
Extracting Diagnostic Knowledge from Natural Language Description



The University of Michigan-Dearborn
Henry W. Patton Center for Engineering
Education and Practice

Henry W. Patton Center for Engineering
Education and Practice
Technical Project Report

Extracting Diagnostic Knowledge from Natural Language Description

(Project #2005/6)

By:

Yi L. Murphey
Professor
Department of Electrical and Computer Engineering

Table of Contents

Synopsis	iii
1. Background	1
2. Objectives	2
3. Technical approach	2
4. Results	8
5. Conclusions.....	12
6. Impact	
Educational	13
Industrial	14
7. Acknowledgments	14
8. References.....	14

Synopsis

This project investigates advanced text document mining technologies with an application to engineering diagnostics. In the automotive industry, auto problem descriptions are often used as the first step of a diagnostic process. The automobile problem descriptions, which are casual statements made by customers and typed in by a mechanic at an auto shop, are mapped to diagnostic categories such as engine, transmission, electrical, brake, etc. This mapping of problem description to diagnostic categories is currently being done manually by mechanics who perform this task largely based on their memory and experience, which usually lead to lengthy repair processes, less accurate diagnostics, and unnecessary part replacement. This report presents our research results in technologies developed for the automatic mapping of problem descriptions to the correct diagnostic categories. Two advanced text categorization systems have been developed: DKE&TC, a vector-space based text classification system, and a hierarchical neural network system. Both systems have been trained and tested on large sets of data collected from auto dealers, and the results are very satisfactory.

1. Background

As computers become more powerful, data storage devices become more plentiful and the amount of information in digital form has dramatically increased. Many organizations have been collecting large amounts of information during daily operations, but most of them are unable to extract useful information from the data to improve their operations due to the lack of automatic tools in data mining. In the automotive industry there is abundant information available in human language description form that contains valuable vehicle diagnostic knowledge, marketing information, and consumer evaluation or satisfaction of certain vehicle models. We are particularly interested in applying text mining technologies to engineering diagnostics. In the automotive industry, several thousand auto problems per week are reported to various dealership shops. In a typical diagnostic process, problems are first described, in casual natural language, by either customers or maintenance technicians. Based on the description, the vehicle problem is assigned with a specific diagnostic code. Each diagnostic code represents a problem category of vehicles, which are used to guide the diagnostic process in search for the true causes of the problem. Problem descriptions are then mapped to diagnostic categories such as engine, transmission, electrical, etc. The diagnostic code is then used to guide the further diagnosis for the true causes of the vehicle problem within the category. This mapping of problem description to diagnostic categories is currently being done manually by mechanics who perform this task largely based on memory and experience, which usually leads to lengthy repair processes, less accurate diagnostics, and unnecessary part replacement. Hence, new techniques are in high demand to correctly and automatically map problem descriptions to problem categories.

Our research is focused on applying text mining technologies to automatically categorize text documents with applications to engineering fault diagnostics. The text descriptions of vehicle problems pose unique challenges: data is often ill-structured, and text descriptions often do not follow English grammar and often contain many self-invented acronyms and shorthand descriptions. A few examples are given below.

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."

These three descriptions are expected to be mapped to the same diagnostic category. From these examples we can see a number of problems, including synonymy, poor wording, and self-invented acronyms. Polysemy is also a problem; for example, the word "light" could mean a number of things in vehicle problem descriptions.

During this project period, we developed two automatic text categorization systems: Diagnostic Knowledge Extraction and Text Classification (DKE&TC), developed based on a vector space model, and a hierarchical neural network system.

DKE&TC has two components: DKE, a component that has functions for learning diagnostic categorization knowledge from a corpus of diagnostic text documents; and TC, which then applies the acquired knowledge to categorizing a customer's description of vehicle problems to diagnostic codes. We also present a number of important issues relating to text document classification, including term weighting schemes, latent semantic analysis (LSA), and similarity functions. The proposed system has been trained on more than 200,000 documents over more than 50 diagnostic categories and tested over 6000 documents. The hierarchical neural network approach was modeled based on the hierarchy of text documents. Therefore, it is best suited by text document categorization when high-dimensional feature spaces and a large number of categories are involved.

2. Objectives

The objective of this project is to investigate advanced technologies for automatic knowledge acquisition and text document classification. This objective is attained through the development of two diagnostic text classification systems, DKE&TC and a hierarchical neural network system. Both systems have been evaluated on real-world data.

3. Technical approach

Two approaches have been investigated. The first approach was based on a vector space model (VSM). It has two components, one which has functions for learning diagnostic categorization knowledge from a corpus of diagnostic text documents, and one which can apply the learnt knowledge to categorize a customer's description of vehicle problems into diagnostic codes. This approach involves the study on a number of important issues relating to text document classification including term weighting schemes, LSA, and similarity functions. The second approach is based on neural networks. This study involves the development of a system built upon multi-layered perceptrons trained on a large number of documents with a large number of document categories.

3.1 DKE&TC: A system for automatic learning of diagnostic knowledge from text documents

In automotive diagnostics, much like other engineering diagnostics, vehicle problems are organized into different categories. We have developed a system, DKE&TC, that automatically maps a problem description, q , to the correct diagnostic categories. The DKE&TC consists of two stages (see Fig. 1): Learning From Text (LFT), and Diagnostic Text Classification (DTC). The LFT system is the machine learning stage that attempts to extract diagnostic knowledge from text documents. The LFT system consists of two major components within our research interest, a Term Category Weight (TCW) matrix and the matrix obtained through the well-known LSA that uses Singular Value Decomposition (SVD) to obtain a weighted matrix. The output from the LFT system is a matrix representing document categories. The second stage is to transform the input problem description q to document vector and find the best matched diagnostic categories using the matrices generated by the LFT system.

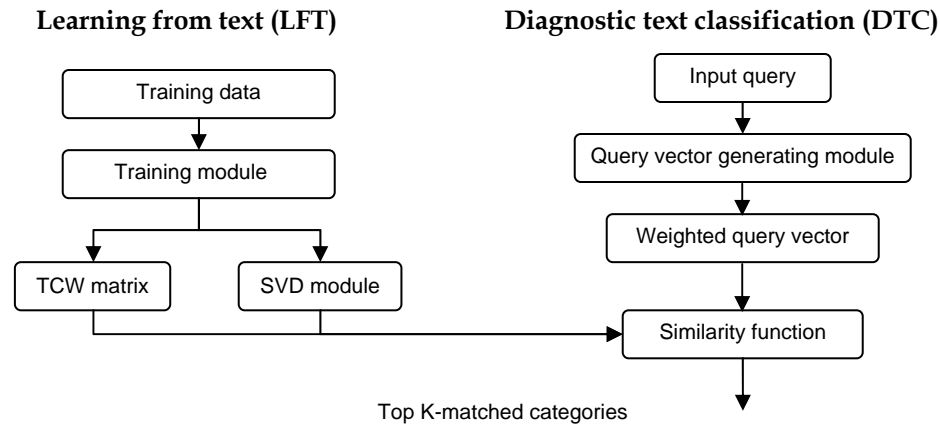


Figure 1. System architecture of Diagnostic Knowledge Extraction and Text Classification (DKE&TC).

3.1.1 Learning from Text (LFT)

The LFT component is developed based on the vector space model [1]. In the vector space model text objects are modeled as elements of a vector space. LFT consists of two major computational steps, the document indexing, and constructing an effective term by diagnostic category matrix. Let us assume a collection of diagnostic text documents are to be classified into N categories, C_1, C_2, \dots, C_N and we have training documents Tr_1, Tr_2, \dots, Tr_N , where Tr_i contains the training documents belonging to category i , $i = 1, \dots, N$. The following subsections describe the two computational steps.

Document indexing process

First we generate a term list T_L , where each term $t_i \in T_L$ is extracted from all documents in Tr , where $Tr = Tr_1 \cup \dots \cup Tr_N$. T_L is then processed by the following steps:

- Stopping word (function word) removal: since stopping words are unlikely to be useful for document matching, they are removed from T_L . Examples of stopping words are *the, from, to*, etc.
- Stemming words: morphological variants of the same lexical items are merged into a single root. For example: *leak, leaks, leaking* and *leaked* are all stemmed to *leak*.
- Selecting frequently occurring words: a word, w , is frequent if and only if it occurs in more than d_th documents and more than w_th times in all documents in Tr . Once a document threshold d_th and word threshold w_th are applied to T_L , infrequent words are removed from T_L .

Constructing an effective term-category-weight (TCW) matrix

In text document categorization, the term by document category matrix is considered one of the most important steps with respect to system performance. The entries of the matrix are usually weighted according to importance for a particular document and for the whole collection [2, 3].

In the proposed system, the TCW matrix is constructed as follows. The TCW matrix A has dimensions of M by N , where M is the number of terms in T_L , and N is the number of diagnostic categories. The entries of A , $a_{ij} = tf_{ij} * G_i$, are the multiplication of local and global weights defined as follows: tf_{ij} is the frequency of i th terms in T_L occurring in training data Tr_j for diagnostic category j , and G_i is the inverse document frequency defined as

$$G_i = \log_2 \left[\frac{ndocs}{df_i} \right] + 1$$

where $ndocs$ is the number of documents in the entire training data Tr , and df_i , stands for the document frequency of term i , defined as the total number of documents in Tr that contain term i .

The local weight is used to evaluate the importance of each term in the each category. However, sometimes less frequently used terms can be more critical in distinguishing one category from another. For example, in a document “check fuel leak, check and advise”, term “leak” is probably more meaningful than “check,” although “check” occurs more often than “leak”. Global weight is therefore introduced to reflect the overall importance of the index term in the whole document collection. The idea is that a term occurring rarely should have a high weight, and a term occurring frequently should have a low weight. Taken to the extreme, stopping words such as “the”, “an”, “a”, that appear in almost all documents, should ideally receive a global weight of 0. The column vectors in TCW matrix A can be interpreted as a representative vector for a diagnostic category.

Another important type of term by diagnostic category matrix is constructed using the latent semantic indexing (LSI) method [4, 5, 6], which is a variant of the vector space model. LSI is a statistical model of word usage that permits comparisons of the semantic similarity between pieces of textual information. While assuming the existence of some underlying or “latent” semantic structure according to the overall term usage pattern that is partially obscured by variability in term choice [5], LSI was originally designed to improve the effectiveness of information retrieval methods by performing text retrieval based on the derived “semantic” content of words in a query, as opposed to performing direct word matching. This approach avoids some of the problems of synonymy and polysemy. A truncated singular value decomposition (SVD) is used to build a lower dimensional space which is associated with this latent semantic structure. Each document and query is mapped into this space. Relevance judgments for the user’s query are also performed in this space.

In our model, we decomposed a weighted term frequency matrix A into the product of three matrices [3, 4]: $A = U \Sigma V^T$, where $U^T U = V^T V = I_n$ and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, $\sigma_i = 0$ for $1 \leq i \leq r$,

$\sigma_j = 0 = 0$ for $j \geq r+1$. Matrices U and V contain left and right singular vectors of A , respectively, and diagonal matrix Σ contains singular values of A . If only the k largest singular values of Σ are kept along with their corresponding columns in the matrices U and V , we obtain new matrices U_k, V_k with reduced dimension. A rank- k approximation to A is constructed with the following formula: $A \approx A_k = U_k \Sigma_k V_k^T$. A_k is the unique matrix of rank k that is closest in the least squares sense to A .

3.1.2. Diagnostic Text Classification (DTC)

The DTC component has the following two major computational steps: the formulation of a query document vector based input problem description, and the determination of the similarity between the query document and the diagnostic categories. An input problem description, d , is first transformed into a pseudo-document within the reduced term-category space. Namely, d is transformed into \bar{q} , which is a term vector with the same length M as T_L , with $\bar{q} = (q_1, \dots, q_M)^T$, where q_i is the frequency of the i th term on T_L occurring in the query document d , $i = 1, \dots, M$. The decision on the diagnostic category of \bar{q} is made based on the similarity measured between \bar{q} and each column vector of A , where the column vectors of A are $\bar{a}_j = (a_{1j}, \dots, a_{Mj})^T$, $j=1, \dots, N$. We have investigated three different similarity measures between the query vector \bar{q} and the TCF matrix A . The first similarity function of the query vector and a diagnostic category is measured by the cosine measure [7] denoted as follows:

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

Pearson's correlation coefficient is another measure of extent to which two vectors are related [8]. The formula for Pearson's correlation coefficient takes on many forms. A commonly used formula is shown below:

$$r_j^p(\bar{q}, \bar{a}_j) = \frac{M \sum_{i=1}^M q_i a_{ij} - \sum_{i=1}^M q_i \sum_{i=1}^M a_{ij}}{\sqrt{\left[M \sum_{i=1}^M q_i^2 - \left(\sum_{i=1}^M q_i \right)^2 \right] \left[M \sum_{i=1}^M a_{ij}^2 - \left(\sum_{i=1}^M a_{ij} \right)^2 \right]}}$$

The third measure we investigated in our system is the Spearman rank correlation coefficient. It provides a measure of how closely two sets of rankings agree with each other. The Spearman rank correlation coefficient is a special case of the Pearson correlation coefficient in which the data are converted to ranks before calculating the coefficient. It can be used when the distribution of the data makes the latter undesirable or misleading [9]. The formula for Spearman's rank correlation coefficient is:

$$r_j^s = 1 - \frac{6 \sum_{i=1}^M d_i^2}{M(M^2 - 1)}$$

where $\bar{d} = (d_1, \dots, d_M)^T$ is the difference vector between ranks of corresponding values of \bar{q} and \bar{a}_j .

3.2 A Hierarchical neural network system for text categorization

A multi-level document category system is a hierarchical structure that has links and nodes to represent relationships between document category concepts in the domain covered by the classification. The top category is the overall general concept of the whole document domain, which can be split into a series of subcategories, (C_1, C_2, \dots, C_N) , which form the first level category (L_1) in the system. Every category in L_1 can also be divided into smaller subcategories; and all of those categories derived from L_1 will constitute the second level category (L_2) in the system hierarchy. After repeating this procedure, a multi-level category system will be generated. In this paper, we focus on a two level hierarchy document categorization system, which can be a part of a larger document hierarchy. In particular we are considering the task of classifying an input document d into one of the leaf categories. The methodology discussed in the report can be extended to an m-level hierarchy of documents.

Figure 2 illustrates a two level hierarchy of document categories. The first level has N categories, each of which has its own subcategories listed at the second level of the hierarchy. Three different neural network systems were studied, each designed to categorize documents into the categories at the second level. We assume features used to represent each document category are term frequency vectors, which are used often in text document classification [10]. The following notations will be used throughout this paper. Tr^d is the set of all training documents, and is defined as

$$Tr^d = Tr_1^d \cup Tr_2^d \cup \dots \cup Tr_N^d$$

where Tr^d is a set of training documents belonging to level 1 category C_i , $i = 1, \dots, N$.

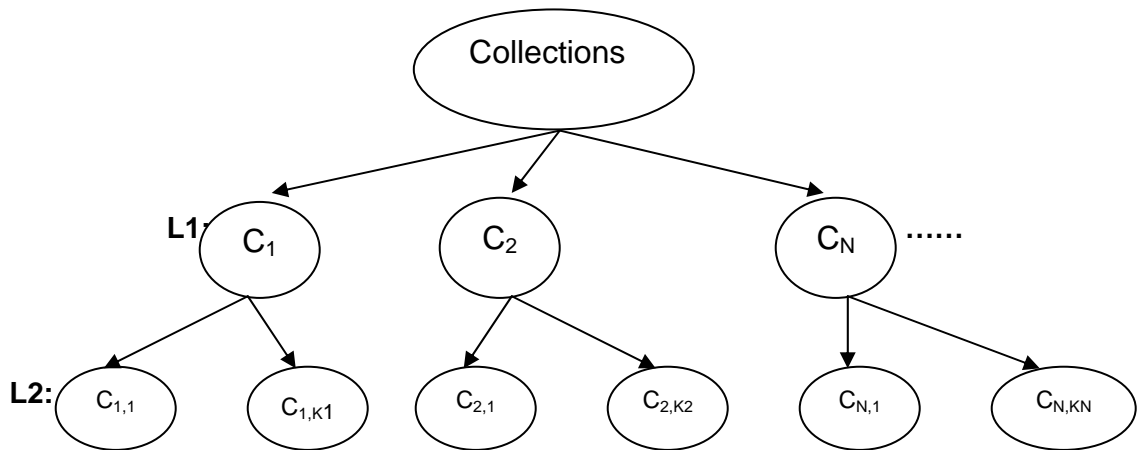


Figure 2. A two-level hierarchy of document categorization.

Each of the level 1 training data sets can be further decomposed into subsets according to its subcategories, i.e.,

$$Tr_i^d = Tr_{i,1}^d \cup Tr_{i,2}^d \cup \dots \cup Tr_{i,k_i}^d$$

where $Tr_{i,j}^d$ is a training data set for the documents belonging to category $C_{i,j}$, for $i = 1, \dots, N$, and $j = 1, \dots, k_i$. The number of all categories at level 2 is denoted as n , where

$$n = \sum_{i=1}^N k_i.$$

A hierarchical neural network system is developed for document categorization (see Figure 3). At level 1, a system of N binary neural networks, F , is trained to categorize documents into C_1, \dots, C_N , i.e. $F(\bar{x}) \rightarrow \{C_1, \dots, C_N\}$. Each neural network in F has the same feature space Ω , with an input layer of η nodes and an output layer of one node to represent two classes, “ C_i ” or “Not C_i ”. The network also gives the belief value of two classes. Each neural network in F is trained on Tr^d with the modification of category labels to binary values: documents in C_i are labeled as “1” and all others are “0” for $i = 1, \dots, N$. The output of F is produced by a Winner-Take-All decision process applied to the N neural network outputs, namely, the output of F is just the category C_i with the maximum belief value from all N neural network outputs.

The level 2 has N neural networks, F_1, F_2, \dots, F_N , each with its own feature space Ω_i . Neural network F_i has η_i input nodes and k_i output nodes, each of which represents a leaf category, $C_{i,j}$, $1 \leq j \leq k_i$. F_i is trained on Tr_i^d , the documents belonging to C_i in the training data set, for $i = 1, \dots, N$.

During the query stage, for a query document d , its feature vector \bar{x} is extracted based on feature space Ω . First the level 1 neural network system F is applied to \bar{x} , where $F(\bar{x}) = C_i, 1 \leq i \leq N$. Then neural network F_i is applied to \bar{x}_i , i.e. $F_i(\bar{x}_i) \rightarrow \{C_{i,1}, C_{i,2}, \dots, C_{i,k_i}\}$, where \bar{x}_i is the feature vector extracted from document d based on the feature space Ω_i for category C_i .

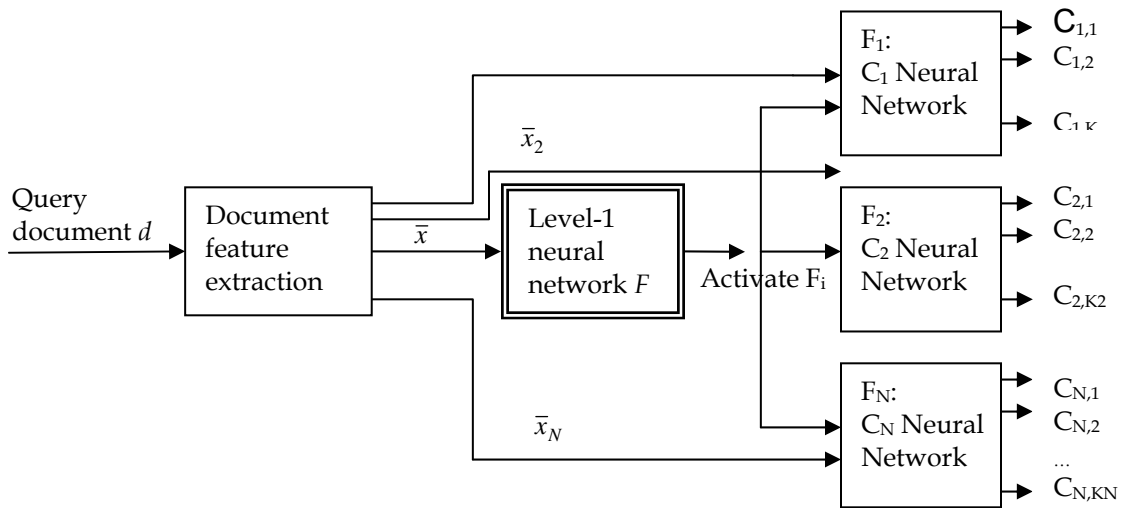


Figure 3. A hierarchy of neural networks for document categorization.

The hierarchical system architecture is flexible to provide top K -matched categories, $K > 1$. The first layer network system F can return top K -matched level 1 categories, $C_{i1}, C_{i2}, \dots, C_{iK}$. The neural networks at level 2, $F_{i1}, F_{i2}, \dots, F_{iK}$ are activated to find the top K -matched level two categories.

This architecture has the following advantages:

- Efficient training time at level 1: At level 1, each of the N neural networks has a binary output, making the training time very efficient for F . Furthermore, all $2*N$ neural networks in the entire system, N at level 1 and N at level 2, can be trained independently on different computers, which can speed up overall training time.
- Lower dimensional feature spaces at level 2. The N neural networks at level 2 have their own feature spaces defined on the training data of their respective categories. The dimensions of categorical feature spaces are usually much lower than the feature space of the entire training data. Furthermore, the N neural networks can be modified individually without affecting other neural networks.

However, it is important to make sure that the neural network system F at level 1 is reliable. If F makes a wrong prediction, then the error propagates to the neural networks at the second layer.

4. Results

Both the DKE&TC system and the hierarchical neural network system have been fully implemented, trained, and tested on a large collection of diagnostic text documents provided by an automotive company. Based on practical considerations for each input query document d , both systems find the three diagnostic categories that best match d . The performance measure is the percentage of the queries in the test set whose true categories are contained in the matched categories returned by the systems.

The following sections describe the performances of these two systems.

4.1 Experimental results of DKE&TC system

The proposed DKE&TC system has been fully implemented. The LFT program used a training data set of 200,000 documents that describe 54 different categories of automotive problems to construct a TCW matrix A with dimensions 3972×54 . Based on the practical consideration, for each input query document d , the DKE&TC finds the three diagnostic categories that best match d . The performance measure is the percentage of the queries in the test set whose true categories are contained in the matched categories returned by DKE&TC.

The performance of DKE&TC system and the SVD system on the test set of 6,000 queries is shown in Figure 2. The retrieval performance of SVD is heavily affected by its value of rank K . We explored the effects of varied K values used in SVD and compared it with the DKE&TC's performance. It appears that the best performance was achieved with $K=14$. However, the DKE&TE system outperformed the best SVD system by more than 11%.

We also compared the performance of the DKE&TE system with other weighing schemes commonly used in text document categorization. The results are shown in Figure 4, along with the performance of the DKE&TC systems that used three different similarity functions: cosine, Pearson, and Spearman. The definition of these weight schemes is shown in Table 1, in which tf_{ij} is the frequency of term i occurring in document category j , df_i is the total number of document categories in the training data that contain term i , gf_i is global frequency at which term i occurs in the entire training data, and $ndocs$ is the total number of document categories in the entire collection. In addition, we define

$$p_{ij} = \frac{tf_{ij}}{gf_i} \text{ and } B_{ij} = \begin{cases} 0 & \text{if } tf_{ij} = 0 \\ 1 & \text{if } tf_{ij} > 0 \end{cases}$$

Figure 4 shows that the DKE&TC systems using the similarity functions of Pearson and cosine gave the best performances.

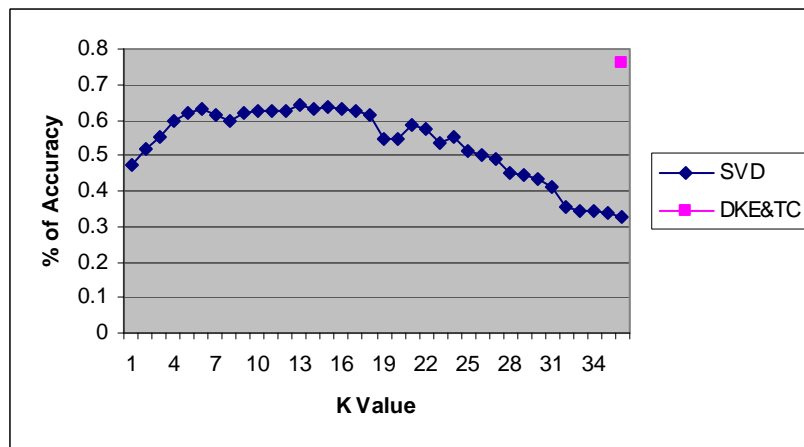


Figure 3. Performance of the proposed DKE&TC system versus SVD with various K values.

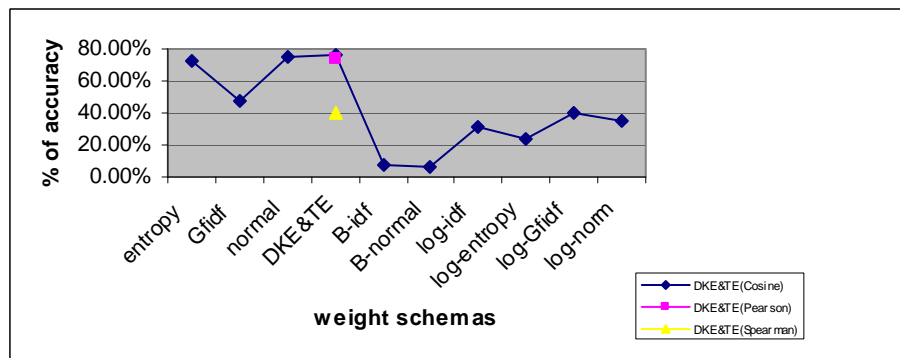


Figure 4. Comparing DKE&TE with various weight schemes and similarity functions.

Table 1. Definitions of various weight schemes.

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

Table 2 shows three examples of document queries and the diagnostic categories classified by the DKE&TC system. The first column lists the three examples of input queries and the right column lists three top-matched diagnostic categories to each input query. The correct categories are highlighted.

Table 2. Examples of matched diagnostic categories classified by the DKE&TC.

Input queries	Three top matched diagnostic categories and associated description
CUST STATES VEHICLE OVERHEAT NO LOSS OF COOLANT BUT HAD BURNING COOLANT SMELL.	C1: COOLANT LEAK C2: ENGINE OVERHEATS/RADIATOR TROUBLES C3: UNUSUAL EXHAUST SYSTEM ODOR
CUSTOMER STATES CHECK FOR OIL LEAK BETWEEN TRANS. AND ENGINE	C1: ENGINE LEAKS OIL C2: UNDETERMINED ENGINE LEAK C3: TRANSMISSION/CLUTCH FLUID LEAKS

SAYS GETTING SQUEAK NOISE FROM BELT OR BEARING TYPE	C1: ENGINE BELT BREAKING/SLIPPING/SQUEALING
	C2: ENGINE BELT SLIPPING/SQUEALING
	C3: ENGINE BELT OFF/FRAYED/COMING APART/BROKEN

4.2 Experimental results of the hierarchical neural network system

Table 3 describes the data used in our experiments. The data are organized in a two-level hierarchy. At the first level there are 7 categories, C_1, \dots, C_7 . The dimensions of these feature spaces are listed. All categories have high dimension feature spaces, the highest one, 911, belonging to category C_2 . The feature space for the entire training data is 2511. Each C_i has its own subcategories, and there are a total of 54 categories at level 2. The training data contains 20,000 documents. A set of test data collected from different customers contains 6,000 documents.

Table 3. Data description.

Category	Feature space dimensions	Level 2 categories
C1	242	3
C2	911	11
C3	536	12
C4	209	3
C5	743	8
C6	461	3
C7	606	14
$C_1 \cup \dots \cup C_7$	2511	54

The categorization accuracies of the hierarchical neural network system on the 6,000 test documents are shown in Figure 5. The figure also lists the performance of two other types of neural networks, a single neural network, and a categorical neural network system [11, 12]. A single neural network can be trained to directly classify any input document into one of the n categories at the second level of the document hierarchy shown in Figure 2. The categorical neural network system has N neural networks, each one trained to represent one of the document categories at level 1 in the document hierarchy (see Figure 2). Each neural network F_i produces a prediction of a category to which document d belong based on its own input vector \bar{x}_i . The output of the N neural networks are sent to a decision module for the final prediction of the category to which d belongs.

Figure 5 lists two types of performance accuracy, TOP 1 and TOP 3. The accuracy on TOP 1 of a neural network system is the percentage of the test documents that are matched correctly with the TOP 1 output of the neural network system. The accuracy on TOP 3 of a neural network system is the percentage of the text documents that have the correct categories within the TOP 3 output of the neural network system. There are two types of performance presented: level 1 and level 2 category matches. For the level 2 categorization, the TOP 1 response from the single neural network has an accuracy of 54.14% and the accuracy for the TOP 3 responses is 69.89%. The categorical neural

network system scored 56.86% accuracy in TOP 1 responses and 61.94% accuracy in TOP 3 responses. The hierarchical system scored 67.25% accuracy in the TOP 1 responses and 84.44% in TOP 3 responses. At the level 1 categorization, the single neural network system has an accuracy of 77.98% in the TOP 1 match, and 90.92% in the TOP 3 matches. The categorical neural network system has an accuracy of 85.32% in the TOP 1 match and 94.84% in the TOP 3 matches. The hierarchical neural network system scored 100% in TOP 1 match, and, of course, in TOP 3 as well. The hierarchical neural network system has the highest accuracies with significant margins in all counts.

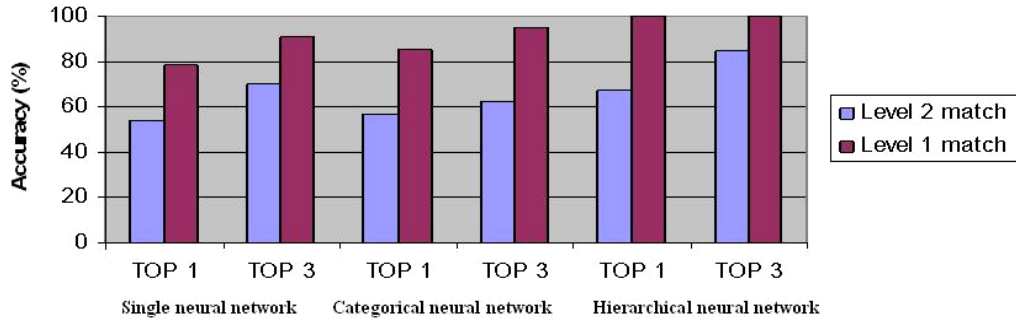


Figure 5. Document categorization accuracy for three neural network systems.

Table 4 shows the computational time consumed by the three neural network systems at both training and test stages. When the feature space of 2511 dimensions was used, the single neural network training was not still completed even after 10 days. The feature space for the single neural network system was reduced to 760 dimensions by applying higher thresholds on the frequencies of terms and phrases extracted from the entire training data Tr^a . When the feature space of 760 dimensions was used, the single neural network still took seven days to train. The individual neural networks within the categorical and hierarchical systems can be trained in parallel on different computers. Thus, the average training time per category is listed. Both systems require significantly less training time than the single neural network. The query time per document is very fast with the categorical neural network system, taking 0.001 seconds per query as the fastest, and the single neural network taking 0.005 seconds as the slowest.

Table 4. Training and test time for the three neural network systems.

	Single neural network (reduced space)	Categorical neural network system	Hierarchical neural network system
Training time	168 hours	20 hours per category average	56 hours total in F; 12 hours per level 2 category average
Query time	0.005 sec	0.001 sec	0.003 sec

5. Conclusion

In this project, we presented two text document mining systems developed for an automotive engineering diagnostic application: DKE&TC and a hierarchical neural network system. We presented

our research results in a number of important areas including various weighting schemes, similarity functions and various neural network systems. Our experiment results show that DKE&TC outperformed the well-known LSA model and many other weight schemes. We also compared the performance of the DKE&TC system with diagnostic engineers on a small data set of 100 queries. The results showed that the DKE&TC system outperformed its human counterpart.

The proposed hierarchical neural network system outperformed the other two neural network systems by a considerable margin in terms of categorization accuracy. Although the categorical neural network system gave the fastest online query response time (0.001 seconds/query), the hierarchical neural network system came in at 0.003 seconds/query, which is still very fast. The training for both the categorical and the hierarchical systems are more efficient than the single neural network system, particularly when more computers are used to simultaneously train the individual neural networks involved in the system.

In conclusion, this project, funded by the Henry W. Patton Center for Engineering Education and Practice (HP-CEEP), has been a success. We have published three conference papers, which are listed below, and participated in a competition sponsored by the National Institute for Standards in Technology (NIST). We have demonstrated our systems to various organizations inside Ford Motor Company, and are currently negotiating a research contract with Ford/Volvo division.

Conference publications:

Zhihang Chen, Chengwen Ni and Yi L. Murphey, "Neural Network Approaches for Text Document Categorization," IEEE International Joint Conference on Neural Networks, July, 2006.

LiPing Huang and Yi L. Murphey, "Text Mining with Application to Engineering Diagnostics," The 19th International conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Annecy, France, June, 2006

LiPing Huang, ZhiHang Chen, and Yi Lu Murphey , "UMD at TREC 2005: Genomics Track," NIST TREK conference, Nov. 2005

6. Impact

Educational Impact

This grant supported one graduate student, LiPing Huang, who developed algorithms including TCF, SVD, and various similarity functions. She conducted many experiments and contributed extensively in the publications listed above.

This project also partially supported another graduate student, Chengwen Ni, who conducted all the experiments related to the neural network systems and contributed to the conference papers listed above.

Much of the research in this project has been used in the PI's course work for the graduate-level class ECE 537/CIS 568 Data Mining. The PI directed four students' projects whose topics are related to this project:

- Jeffrey Showiak, who implemented algorithms to classify text documents based on the authors;
- Elizabeth Hershey and William Serlin, who did an in-depth study in text mining relating to web blogs, and used a blog-mining tool to mine blog documents; and
- Anju Kapoor, who developed and implemented an algorithm to mine résumé documents.

Industrial Impact

In industry there is abundant text information available in human language description form that contains expert engineering knowledge, marketing information, and consumer satisfaction. This project is driven by the urgent need of automatic text mining tools to extract knowledge from ill-structured documents. The systems developed under this HP-CEEP grant have been demonstrated to a number of units under the Ford Motor Company. We have submitted two proposals to the Ford Motor Company for further development of this research. Currently, we are negotiating a research contract with Ford/Volvo. Other units in Ford have contacted us for information about this project, and we are optimistic about future funding.

7. Acknowledgments

This work is supported in part by a grant from HP-CEEP in the College of Engineering and Computer Science at the University of Michigan-Dearborn, and by Ford Motor Company.

8. References

- [1] V. Raghavan and S.K.M.Wong. "A Critical Analysis of Vector Space Model for Information Retrieval." *Journal of the American Society for Information Science* 37 (1986): 279-87.
- [2] G. Salton and C. Buckley. "Term weighting approaches in automatic text retrieval." *Information Processing and Management* 24 (1988): 513-23.
- [3] S.T. Dumais. "Enhancing performance in latent semantic indexing (LSI) retrieval." *Technical Report Technical Memorandum*. Bellcore: 1990.
- [4] M.W. Berry, S.T. Dumais, and G.W. O'Brien. "Using linear algebra for intelligent information retrieval." *SIAM Review* 37 (1995): 573-95.
- [5] S. Deerwester, G. Furnas, T. Landauer, and R. Harshman. "Indexing by Latent Semantic Analysis." *Journal of the American Society for Information Science* 41 (1990): 391-407.

- [6] T. K. Landauer and D. Laham. "Introduction to latent semantic analysis." *Discourse Processes* 25 (1998): 259-84.
- [7] M.W. Berry and E.R. Jessup. "Matrices, Vector Spaces, and Information Retrieval." *SIAM Review* 41 (1999): 335-62.
- [8] A.L. Edwards. *The Correlation Coefficient*. San Francisco: W. H. Freeman, 1976.
- [9] E.L. Lehmann and H.J.M. D'Abbrera. *Nonparametrics: Statistical Methods Based on Ranks*. Ed. E. Cliffs. New Jersey: Prentice-Hall, 1998.
- [10] K. Nigam, A. McCallum, S. Thrun and T. Mitchell. "Text Classification from Labeled and Unlabeled Documents using EM." *Machine Learning* 39 (2000): 103-34.
- [11] J. Farkas. "Generating Document Clusters Using Thesauri and Neural Networks." *Second Volume of the Proceedings of the Canadian Conference on Electrical and Computer Engineering*, 1994: 710-13.
- [12] O. Guobing, Y.L. Murphey, and L. Feldkamp. "Multiclass Pattern Classification Using Neural Networks." *International Conference on Pattern Recognition*, 2004: Cambridge, UK.