximum margin that can separate the positive from negative examples in the feature space. According to the structural risk minimization, the generalization error is bounded by the sum of the training set error and a term derived from the Vapnik-Chervonenkis (VC) dimension of the learning machine. Unlike traditional artificial neural networks (ANNs), which minimize the empirical training error, SVM aims at minimizing the upper bound of the generalization error, which represents the error on unseen data for a classifier. Thus high generalization performance can be achieved. SVM can potentially learn a larger set of patterns and be able to scale better than artificial neural networks [31]. Many published literatures show that SVM learning can lead to high performance in a broad range of pattern classification applications [31~34].

SVMs have been popular in text classification and categorization [35, 36]. SVM is designed for two-class pattern classification. However in text document classification, most applications involve more than two categories of documents. Therefore a text categorization system developed using SVMs usually use one of the following two approaches to build a multiclass SVM system. Let $N > 2$ be the number of text categories. The first approach would design N SVM classifiers, each of which discriminates the kth class against the remaining N-1 classes, k = 1 to N. The SVM associated with the class k seeks a decision surface in the feature space that separates class k from all other classes. Collectively the N SVM models result in N decision boundaries [37]. When a new document x is submitted to the system, all N SVMs are applied to x, and the class represented by the SVM that generates the largest output value is assigned to the input document x. The second approach is to train $N(N-1)/2$ SVMs, each of which is trained to pair-wisely separate two different classes in the training data set. Different voting strategies, such as Max-Wins [37] or directed acyclic graph (DAG) can be used to make the final classification decision based on the results from the $N(N-1)/2$ pair-wise SVMs [38].

Even with the advanced technologies discussed above text mining continuous to be a challenging research area. In this paper we present an innovative technique that combines automated text document classification with domain knowledge to derive a search result that precisely matches the input query.

3. SeaProSel: an Intelligent Vehicle Fault Diagnostic System

All automotive companies develop its own diagnostic codes that are used in their vehicle fault diagnostic processes. Some companies may have several sets of diagnostic codes with names such as CSC (Customer Symptom Codes), CCC (Customer Concern Codes), and etc. Without losing generality, we refer to such a code system as a vehicle diagnostic code (VDC). The SeaProSel system is designed to map a query description to a specific VDC that accurately matches the problem description. This query-to-VDC mapping is a M-to-M mapping. Multiple descriptions can be mapped to the same VDC, and one query can be mapped to multiple VDCs due to ambiguity in language. For example, the three customer descriptions given in Section 1 have the same diagnostic code that represent the problem of "ENGINE WOULD NOT START."

In addition to the synonymy and polysemy problems existing in general text documents, the documents occurring in vehicle fault diagnostics pose particular challenges: typos, grammar errors, self-invented terms and acronyms, inappropriate usages of punctuations, and etc. The proposed SeaProSel is designed to deal with these challenging issues in order to generate a unique VDC that accurately matches the input query. SeaProSel has a hierarchical matching and searching system that uses text data mining technology to quickly retrieve diagnostic code that accurately matches the

query, i.e. the problem description provided by a user. In the cases that no unique diagnostic code is found, the system will follow the given automotive diagnostic system to prompt questions to user in an attempt to obtain more information from the user about the vehicle problem. It then uses the answer provided by the user to search for the correct diagnostic code. This prompt and search process can be repeated until a unique diagnostic code is found.

Figure 1 illustrates the system architecture of the SeaProSel. At the first stage the SeaProSel directly searches for a VDC that matches the input query by using a Vector Space Model. We present two approaches, first is a Term-Code-Weight (TCW) model and the second a latent semantic indexing (LSI) model. The TCW model is a weighted matrix that is obtained through a machine learning algorithm. The LSI model uses the reduced-rank matrices to approximate the original TCW matrix. Each category and query is converted into a low-dimensional vector in a LSI space. Both models along with the critical research issues related to the two models, such as term selection and weight functions, are discussion in depth in section 3.A. If no exactly matched VDC is found, and the system outputs a list of best matched VDCs and then enters the second stage, *Prompt & Select*, which follows the vehicle diagnostic engineering hierarchy to prompt questions for user to select the more detailed and better described vehicle problem. Based on the user's answers, the system generates a new list, VDC_list2, which is a sublist of VDC_list1 and contains the VDCs that satisfy the user selected descriptions. The *Prompt & Select* process is repeated until a satisfactory VDC is found. This process is described in Section 3.B.

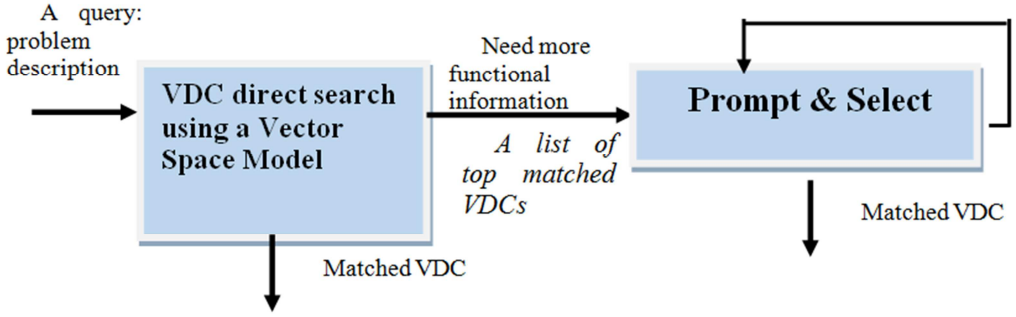


Figure 1. *Overview of SeaProSel System.*

A. Building a diagnostic document classification model using machine learning

In text data mining, a Vector Space Model (VSM) is an algebraic representation of text documents that contains vectors of identifiers or index terms such as words or phrases [39, 40]. In a VSM, all documents are represented in term weighted vectors. In this research we investigate two VSM models, TCW(Term-Code-Weight) matrix and LSI matrix. Both models are built using machine learning algorithms to represent vehicle fault diagnostic knowledge.

The two-dimensional TCW matrix, denoted as A_{MxN} , is generated from a training data set *Tr*, where *M* is the number of effective terms in *Tr*, and *N* is the number of diagnostic categories in *Tr*.

The training data set contains samples of customer descriptions of vehicle problems, and each description is associated with a correct diagnostic code assigned by automotive diagnostic experts. The machine learning algorithm consists of two major computational components: Term Extraction, and TCW Matrix Construction. The Term Extraction process involves the detection of a list of distinctive and effective terms, removal of punctuations and stopping words, word stemming and word variation detection. The TCW matrix contains *M* terms generated by the Term Extraction process, and *N* vehicle diagnostic codes, which represent document classes. An entry in a TCW, A_{MxN} (i,j), represents the weight of the i^{th} term associated with the j^{th} diagnostic code, which is generated based on the statistical analysis of the i^{th} term occurring in the training documents labeled with j^{th} diagnostic code. The TCW matrix is used to map directly from a problem description to the best matched diagnostic codes. The TCW matrix is built using a machine learning algorithm that contains the following major processes,

Document preprocessing
Document indexing
Term weight generation

Once we obtain the TCW matrix, the VDC that matches an input document can be generated by the process, *Vehicle Fault Diagnostics* using a similarity function.

1) *Document Preprocessing:* Two data noise problems associate with casual text documents, one is improper use of punctuations and special symbols, and another mislabeling document categories in training data. The misuse of punctuations in the text documents makes the text categorization less accurate. For example, the punctuations in 'ACCEL.', 'Dead.NEEDS' make the two terms different from their correct forms, 'ACCEL' and 'Dead' and 'NEEDS'. The existence of such inappropriate use of punctuation results in a lot of additional entries in the term list that actually should not exist, and cause misleading statistics. Three different types of processes are implemented. First type of processes involves the search for special symbols or punctuations appearing at one end or both ends of a word, for example, 'ACCEL.', '*DIESEL*', '****

TOW', and 'IN ****'. These symbols can be removed directly without any possibility of alternating the meaning of the word. The second type of processes is to search for special symbols or punctuations such as ".","(", ")', '&', '*', ' / ', '+', etc, These symbols are either removed or replaced by a space. The third type of processes is more sophisticated. It mainly concerns the punctuations such as ',', ':', '.', etc. When they occur as a part of initials, numerical or time/date formats, such as 'A.C.', 'P.I.D.', '2,000', '16.00', '4:00', '4×4', they are kept as part of the original string. If they occur between two words for example, 'engine,check', the punctuations are replaced with a space.

In supervised machine learning, each training data sample is assigned a target class code, i.e. diagnostic code in our application. The class code assignment is still been done largely by diagnostic experts. Some documents may have the class code missing, others may be assigned of multiple codes because the person who assign the class codes is not sure which one is correct. We developed the following procedure to deal with this problem.

For all the documents with missing labels, we build a standard diagnostic code matrix, DC, based on standard descriptions of diagnostic code, which are available in mechanics' handbook. We extract the training documents with specific diagnostic codes to form a subset of training data, denoted as TrC, which is then used to generate the TCW matrix $C \in R^{p \times q}$ and term list T_Lc, where p is the number of index terms, and q is the number of codes. For a document q with n labels, X1, X2, …, Xn, the relevance between q and Xi is calculated using the cosine similarity function shown below:

$$s_i = sim(\vec{q}, \overrightarrow{C_{X_i}}) = \frac{\sum_{j=1}^{p} q_j (C_{X_i})_j}{\left(\sum_{j=1}^{p} (q_j)^2 \sum_{j=1}^{p} ((C_{X_i})_j)^2 \right)^{\frac{1}{2}}}.$$

Let the similarity scores between q and the n diagnostic code vectors be $s_1, s_2, \ldots, s_n$, and $S_{max} = Max\{s_i \mid i = 1, \ldots, n\}$. The diagnostic code corresponding to S_{max} is assigned to document q. In the cases that multiple diagnostic codes are useful, we set a threshold *th*, and if the difference between S_{max} and s_i is smaller than *th,* diagnostic code X_i is also assigned to the document q.

2) *Document Indexing:* The terms used in the TCW matrix need to be derived automatically from training documents, and carefully selected so they effectively represent document contents. We developed the following document indexing algorithm to extract effective indexing terms automatically from training data. Let us assume a collection of documents are to be classified into N diagnostic codes or categories, (C_1, C_2, … C_N), and we have training documents Tr_1, Tr_2, …, Tr_N, where Tr_i contains the training documents belonging to category i, i = 1, …, N. The objective of the following algorithm is to generate a list of indexing terms, T_L, where each term $t_i \in T\_L$ that effectively represents the contents in the documents contained in Tr, where Tr = $Tr_1 \cup \ldots \cup Tr_N$. The document indexing algorithm contains the following major computational components.

Step 1: Extract all distinct terms from Tr to form an initial term list T_L.

Step 2: Generate a stop word list, stop_word list, which is used to make sure those words do not occur in the term list T_L. T_L contains the words, such as "the", "about", "an", "and", etc. that provide little information for document class discrimination. It also contains words have no specific meaning in a given application domain. For example, in vehicle fault diagnostic documents, terms such as 'customer', 'states', 'said', 'ck', 'cust', 'driving', etc. occur in documents of all classes.

Step 3: We implemented the well-known Porter Stemming algorithm (or 'Porter stemmer') [41] and applied it to the training data to generate groups of words that have the same stem, and the variant word forms is represented by one root word.

The Porter Stemming algorithm is based on the idea that the suffixes in the English language mostly consist of a combination of smaller and simpler suffixes. It has five computational processes. In each step, if a suffix rule matches with a word, then the conditions attached to that rule are tested on the resulting stem. A condition, for example, may be the number of stem length after suffix removal must be greater than the threshold. For example, the suffix of 'ing' can be safely removed from the word "singing", and the remaining part, i.e. the stem "sing" replaces the original word. Stemming word processing reduces the dimensionality to the word list significantly.

Step 4: Eliminating low-frequency words. Two frequency thresholds d_th, w_th, are defined to remove words occurring infrequently. A term is removed from T_L if its occurring frequency in the number of different vehicle diagnostic code categories is less than d_th or its occurring frequency in all training documents is less than w_th times. The optimal values for d_th and w_th can be obtained through experiments.

Step 5: Eliminating words evenly distributed across all categories. Words that have even distributions among all document categories are also removed from T_L, since they appear in the same frequency over all document categories.

Step 6. Output T_L, which is used for building the TCW matrix described below.

3) *Modeling diagnostic documents using a TCW Matrix:* An entry in a TCW matrix A_{MxN}, denoted as $a_{i,j}$, is the weight of the i^{th} term in *T_L* belonging to the j^{th} VDC, for $i = 1, \ldots, M$ and $j = 1, \ldots, N$. The weight in each entry in A_{MxN} is a function of the occurrence frequency of a term with respect to a category. The function is referred to a weight function. Term weighting is an important component for improving performance in the VSM based text mining [42]. Terms need to be weighted according to their importance for a particular

document category and for the whole document collection. A useful index term must fulfill a dual function: it occurs in the documents of the same category with high frequency so as to render the document retrievable, and it is useful to distinguish the documents of one category from the others. A term weight function is usually a combination of a local weight and a global weight function. The following describes three popular local weight functions.

Term Frequency: l_{ij} = tf_{ij}, which is the occurrence frequency of term i within document category j,

$$\text{Binary: } B_{ij} = \begin{cases} 0 & if\ tf_{ij} = 0 \\ 1 & f_{ij} > 0 \end{cases},$$

and

$$\text{Log function: } l_{ij} = \text{Log}_2(tf_{ij} + 1).$$

A local weight function provides a measure of how well that a term describes the document contents in a particular category. However, using only local weight is not enough to evaluate the importance of a term in the document classification. Some terms, due to their rarity use in a particular category of documents, are more important in identifying these documents than others do. Some terms, however, because they appear in many documents, are not useful to discriminate documents in one category from the others. A global weight measure is used to reflect the overall importance of the index term in the entire document collection. Four well-known global weights introduced by Dumais [43] are:

$$\text{Normal: } \sqrt{\frac{1}{\sum_j tf_{ij}^2}},$$

$$\text{GfIdf: } \frac{gf_i}{df_i},$$

$$\text{Idf: } \log_2\left[\frac{ndocs}{df_i}\right] + 1,$$

$$\text{Entropy: } 1 - \sum_j \frac{p_{ij}\log(p_{ij})}{\log(ndocs)} \text{ where } p_{ij} = \frac{tf_{ij}}{gf_i},$$

where df_i, the document frequency, is the total number of documents in the document collection, i.e. training data, that contain term i, gf_i, the global frequency, is the frequency of term i occurring in the entire document collection, and *ndocs* is the total number of documents in the document collection.

Different weight functions transform the raw occurrence frequency of a term in a document to different weights. In general, the entry $a_{i,j}$ of a TCW matrix A is a function of a local and a global weight components. The most commonly used term weight functions are listed in Table 1. These weight functions have been evaluated through extensive experiments and the results are discussed in Section 4. Based on these experiments, the proposed SeaProSel system uses the tf-idf weight function in its VCD direct search component.

Table 1. *Popular weight functions.*

Entropy	**Gfidf**	**Normal**	**tf-idf**	**B-idf**
$tf_{ij} * (1 - \sum_j \frac{p_{ij}\log(p_{ij})}{\log(ndocs)})$	$tf_{ij} * \frac{gf_i}{df_i}$	$tf_{ij} * \sqrt{\frac{1}{\sum_j tf_{ij}^2}}$	$tf_{ij} * \log\left[\frac{ndocs}{df_i}\right] + 1$	$B_{ij} * \log\left[\frac{ndocs}{df_i}\right] + 1$
B-normal $B_{ij} * \sqrt{\frac{1}{\sum_j tf_{ij}^2}}$	Log-idf $\log(tf_{ij} + 1) * (\log\left[\frac{ndocs}{df_i}\right] + 1)$	Log-entropy $\log(tf_{ij} + 1) * (1 - \sum_j \frac{p_{ij}\log(p_{ij})}{\log(ndocs)})$	log-Gfidf $\log(tf_{ij} + 1) * \frac{gf_i}{df_i}$	log-norm $\log(tf_{ij} + 1) * \sqrt{\frac{1}{\sum_j tf_{ij}^2}}$

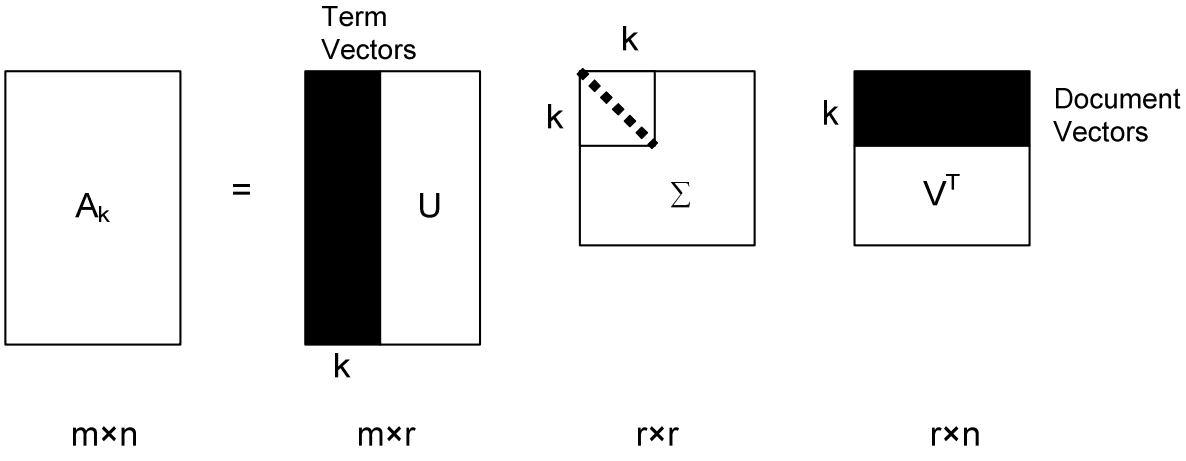


Figure 2. *A rank-k approximation matrix.*

4) *Modeling diagnostic documents using a LSI Matrix:* A popular variant of TCW matrix is constructed using latent semantic indexing (LSI) method [44~46]. It uses the reduced-rank matrices to approximate the original TCW matrix. Each category and query is converted into a low-dimension vector and mapped into the LSI space. Relevance measures for the user query are also performed in this space.

A LSI matrix is built from the TCW matrix. A is decomposed into the product of three matrices $A = U\Sigma V^T$, where $U^T U = V^T V = I_n$, and $\Sigma = diag(\sigma_1,...,\sigma_n)$, σ_i >0 for $1 \le i \le r, \sigma_j = 0$ for j ≥ r+1. Matrices U and V contain left and right singular vectors of A, respectively, and diagonal matrix Σ contains singular values. A rank-k approximation to A is represented $A_k = U_k \Sigma_k V_k^T$, where Uk and Vk are constructed by taking only the k largest singular values of Σ along with their corresponding columns in the matrices U and V respectively. Ak is the unique matrix of rank k that is closest in the least squares sense to A. Fig. 2 illustrates the relationship between A and Ak.

The SVD method attempts to capture most important underlying structure in the association of terms and documents. Since k is usually much smaller than the number of terms m, some "noise" are eliminated by deleting low ranking columns. Because SVD is a strictly mathematical method, the contents of the matrices are not interpretable with respect to the documents or terms it analyzes. The best rank k in the SVD model depends on the training data, which will be further discussed in the experiment section. However, it is a powerful technique to reduce the dimension of any term-by-document matrix.

5) *Vehicle Fault Diagnostics using TCW and LSI matrices:* The objective of vehicle fault diagnostics is to classify the user query to a diagnostic code that accurately matches the input query. Vehicle Fault Diagnostics using TCW algorithm consists of two processes, formulating query vector, and measuring similarity between the term vector and a column vector in the TCW. An input problem description d is firstly preprocessed using the same procedures as described earlier, including removing unnecessary punctuations, stop words and word stemming, etc. It is then transformed into a term vector $\overline{q}$ with the same length M of T_L. Let $\overline{q} = (q_1,...,q_M)^T$, where q_i is the frequency of the ith term on T_L occurred in the query document d, i = 1, …, M. The classification decision on which VDC category that best matches with $\overline{q}$ is made based on the similarity measure between $\overline{q}$ and each column vector of A. Let the column vectors of A be $\overline{a}_j = (a_{1j},...,a_{Mj})^T$, j=1, …, N. We use the following cosine based similarity measure to generate a similarity score between the vector $\overline{q}$ and the column vector $\overline{a}_j$,

$$r_j^c(\overline{q},\overline{a}_j) = \frac{\overline{q} \bullet \overline{a}_j}{\|\overline{q}\| \bullet \|\overline{a}_j\|} = \frac{\sum_{i-1}^{M} q_i a_{ij}}{\sqrt{\sum_{i=1}^{M} q_i^2 \sum_{i=1}^{M} a_{ij}^2}}$$

After similarity score is obtained for every VDC category and the input query, there are two approaches by which our system can use to determine if the diagnostic codes with the best similarity score should be returned as matched code class. One method is to use a threshold: all diagnostic categories with similarity scores larger than the threshold are regarded as relevant and assigned to the query. The second method is to output the VDC category represented by the column vector in the TCW matrix that has the highest similarity score with the input query vector.

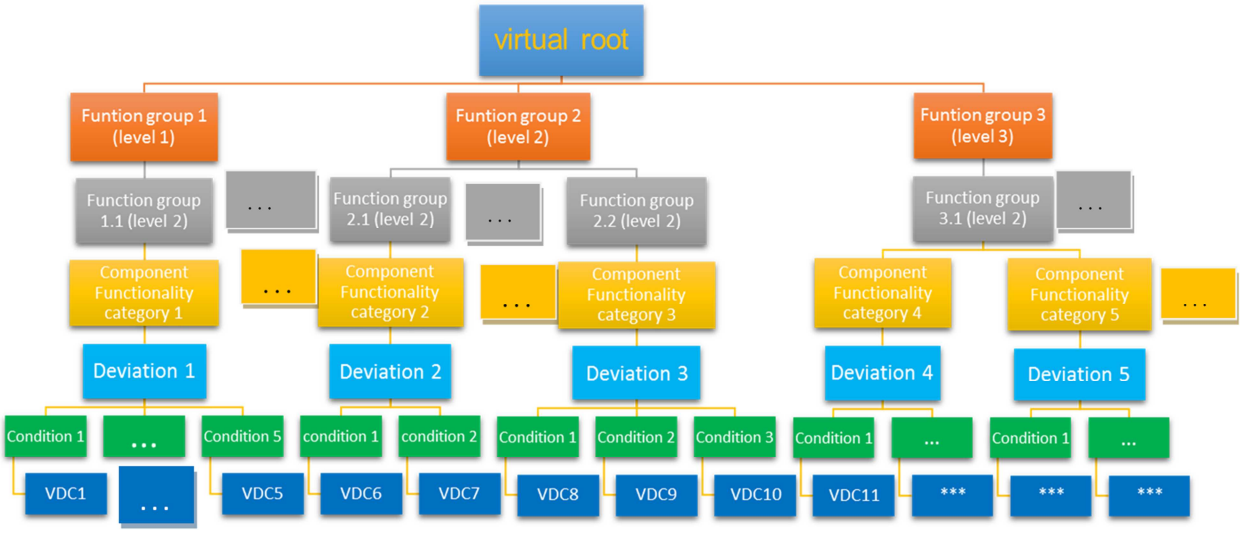
In the LSI model, a user's query is represented by a vector in the reduced-rank space. From a user query, we first construct the same term vector $\overline{q}$ as in the TCW model. Then $\overline{q}$ is converted into the vector in the reduced-rank space by the following formula: $\overline{q}_k = \overline{q}^T U_k \Sigma_k^{-1}$. The classification decision on which VDC category that best matches with $\overline{q}_k$ is made based on the similarity measure between $\overline{q}_k$ and each column vector of Ak, the same process as in the TCW classification process described above.

Both TCW based and LSI based VDC direct search system will be evaluated in Section 4.

B. Integrating automatic search with vehicle engineering structure

The proposed vehicle fault diagnostic system, SeaProSel, is an integration of direct search using the TCW matrix and the progressive prompt and select process based on a hierarchical vehicle fault diagnostic engineering structure. A diagnostic code system is usually organized in a hierarchical structure that contains multiple levels of functional descriptions, and each level provides descriptions about a class of symptoms, specific function or component faults, conditions, and etc. In this representation, the vehicle fault diagnostic codes are represented in the leaf nodes, the root of the tree is the entire vehicle system, and the subsequent levels represent the hierarchies of subsystems, components or devices. Figure 3 shows an example. The highest layer has three function groups. Under each function group, there are sub-function groups. For each function group at level 2, there are component groups. Under each component group, there are different categories of deviations, under each of which, there is a layer of conditions. Each node in the tree is accompanied with a brief description. For example, a description for a function group could be "Engine with mountings and equipment," a description for a component category under the function group could be "starting," the descriptions for conditions under such function group could be "engine turns", "cold start" or "unsure when", and a description for a VDC could be, "ENGINE WOULD NOT START."

The SeaProSel system uses the TCW matrix to directly obtain highly matched diagnostic codes, interacts with user by prompting diagnostic questions based on the vehicle engineering structure, takes user's selection/answer to either generate a diagnostic code that accurately match the user's answers or lead to the next level of functional prompts. Figure 4 shows the architecture of the SeaProSel system developed based on the vehicle fault diagnostic system illustrated in Figure 3. In Figure 4, SQ_1 represents the input problem description, SQ_2 represents the selected level 2 function group, and SQ_3 represents the selected component/functionality. The SeaProSel algorithm has the following major computational steps.



Figutre 3. *A hierarchical vehicle fault diagnostic system architecture.*

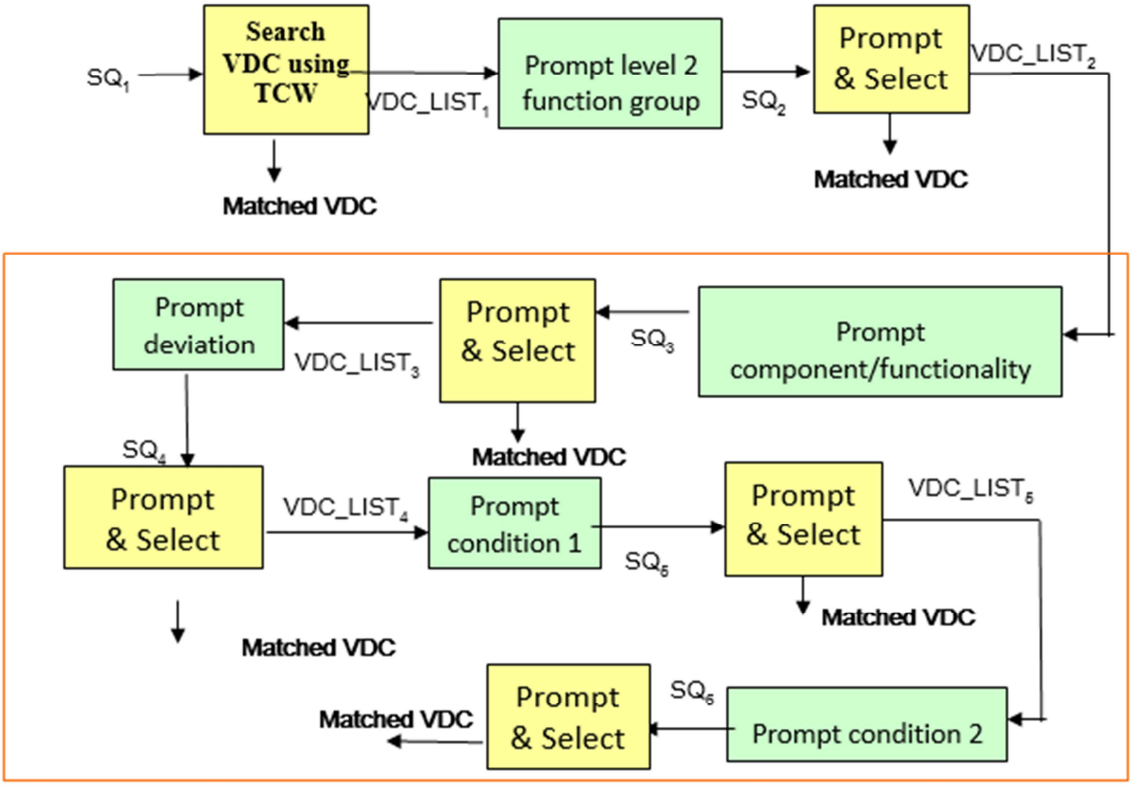


Figure 4. *Overview of processes in SeaProSel for vehicle fault diagnostics.*

Step 1: process the input query document SQ1 using the procedure, *VDC Direct Search using TCW*. Let the output of the procedure be F1 (SQ1) = {VDC_1, $conf_1$, …, VDC_k, $conf_k$} , where $conf_1 \leq conf_2 \leq \ldots, \leq conf_k$

Step 2: If $\Delta 1 = conf_1 - conf_2$ is high, then output VDC_1 and exit

Step 3: Find $c \leq k$ such that $\Delta c = conf_c - conf_{c+1}$ is high

Step 4: Find the node H in the tree structure such that

Found_Codes = {VDC_1, …, VDC_c} are all H's descendants, and no other nodes in the tree has this property except H's parent nodes.

Example 1: If Found_Codes = {VDC_8, VDC_9, VDC_{10} } , the H code is "Deviation 3".

Example 2: If Found_Codes = {VDC_6, VDC_9, VDC_{10} } , the H code is "Function Group 1".

Step 5: Call the following *Prompt & Select* procedure.

Step 5.1 Following the H node's direct descendants, and present the descriptions associated with the descendants to the user.

For example 1, the descriptions of "Condition 1," "Condition 2" and "Condition 3" under "Deviation 3" are presented to the user

Step 5.2 Based on the user selection, if the unique VCD is found, output the VCD and exit the program.

Step 5.3 Follow the descendant node selected by the user and find the VCDs that match the user's selections, and denote them F2 (SQ2) = {VDC'_1, $conf'_1$, …, VDC'_{k1}, $conf'_{k1}$} , where $conf'_1 \leq conf'_2 \leq \ldots, \leq conf'_{k1}$

Step 5.4 If $\Delta 2 = conf'_1 - conf'_2$, is high, then output VDC'_1 and exit

Step 5.5: goto Step 3.

4. Experiments

We were provided by an automotive company with the hierarchical vehicle fault diagnostic system illustrated in Figure 3. The hierarchical vehicle fault diagnostic system has 540 vehicle diagnostic codes, and each node is accompanied with a general description of the vehicle problems the node covers.

We conducted three different sets of experiments to evaluate, respectively, different weight functions, TCW matrix verse the LSI matrix, and the entire SeaProSel system.

The TCW and LSI matrices were all trained on a data set of 200,000 real-world customer descriptions of vehicle problems. After removing extraneous documents and eliminating documents with wrong labels, we had 199,552 valid documents as training data. After data preprocessing such as punctuation preprocessing, Portel stemming and typo removal, the number of index terms generated from the training data were reduced from 7033 to 3883. The TCW and LSI components as well as the weight functions were evaluated on TEST6K, a testing set of 6000 vehicle diagnostic documents collected from different retailer service shops in USA during one week time period. All test documents were labeled with true diagnostic codes by auto technicians.

The following evaluation criteria are used to analyze system performances. For each input query, two levels of matching accuracy are measured: the low level of matching (LLM) and high level matching (HLM). If the *VDC Direct Search* system using either TCW or LSI matrix returns the correct VDC code, i.e. it exactly matches one of the leaf nodes in the hierarchical vehicle fault diagnostic system shown Figure 3, then it is a low level matching. In this case, the ProSeaSel system will output the VDC code and terminate the search. If the VDC Direct Search system returns the code that does not match any of the leaf nodes but matches the correct higher level categories in the hierarchical system shown in Figure 3, then it is a high level matching. For example, if an input query's true VDC is "vdc1", but the output of the VDC Direct Search system is "vdc5". Since "vdc1" and "vdc5" have the same deviation category, the system has a wrong LLM, but a correct HLM. Based on these two types of matching criteria, we define two accuracy measures of system performances when a batch of test queries is used as test data, Exact Match Rate (EMR) and Category Match Rate (CMR). EMR is defined as the number of correct matched outputs in LLM over the size of the training data, and CMR the number of correct outputs in HLM over the size of the training data.

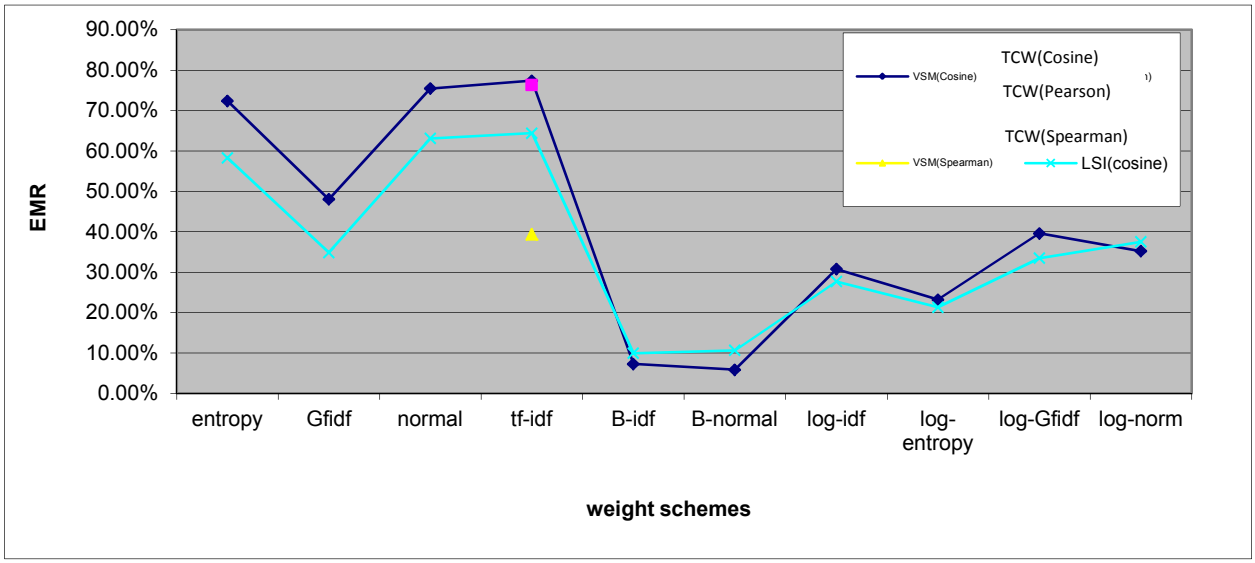


Figure 5. *Effects of Weight Schemes and Similarity Functions.*

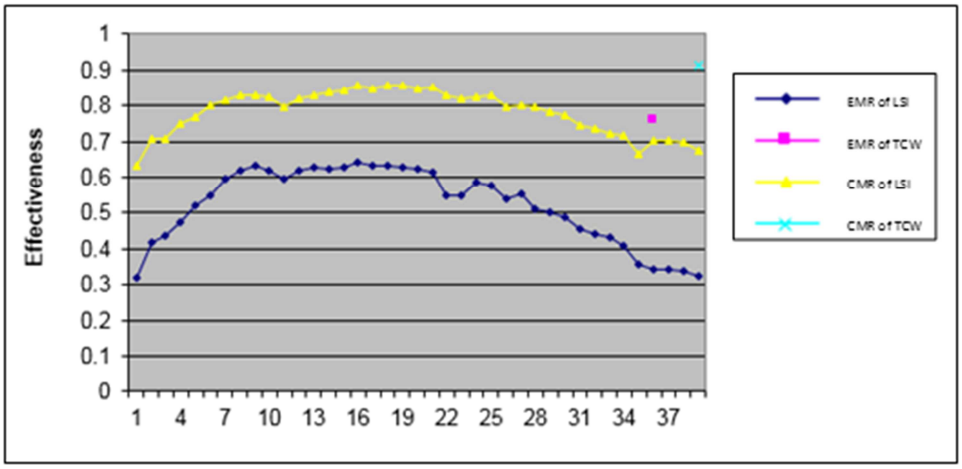
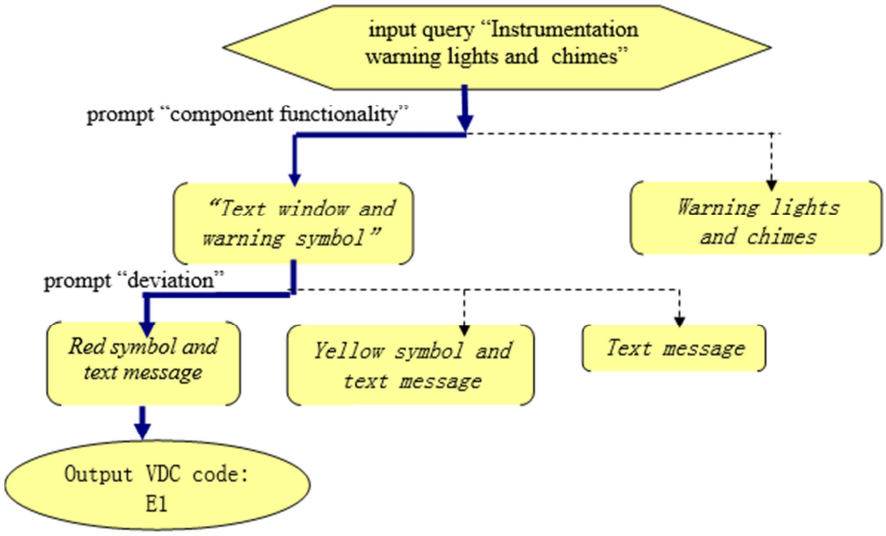
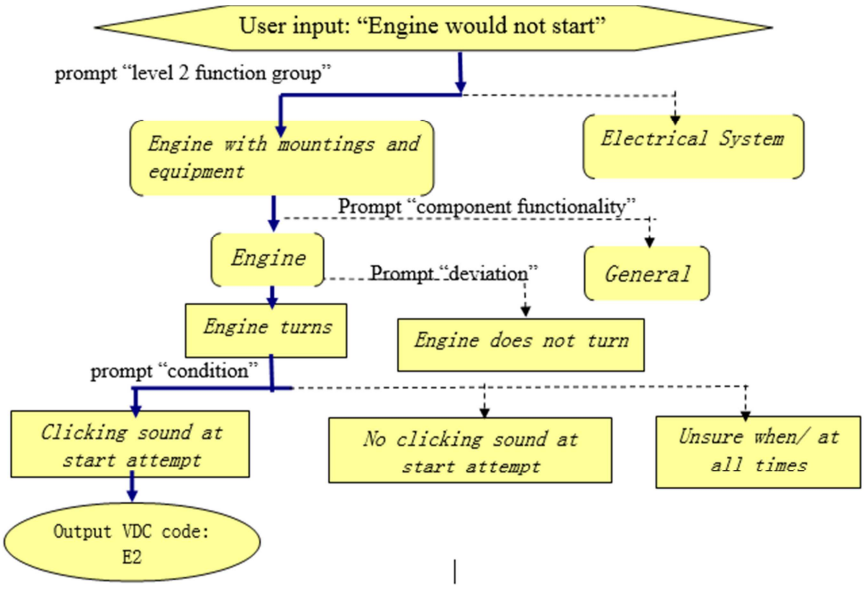


Figure 6. *Performances of the LSI systems of various K-values.*



(a)



(b)

Figure 7. *Two examples of query processes by SeaProSel system*

A. Evaluation of weight functions

As described before, several weight schemes can be used in both TCW and LSI models. All weight functions shown in Table 1 were implemented in both the VSM and the LSI models. We used three different similarity functions: Cosine, Pearson, and Spearman in the query classification processes. The results are shown in Figure 5. It shows that the best result is generated by the TCW model that uses the tf-idf weight function combined with cosine similarity function. Pearson similarity measure used in the TCW model combined with the tf-idf weight function achieved the similar performance to that of cosine measure. Both of them are better than the Spearman similarity function.

B. Evaluation of TCW and LSI models

The accuracy of the LSI model is heavily affected by value of rank, *K*. Since the results above indicates that the weight scheme tf-idf combined with cosine measure produced the highest EMR, they are used in both TCW and LSI systems. In order to explore the effects of parameter K, we applied various K values to construct the singular value decomposition matrices, $A_k = U_k \Sigma_k V_k^T$, and compared the performances of these SVD matrices with the TCW matrix. Theoretically, the range of K-value is between 1 to the number of columns of TCW matrix. However, our experiments shows that EMR degrades rapidly when K-value is larger than 40 in this application. Figure 6 showed that the CMR and EMR of LSI models with K values between 1 and 40, as well as the performances of the TCW classification system. It appears that the best performance has been achieved with K equal to 17. But the TCW outperformed the best LSI system by more than 11%.

C. Evaluation of SeaProSel

We tested the entire SeaProSel system on a set of 3273 query examples, none of which were included in the training data. The performances are analyzed as follows. 97% of the test queries were answered with unique and correct VDCs by the *VDC direct search using TCW Model* without going to the *Prompt & Select* process. The other 3% of the test queries were processed through subsequent Prompt & Select processes. At the end of the processes, correct VDCs were found for all test queries. Overall the prompt and select process was used at the rate of 0.065/query.

Figure 7 shows the processes of two test queries by the SeaProSel system. In Figure 7 (a), the test query was "Instrumentation warning lights and chimes". The *VDC direct search* component returned multiple mached VDCs. By finding the H node of these VDCs, the *Prompt & Select* component present to the user the descriptions of different problems at the component functionality level (see Figure 3) related to the input query. After the user selected "Text window and warming symbol", the SeaProSel went on to the questions at the "deviation" level. When the user selected "Red symbol and text message", a unique VDC code is found that matches the input query as well as the answers selected by the user during the *Prompt & Select* processes.

In the second example (see Figure 7 (b)), the test query is "Engine would not start". The *VDC direct search* component returned multiple VDCs, which is represented as VDC_List1. By finding the H node of these VDCs on the VDC_list1, the *Prompt & Select* component gave descriptions of different problems at level 2 function groups related to the VDC_List1. After the user selected "Engine with mountings and equipment", the SeaProSel went on to display the questions under the selected node at the "Component Functionality" level. When the user selected "Engine", the SeaProSel went on to display the questions under the selected node at the "Deviation" level. When the user selected "Engine turns", the SeaProSel went on to display the questions under the selected node at the "Condition" level. When the user selected "clicking sound at start attempt", a unique VDC, E2, is found that matches the input query as well as the answers selected by the user during the *Prompt & Select* processes.

5. Conclusion

We have presented an intelligent vehicle fault diagnostic system, SeaProSel. SeaProSel consists of two major components, *VDC Direct Search* using a VSM, and *Prompt & Select*. Two VSM technologies were developed, implemented and evaluated, a TCW model and a LSI model. Both models were developed based on machine learning and text mining techniques. The *Prompt & Select* component is a system that is built upon a vehicle fault diagnostic engineering structure with a progressive process of query, select, and search to achieve efficient and accurate classification of vehicle problem descriptions. We also presented algorithms for preprocessing text documents that contains spelling errors, typos and self-invented terms, choosing effective weight functions, and building an effective TCW matrix and LSI matrices from a given training data set. We have conducted extensive experiments to evaluate the algorithms and the entire SeaProSel system.

Based on our experimental results we conclude that, in the application domain of vehicle fault diagnostic text documents, the tf-idf weight function gives the best performance when it is used in either TCW or LSI models with the similarity function being either Cosine or Pearson. In terms of the optimal ranks in the LSI systems, our experiments show that the optimal K values are in the range of K=13 through K=19, with K=17 giving the best performance. When we compare the performances of the TCW model with the best LSI model, i.e. the LSI system used K=17, we notice that the TCW model outperforms the best LSI system by more than 11%.

The SeaProSel system is implemented based on a real-world vehicle diagnostic code system with 540 different code classes, and is evaluated on 3273 query documents, which are verbatim vehicle problem descriptions by customers. The SeaProSel system achieved 97% accuracy in finding the diagnostic codes directly based on the *VDC Direct Search using TCW*. Through the innovative processes of *Prompt and Select* procedure, the SeaProSel was able to find the correct diagnostic code 100% for all test queries.

Our major contributions are summarized as follows.

(1) Presented an innovative computational framework, SeaProSel, that combines automatic search with engineering structural search through a Prompt & Select strategy. Experimental results show that SeaProSel is effective in searching for diagnostic code accurately matching a given problem description.

(2) Presented new algorithms for learning automotive diagnostic code using TCW matrix and LSI model. Based on our experimental results, TCW is more effective in the application of casual text document categorization

(3) Presented new algorithms for preprocessing engineering diagnostic documents.

Although the application domain we presented is in the area of vehicle fault diagnostic documents, some techniques we presented are applicable to other applications that involve processing casual text documents such as automated question answering services, and classification of Tweets, instant messages, and e-mail messages.

References

[1] Fang, J., Guo, L., Wang, X. D., & Yang, N. 2007. Ontology-Based Automatic Classification and Ranking for Web Documents. *Fourth International Conference on Fuzzy Systems and Knowledge Discovery -FSKD* , 2007.

[2] Zhuang, F. Z.; Luo, P.; Shen, Z. Y.; He, Q.; Xiong, Y. H.; Shi, Z. Z. & Xiong, H. 2012. Mining Distinction and Commonality across Multiple Domains Using Generative Model for Text Classification. *IEEE Transactions on Knowledge and Data Engineering,* Volume: 24 , Issue: 11, Page(s): 2025 – 2039, 2012.

[3] Huang, Y.H., Seliya, N., Murphey, Y. L., & Friedenthal, R. B. 2010. Classifying Independent Medical Examination Reports using SOM networks. *Proceeding of the 6th International conference on Data Mining*, Las Vegas, Nevada, USA, 2010, p58-64.

[4] Mencıa, E. L., Park, S. H., & Fürnkranz, J. 2010. Efficient voting prediction for pairwise multilabel classification. *Neuro computing* 73 pp.1164–1176, 2010.

[5] Zeng, Q.; Zhang, X.; Zhang, W.; Li, Z. & Liu, L. 2010. Extracting Clinical Information from Free-text of Pathology and Operation Notes via Chinese Natural Language Processing. *2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops*, pp 593-597, Hong Kong, 2010.

[6] Huang, Y. H., Murphey, Y. L., & Ge, Y. 2013. Automotive diagnosis typo correction using domain knowledge and machine learning. *IEEE Symposium Series on Computational Intelligence,* 2013.

[7] Creecy, R.M., Masand, B. M., Smith, S. J., and Waltz, D. L. 1992. Trading MIPS and memory for knowledge engineering: classifying census returns on the Connection Machine, *Communications of the ACM*, 35(8): p. 48—63, 1992.

[8] Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys,* 2002. 34(1): p. 1-47.

[9] Yang, Y. & Liu, X. 1999. A re-examination of text categorization methods. *Proc. 22th ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR'99)*. 1999. Berkeley, CA.

[10] Masand, B., Linoff, G., & Waltz, D. 1992. Classifying news stories using memory based reasoning. *Development in Information Retrieval*, 1992: ACM Press, New York, US.

[11] Radovanović, M. & Ivanović, M. 2008. Text mining: approaches and applications, *Novi Sad J. Math*. Vol. 38, No. 3, 2008, 227-234

[12] Lu, F. & Bai, Q. Y. 2010. Refined weighted K-Nearest Neighbors algorithm for text categorization. *International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, 2010.

[13] Bijalwan. V., Kumar, V., Kumari, P., & Pascual, J. 2014. KNN based Machine Learning Approach for Text and Document Mining. *International Journal of Database Theory and Application*, Vol. 7, No. 1, 2014, pp. 61 – 70.

[14] Baeza-Yates, R., Ribeiro-Neto, B., *Modern Information Retrieval*, 1999: Addison Wesley.

[15] Syu, I., Lang, S.D. & Deo, N.; 1996. Incorporating latent semantic indexing into a neural network model for information retrieval. *Proceedings of the fifth international conference on Information and knowledge management*, 1996.

[16] Chen. Z.H., Ni, C. W. and Murphey, Y. L., 2006. Neural Network Approaches for Text Document Categorization. *IEEE International Joint Conference on Neural Networks,* July, 2006.

[17] Zhang, M.L. and Zhou, Z. H. 2006. Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. *IEEE Transaction in Knowledge and Data Engineering*, Vol. 18, Issue 10, Oct. 2006.

[18] Cho, S.B. and Lee, J. H., 2003. Learning Neural Network Ensemble for Practical Text Classification. *Lecture Notes in Computer Science*, Volume 2690, Pages 1032– 1036, 2003.

[19] Yu, B.; Xu, Z. B. & Li, C. H. 2008. Latent semantic analysis for text categorization using neural network. *Knowledge-Based Systems*, 21- pp. 900–904, 2008

[20] Thi, H. N. T.; Huu, O. N. & Ngoc, T. N. T.;2013. A supervised learning method combine with dimensionality reduction in Vietnamese text summarization. *IEEE Computing, Communications and IT Applications Conference (ComComAp)*, 2013.

[21] Vinodhini, G. & Chandrasekaran, R.M.; 2014. Sentiment classification using principal component analysis based neural network model. *2014 International Conference on Information Communication and Embedded Systems (ICICES),* 2014

[22] Li, C. H. and Park, S. C., 2009. An efficient document classification model using an improved back propagation neural network and singular value decomposition. *Expert Systems with Applications,* 36, pp- 3208–3215, 2009.

[23] Kohonen, T. 1990. The self-organizing map. *Proc. of the IEEE*, 9, 1464-1479, 1990.

[24] Manomaisupat, P., and Abmad k. Feature Selection for text Categorization Using Self Orgnizing Map. *2nd International Conference on Neural Network and Brain, 2005*, IEEE press Vol 3, pp.1875-1880, 2005.

[25] Liu, Y.C.; Wang, X.L.; & Wu, C.; 2008. ConSOM: A conceptional self-organizing map model for text clustering. *Neurocomputing*, 71(4-6), 857-862, 2008.

[26] Liu, Y.C., Wu, C., & Liu, M. 2011. Research of fast SOM clustering for text information. *Expert Systems with Applications*, 38(8), 9325-9333, 2011.

[27] Lewis, D.D. 1998. Naive (Bayes) at forty:The independence assumption in information retrieval. *Proceedings of ECML-98.* Springer Verlag, Heidelberg, 1998.

[28] Friedman, N.; Geiger, D.; Goldszmidt. M.; 1997. Bayesian Network Classifiers. *Machine Learning*, November 1997, Volume 29, Issue 2-3, pp 131-163.

[29] Theodoridis, S.; 2015. Machine Learning: A Bayesian and Optimization Perspective. *Academic Press*, 2015.

[30] Vapnik, V.; 1995. The Nature of Statistical Learning Theory. *Springer Verlag, New York,* 1995.

[31] Mukkamala, S., Janoski, G., Sung, A H.. 2002. Intrusion Detection Using Neural Networks and Support Vector Machines. *Proceedings of IEEE International Joint Conference on Neural Networks,* IEEE Computer Society Press, pp.1702-1707.

[32] Murphey, Y.L.; Chen, Z.H.; Putrus, M. & Feldkamp, L.A. 2003. SVM learning from large training data set. *IEEE International Joint Conference on Neural Networks*, July, 2003.

[33] Hong, H.B.; Murphey, Y.L.; Gutchess, D. & Chang, T.S. 2005. Identifying knowledge domain and incremental new class learning in SVM. *IEEE International Joint Conference on Neural Networks,* July, 2005.

[34] Chapelle, O. & Vapnik, V. 2000. Model selection for support vector machines. In S.A. Solla, T.K. Leen, and K.R. Muller, editors, *Advances in Neural Information Processing Systems,* volume 12. MIT Press, Cambridge, MA, 2000.

[35] Zhang, W.; Yoshida, T.; & Tang, X. 2008. Text Classification based on Multi-word with Support Vector Machine. *Knowledge-Based Systems,* vol. 12, 2008.

[36] Feinerer, I. & Karatzoglou, A., 2010. Support Vector Machines for Large Scale Text Mining in R. *19th International Conference on Computational Statistics*, 2010.

[37] Hsu, Chih-Wei and Lin, Chih-Jen, 2002. A Comparison of Methods for Multiclass Support Vector Machines. *IEEE Transactions On Neural Networks*, VOL. 13, NO. 2, MARCH 2002.

[38] Platt, J. C., Cristianini, N., and Shawe-Taylor, J., 2000. Large margin DAG's for multiclass classification. *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, vol. 12, pp. 547–553, 2000.

[39] Huang, L.P. 2006. Intelligent Systems for text categorization and retrieval. *M.S. Thesis*, *Department of Electrical and Computer Engineering*, University of Michigan-Dearborn, 2006.

[40] Raghavan, V.V., & Wong, S.K.M. 1986. A Critical Analysis of Vector Space Model for Information Retrieval. *Journal of the America Society for Information Science*, 1986. 37(5): 279-287.

[41] Porter, M.F. 1997. An algorithm for suffix stripping. *Readings in Information Retrieval*, 1997. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

[42] Dumais, S.T., 1991. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 1991. 23(2): p. 229-236.

[43] Dumais, S.T., 1990. *Enhancing performance in latent semantic indexing (LSI) retrieval*. *Technical Report Technical Memorandum*, Bellcore, 1990.

[44] Dumais, S.T., Furnas, G. W., Landauer, T. K. and Deerwester, S. 1988. Using latent semantic analysis to improve information retrieval,. *In Proceedings of CHI'88: Conference on Human Factors in Computing*. 1988. New York: ACM.

[45] Jessup, E. R., & Martin, J.H., 2001. Taking a new look at the latent semantic analysis approach to information retrieval. *Computational information retrieval,* 2001: p. 121-144.

[46] Sebastiani, F. & Ricerche, C. N., 2002. Machine learning in automated text categorization. Journal of ACM Computing Surveys, Volume 34, Issue 1, March 2002.