Applied Intelligence 23, 277–294, 2005

A Hierarchical Neural Network Document Classifier with Linguistic Feature Selection

CHIH-MING CHEN
Graduate Institute of Learning Technology, National Hualien University of Education, 123 Hua-His Rd., Hualien 970, Taiwan, Republic of China
cmchen@mail.nhlue.edu.tw

HAHN-MING LEE AND CHENG-WEI HWANG
Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, 43 Sec. 4, Keelung Rd., Taipei 106, Taiwan, Republic of China

Abstract. In this article, a neural network document classifier with linguistic feature selection and multi-category output is presented. It consists of a feature selection unit and a hierarchical neural network classification unit. In the feature selection unit, the candidate terms are extracted from some original documents by text processing techniques, and then the conformity and uniformity of each term are analyzed by an entropy function which can measure the significance of terms. Terms with high significance are selected as input features for training neural network document classifiers. In order to reduce the input dimensions, a composition mechanism of fuzzy relation is employed to identify synonyms. By this method, a synonym thesaurus can be constructed to reduce input dimensions. To simplify the learning scheme, the well-known back-propagation learning model is used to build proper hierarchical classification units. In our experiments, a product description database from an electronic commercial company is employed. The experimental results show that this classifier achieves sufficient accuracy to help human classification. It can save much manpower and work time classifying a large database.

Keywords: information retrieval, hierarchical document classifier, back-propagation neural network, feature selection

1. Introduction

In two of the most important Internet applications, search engines and electronic catalogs [1], documents are presented in a natural language. Undoubtedly, World Wide Web (WWW) directory maintained by search engine sites and electronic catalogs used by commercial sites with hierarchical document taxonomy are two kinds of popular information representation methods on the Internet. The number of Web pages and commercial sites is increasing exponentially so automatically classifying linguistic, descriptive documents is an important research topic. To analyze the category of a document, the document classification system should include a linguistic word analyzer and an effective classifier.

In a hierarchical document classification problem, the sub-categories of a category are called sibling categories. Generally, each classifier of hierarchical classification tries to choose an appropriate sub-category for the uncategorized documents. However, the categories that are far from the root category are similar in a hierarchical directory category, because the upper categories in a hierarchical directory constitute a broad and rough document taxonomy, but the lower categories constitute a fine document taxonomy. That is, documents in the lower categories are similar to each other, so the documents in the lower categories are more difficult to distinguish using a linear classifier than are those in the upper categories. Therefore, to solve hierarchical document classification problems, a non-linear classifier is needed specially on the categories which are far away from the root category. In past years, several hierarchical classification methodologies based on various classifiers and feature selection approaches were proposed to solve document classification problems with hierarchical document taxonomy [2–6]. However, most approaches focus on using similarity measures to construct a hierarchical classification system. In this study, a hierarchical multilayer Artificial Neural Network (ANN) with Information Retrieval techniques for feature selection is presented to construct an automatic document classification system. Although the ANN has been widely applied in artificial intelligence (e.g., classification and pattern recognition) [7, 8], new challenges have arisen with the trend of intelligent Web development [9]. That is because World Wide Web is potentially a huge-scale dataset and may consist of tens of thousands of features. Therefore, it is difficult to design an effective classifier using an ANN for Web page classification. However, Information Retrieval (IR) systems [10] have become popular because of the variety of information services on the Internet [11, 12]. In the proposed document classification system, the IR technique is used to extract useful information for document representation and feature selection. In order to design an effective classification system for linguistic description documents, the proposed classification system combines ANN with IR techniques. Actually, the proposed method can be employed as an enhanced hybrid neural system [13, 14] to improve the performance of document classification.

In IR systems, the text processing techniques [15] are usually used to analyze the word structure. In this study, two well-known text processing approaches, stop-list elimination [16] and stemming rules [17], are used to process all words in the solved database before performing classification. A stop-list is a list of stop-words that occur frequently, excluding insignificant words in English (such as “the”, “of”, “and”, “to”, etc.). Eliminating such words increases the index processing speed and saves large amounts of space in the indexes. In our IR system, a particularly well-established stop-list from the Brown corpus [18] is applied for text processing. Stemming rule techniques improve IR performance by analyzing the morphological variants of words. It extracts the stems of words, so that a stem can stand for all its morphological variants. For example, “engineering”, “engineered” and “engineer” can all be stemmed to “engineer”. Hence, stemming rules combine words to improve retrieval effectiveness and reduce the size of indexing files. Porter [17] proposed an algorithm for stemming in 1980, and his stemming rule is employed in our system.

Reducing the feature dimensions in a large-scale document classification system is essential. Among the available feature dimensionality-reduction methods, Shannon [19] applied an entropy function to measure the significance of information. Yang et al. and Cheng et al. [9, 20] both applied the entropy function to measure words’ significance by computing their conformity and uniformity [1, 15]. Conformity is the extent to which the occurrence of a term is concentrated in documents belonging to some categories, as opposed to being spread across most categories. Uniformity is the extent to which a term is widespread in the documents of a given category, rather than only being concentrated in a few documents.

In this article, the conformity and the uniformity of each term are evaluated, and then used to measure the significance of the terms. Terms with high significance will be selected as input features for the proposed hierarchical neural network document classifiers. Furthermore, to further reduce the input dimensions, a composition mechanism of fuzzy relation is employed to merge synonyms. Based on the uniformity analysis, a term similarity matrix is obtained by the fuzzy relation operation. By this method, we can construct a synonym thesaurus to reduce input dimensionality. The experimental results reveal that this classification system is sufficiently accurate to support the document classification by humans. It can save much time and labor classifying a large database.

2. Problem Description

In this article, a product database built by All Products Online Corporation, an electronic commerce company (http://www.allproducts.com/), is used as data sets of training and testing documents. Each document records the attributes and information of a product. All descriptions are presented as written English, except for a small amount of data presented in numeric or others forms. These products have been manually classified according to their descriptions. Each product is assigned to one or more categories. Those categories that are assigned to the same product are strongly related to each other.

To simplify the complexity of our research, numbers and non-textual information (such as numeric data, symbols, notations and ASCII drawings) are ignored. The properties of the input and output in the problem domain are described below:

(1) Input (product description documents)

- Written in natural language (English).
- Information has been divided into fields, such as product name, product specification, etc.
- The mapping between a product and the set of categories is one-to-many.

(2) Output (categories)

- Categories are arranged in a multi-level structure. Parent categories are major categories whose representations are broad and rough. Categories that directly succeed a parent category are called child categories. A parent category may extend to many child categories in the next level for more precise definition. Thus, parent categories and child categories are connected with high similarity. Furthermore, if a parent category includes a product set **P** and its n child categories include product sets $\mathbf{P}_1$,$\mathbf{P}_2$... and $\mathbf{P}_n$, then **P** is the union of $\mathbf{P}_1$,$\mathbf{P}_2$... and $\mathbf{P}_n$. Note that the above parent-child structure may be extended to any number of levels, but we only use three levels of categories in testing the performance of our system in order to simplify the analysis.
- A child category may be derived from more than one parent category in a hierarchical directory category.

3. System Architecture

In order to solve the document classification problem described in the previous section, a hierarchical neural network classification system with linguistic feature selection is proposed. The basic unit of this classification system is a 3-layer back-propagation (BP) learning model [21]. The system architecture is shown in Fig. 1. It consists of a feature selection unit and a hierarchical neural network classification unit.

In the feature selection unit, textual terms can be obtained from the original linguistic description documents via the text processing techniques described previously: stop-list elimination and stemming rules.

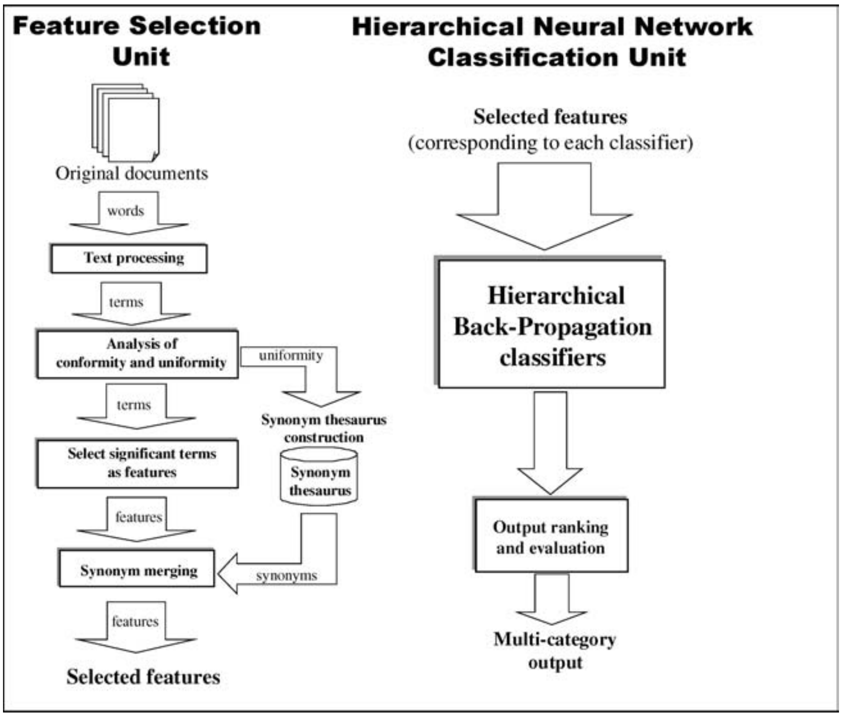

Feature Selection Unit
Hierarchical Neural Network Classification Unit
Original documents
words
Text processing
terms
Analysis of conformity and uniformity
uniformity
terms
Synonym thesaurus construction
Select significant terms as features
Synonym thesaurus
features
Synonym merging
synonyms
features
Selected features
Selected features
(corresponding to each classifier)
Hierarchical Back-Propagation classifiers
Output ranking and evaluation
Multi-category output

Figure 1. System architecture.

Each word that passes through text processing is a named term. After that, each term's conformity and uniformity relative to each category are measured to determine whether the term is significant. Significant terms are treated as features for BP classifiers. To further reduce the number of features, the fuzzy relations [22] are applied to construct a concise synonym thesaurus and then merge these synonyms according to the uniformity measure.

The hierarchical neural network classification unit proposed by this study is a 3-level hierarchical structure, and the basic unit in each level is a 3-layer BP learning model [21]. Each BP model is employed to classify a particular category. The classification unit is multi-category output, and the output results are ranked. The classification results from a higher level BP classifier are propagated to the next level BPs for further classification. We also rank and evaluate outputs to determine those outputs that are most desired.

3.1. Feature Selection Unit

In the feature selection phase, text processing is first used to extract terms from original documents. Next, the conformity and uniformity of each term are calculated, and then select significant terms through a simple but effective feature selection method. To reduce input dimensionality, a synonym merging process is applied. The feature selection operation is shown in Fig. 2, and its details will be described in the following subsections.

3.1.1. Text Processing. The purpose of text processing is to remove non-textual words (e.g., numeric data, symbols, notation and ASCII drawings) and stop-words from the original documents, and then transfer the remaining words to stem words, by applying a stemming rule.

The first phase of text processing filters out non-textual words, as mentioned above. Also stop-words that do not contain any usable information (such as pronouns, prepositions, etc.) are eliminated [10]. A collection of stop-words is called a stop-list. In this study, we use the stop-list from the Brown corpus [18].

The second phase of text processing is word stemming. Porter [17] proposed a stemming rule to extract the stem of a word based on the word's prefix and suffix. Hence, words with different morphological features but with basically the same meaning can be represented by a single stem. After stemming, words are mapped to a stem domain and referred to as terms.

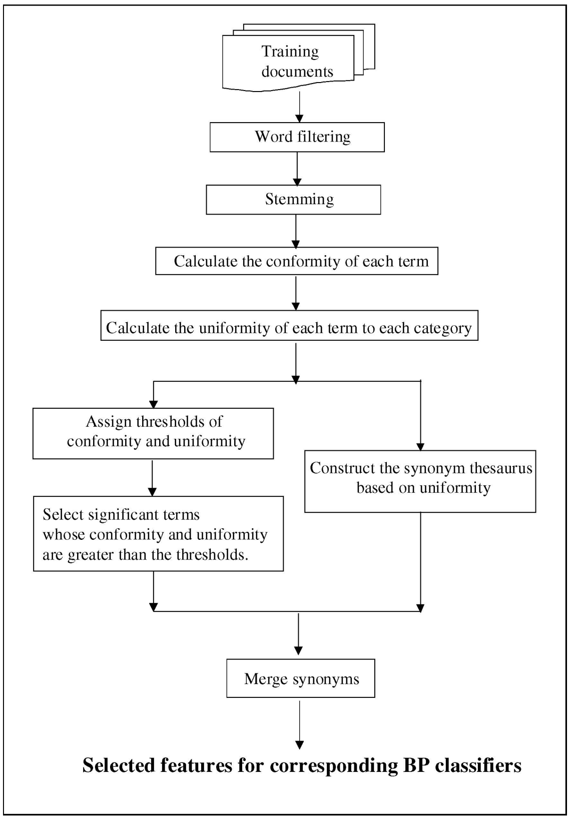

Training documents
Word filtering
Stemming
Calculate the conformity of each term
Calculate the uniformity of each term to each category
Assign thresholds of conformity and uniformity
Construct the synonym thesaurus based on uniformity
Select significant terms whose conformity and uniformity are greater than the thresholds.
Merge synonyms
Selected features for corresponding BP classifiers

Figure 2. Feature selection flowchart.

3.1.2. Conformity and Uniformity. Huang [23] mentioned two factors in influencing the accuracy of classification system: conformity and uniformity [15, 23]. He defined conformity and uniformity by applying the entropy function [19] which has been used to measure signal-noise ratios in the communications field. The definitions of conformity and uniformity are as follows:

(1) **Conformity**
It is difficult to distinguish between categories if we use a general term as a feature. Therefore, a significant term should occur in documents that belong to some categories but not be spread across most categories. This kind of concentration among categories is known as conformity. Conformity is sometimes measured in terms of ICF (Inverted conformity frequency), in which case it is defined as

follows:

$$
\begin{aligned}
ICF_j &= -\sum_{i=1}^{n} p_{ij} \log p_{ij} \\
p_{ij} &= \frac{d_{ij}}{\sum_{i=1}^{n} d_{ij}}
\end{aligned}
\tag{1}
$$

where d_{ij} is the document frequency of the term j in the ith category, p_{ij} is the probability that term j occurs in category i, and n is the number of categories.

(2) **Uniformity**

For a term is to be meaningful for a category, it should be relatively uniform. That is, it should occur in many of the documents belonging to the category, rather than be concentrated in a few documents of this category. The uniformity of term j in the category i can be measured as:

$$
\begin{aligned}
U_{ji} &= -\sum_{k=1}^{n} q_{kj} \log q_{kj} \\
q_{kj} &= \frac{tf_{kj}}{\sum_{k=1}^{n} tf_{kj}}
\end{aligned}
\tag{2}
$$

where q_{kj} is the probability that term j occurs in document k of the ith category, tf_{kj} is the term frequency of term j in document k, and n is the number of categories.

In this research, the conformity and uniformity are applied to measure the significance of terms obtained from linguistic documents and then select those with high significance. The first phase of selecting significant terms is to select terms whose conformity (ICF value) surpasses an assigned threshold. We call the selected set of significant terms as the ICF-qualified term set. In this way, terms are excluded if their occurrence is widespread across categories.

The second phase in selecting significant terms is to examine each term in the ICF-qualified term set to see if it is significant in one or more categories. For a given category, a term will be rejected if it does not appear in most documents of the category. Terms whose uniformity (U) in a category surpasses an assigned threshold are called U-qualified terms. How to determine the thresholds for both conformity and uniformity is a trade-off issue. Namely, these two threshold values might affect the number of the selected input feature terms, thus producing different classification accuracy rates and training speeds. Using the proposed feature selection method, if a candidate term can be selected as a significant feature term, then its conformity value must be greater than an assigned threshold value and its uniformity value must be less than an assigned threshold value. In general, a higher threshold value of conformity can effectively reduce input feature terms and promote training speed, but it may lead to a lower classification accuracy rate because it ignores many significant feature terms. On the other hand, a lower threshold value of uniformity will select many general terms as feature terms because it cannot identify significant feature terms well. Therefore, these two threshold values indeed are very difficult to be appropriately determined based on a crisp rule. In particular, the threshold parameters can be optimized using the genetic algorithm [24] or line search algorithms [25]. However, to optimize the two threshold parameters is a time-consumed job for the proposed hierarchical BP classifier because the classification accuracy rate must be repeatedly evaluated to test various combinations of threshold parameters. In order to appropriately determine the number of feature terms, later experimental results will illustrate a suggested heuristic criterion which can obtain a satisfied classification rate in our tested dataset.

3.1.3. Synonym Thesaurus. Consider two terms whose significance in each category is alike, they probably represent an identical meaning and we can regard them as synonyms [10, 15]. The above-mentioned "significance" in each category can be considered as the term's uniformity in each category as introduced in Section 3.1.2. Thus, based on the uniformity of terms, similar terms can be clustered to build a synonym thesaurus. In our study, the fuzzy relation [22] is applied to measure the similarity of two feature terms based on uniformity. Assuming that V and W are two collections of objects, an arbitrary fuzzy set R, defined in the Cartesian product $V \times W$, will be called a fuzzy relation in the space $V \times W$. The fuzzy relation R can be represented as follows:

$$
R : V \times W \to [0, 1] \tag{3}
$$

where R is a fuzzy set defined in the space $V \times W$, which takes values from the interval [0, 1].

So far we have considered operations on fuzzy relations defined in the same space as the Cartesian product of two collections of objects. In order to reduce the input feature terms for document classification, the composition of two fuzzy relations is applied to infer terms' similarities based on uniformity. Assume R_1 be a fuzzy relation in $U \times V$ and R_2 a fuzzy relation in $V \times W$. Generally, Max-Min and Max-Star compositions are frequently used two ways of composing such relations.

The fuzzy relation can extend the crisp relation concept to allow for various degrees of interaction between elements; accordingly, elements are related to some extent or unrelated. Thus, two fuzzy relations based on uniformity are modeled: the set of uniformities for a given term in each category, and that for all terms in each category. To measure terms' similarities, a composition of the above two fuzzy relations is needed. That is, we need to combine two fuzzy relations into a new fuzzy relation. Max-Min composition, which is one of several composition operators for fuzzy relation [22], is used in this work.

In what follows, we describe the composition of the new fuzzy relation, which is used to measure the similarity among terms using uniformity:

Assume that there are n categories and m terms, then a composition of fuzzy relations denoted by "o" for a term can be defined as:

$$\mathbf{A}_j \, \mathrm{o} \, \mathbf{U} = \mathbf{R}_j \tag{4}$$

where $\mathbf{A}_j$ is a $1 \times n$ matrix whose elements u_{ij} are the uniformities of term j to the ith category, $\mathbf{U}$ is an $n \times m$ matrix comprising the uniformities of terms across categories, and $\mathbf{R}_j$ is a $1 \times m$ matrix containing the similarities between term j and other terms. That is,

$$\begin{array}{c} A_j \\ \begin{array}{ccccccc} \mathbf{c}1 & \mathbf{c}2 & \mathbf{c}3 & .. & \mathbf{c}i & .. & \mathbf{c}n \\ {[u1j} & u2j & u3j & .. & uij & .. & unj]_{1\times n} \end{array} \end{array} \circ \begin{array}{c} \mathbf{U} \\ \begin{array}{c|ccccccc|} & \mathbf{t}1 & \mathbf{t}2 & \mathbf{t}3 & .. & \mathbf{t}j & .. & \mathbf{t}m \\ \mathbf{c}1 & u11 & u12 & u13 & .. & .. & .. & u1m \\ \mathbf{c}2 & u21 & u22 & u23 & .. & .. & .. & u2m \\ \vdots & & & & & \vdots & & \\ \mathbf{c}i & & & & & uij & & \\ \vdots & & & & & \vdots & & \\ \mathbf{c}n & un1 & un2 & un3 & .. & .. & .. & unm \end{array}_{n\times m} \end{array}$$

$$\begin{array}{c} \mathbf{R}_j \\ \begin{array}{cccccccc} & \mathbf{t}1 & \mathbf{t}2 & \mathbf{t}3 & .. & \mathbf{t}k & .. & \mathbf{t}m \\ = & {[s1j} & s2j & s3j & .. & skj & .. & smj]_{1\times m} \end{array} \end{array} \tag{5}$$

where $u_{ij}$is the uniformity of term$j$ in category$i$; s_{kj} is the similarity of term j to term k; and $\mathbf{t}_j$, $\mathbf{t}_k$ and $\mathbf{c}_i$ indicate the jth term, kth term and ith category, respectively.

The result of this composition is the matrix R_j for term$_j$, and the elements of this matrix represent the similarities between term$_j$ and other terms. After completing the fuzzy composition for all terms, the similarities between every two terms can be measured. We can combine m 1×m matrices R_j into an m×m matrix R called the term similarity matrix, and shown as follows:

$$R = \begin{array}{c|ccccccc|} & t1 & t2 & t3 & .. & t_j & .. & tm \\ t1 & s11 & s12 & s13 & .. & .. & .. & s1m \\ t2 & s21 & s22 & s23 & .. & .. & .. & s2m \\ t3 & s31 & s32 & s33 & .. & .. & .. & s3m \\ \vdots & & & & \vdots & & & \\ t_i & & & & \vdots & s_{ij} & & \\ \vdots & & & & \vdots & & & \\ tm & sm1 & sm2 & sm3 & .. & .. & .. & smm \end{array}_{m\times m} \tag{6}$$

The term similarity matrix R, obtained by the fuzzy relation operation, is a symmetrical m×m matrix because the element s_{ij} equals the element s_{ji} in matrix R. Next, a threshold for the elements is assigned in the term similarity matrix. Terms will be grouped together if their similarities exceed the assigned threshold. All terms in a group can be considered as synonyms; as a result, a synonym thesaurus is accomplished. Note that the synonym thesaurus is built by analyzing the similarities of terms based on the uniformity of these terms in categories. Terms are merged only because they are significant in the same categories; their precise meanings may not be identical.

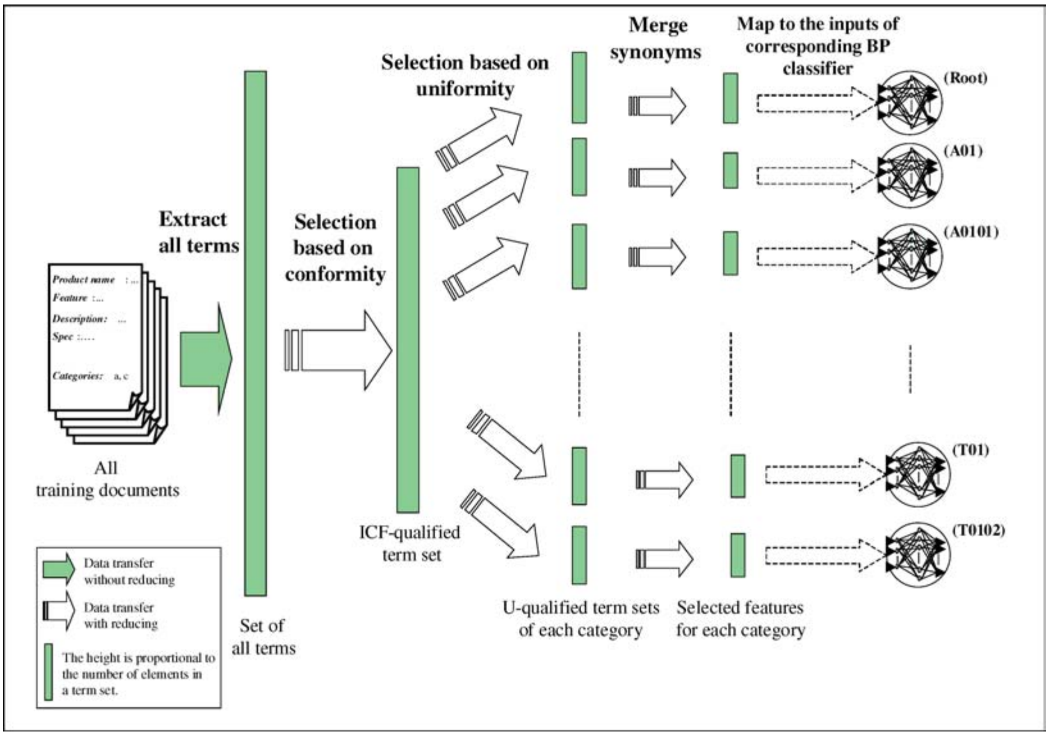

Product name : ...
Feature :...
Description: ...
Spec :....
Categories: a, c
All training documents
Extract all terms
Set of all terms
Selection based on conformity
ICF-qualified term set
Selection based on uniformity
U-qualified term sets of each category
Merge synonyms
Selected features for each category
Map to the inputs of corresponding BP classifier
(Root)
(A01)
(A0101)
(T01)
(T0102)
Data transfer without reducing
Data transfer with reducing
The height is proportional to the number of elements in a term set.

Figure 3. Feature selection for a corresponding BP classifier.

3.1.4. Selecting Features and Linguistic Term Transformation. Feature selection determines input features for each BP classifier in the hierarchical neural network classification unit. Figure 3 illustrates the flow in selecting features for a corresponding BP classifier. The detailed steps are:

Step 1. Extract terms from all training documents by text processing techniques.

Step 2. Select significant terms whose conformity exceeds the conformity threshold to form the ICF-qualified term set.

Step 3. From the ICF-qualified term set, the significant terms for each category are further selected based on uniformity. If the uniformity of a term in a category surpasses the uniformity threshold, it is selected and all such terms for each category form a U-qualified term set. These terms are significant for this category.

Step 4. Use the synonym thesaurus to merge synonyms in each U-qualified term set. The final selected features for each category are the corresponding input of BP classifier.

During the training and testing phases, linguistic terms need to be transformed into numeric input vectors. Figure 4 illustrates how a document is transformed from linguistic terms into numeric input vectors. The steps followed in this transformation are:

Step 1. Based on the U-qualified term sets mentioned above, candidate terms are extracted from input documents. For each extracted term, the Term Frequency (Tf) which indicates how many times a term appears in a document [15], can be calculated.

Step 2. Add up Tfs if two terms are synonyms.

Step 3. Tf vectors are delivered to a corresponding BP classifier for training or testing.

3.2. *Hierarchical Neural Network Classification Unit*

3.2.1. Hierarchical Classifiers. The classified categories are represented as a tree structure. We assign a code to each category and the code involves hierarchical categories. The collection of all codes is referred to as the code book, part of which is shown in Fig. 5(a). In this figure, categories are divided into three levels.

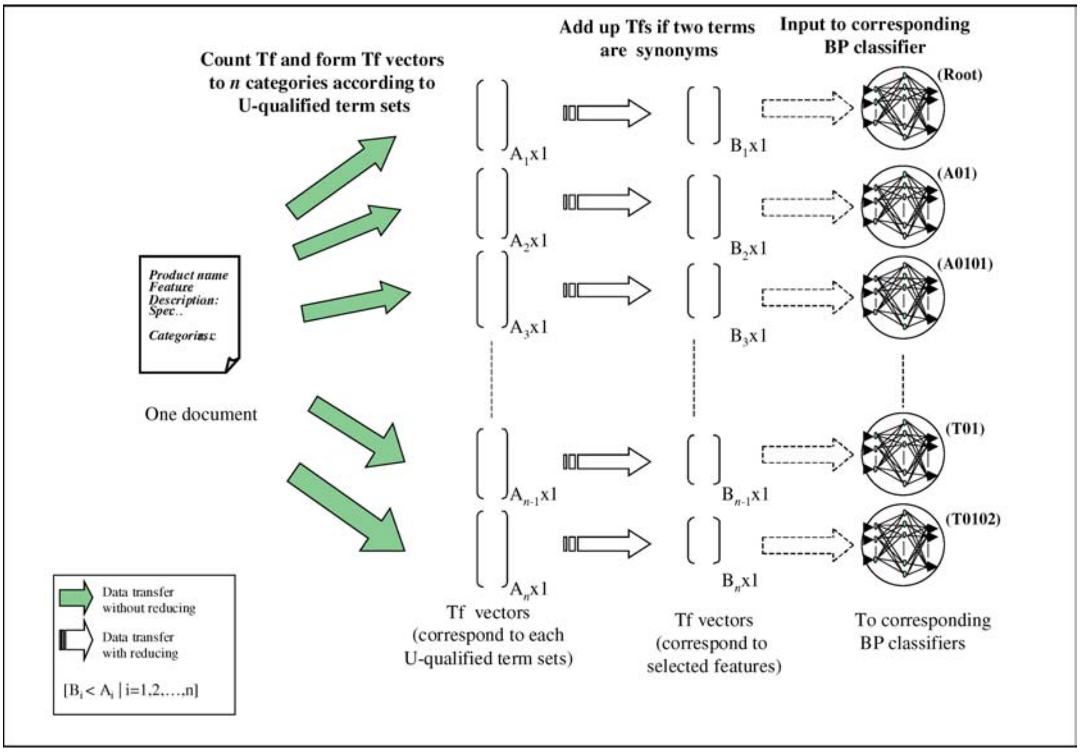

Count Tf and form Tf vectors to n categories according to U-qualified term sets
Add up Tfs if two terms are synonyms
Input to corresponding BP classifier
Product name
Feature
Description:
Spec..
Categorias:
One document
A1x1
A2x1
A3x1
An-1x1
Anx1
B1x1
B2x1
B3x1
Bn-1x1
Bnx1
(Root)
(A01)
(A0101)
(T01)
(T0102)
Tf vectors (correspond to each U-qualified term sets)
Tf vectors (correspond to selected features)
To corresponding BP classifiers
Data transfer without reducing
Data transfer with reducing
[Bi < Ai | i=1,2,…,n]

Figure 4. The transformation of linguistic terms into numeric input vectors.

Every category in level 1 or 2 is a parent category which branches into child categories in the next level. Child categories inherit the properties of the parent category; in addition, they have their own properties. The prefixes of child categories' codes are the same as their parent category's code, so if a parent category's code is 'C01', say, then its child categories' codes will be 'C0101', 'C0102'. . . , etc.

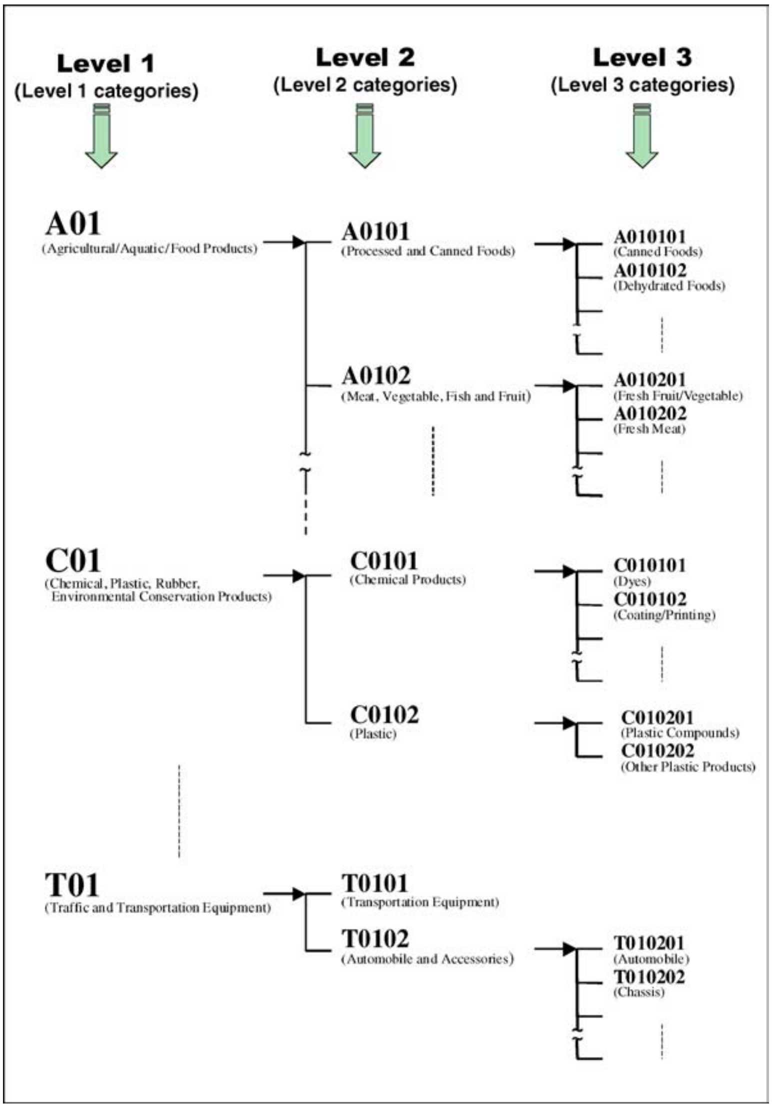
Figure 5(b) shows 3-level hierarchical classification unit with BP classifiers which corresponds to the structure of code book. Each BP classifier stands for a parent category. Its input is the features chosen in the feature selection process, and its output is the child categories that this parent category branches into. A virtual code "Root" is assigned as the level 1 categories' parent category. Initially, every training or testing document is classified into level 1 categories by a BP classifier that represents the code "Root" in level 1. The results from level 1 then trigger the BP classifiers in level 2, and these BP classifiers perform a further classification and yield a third layer of categories. For instance, say a training document is first classified into level 1 categories 'C01' and 'M01' in level 1. Then BP classifiers which represent 'C01' and 'M01' in level 2 are triggered and assume their output categories are 'C0102', 'C0103' and 'M0101'. Likewise, the results from level 2 trigger the BP classifiers in level 3 and those classifiers give the output categories.

3.2.2. Back-Propagation Classifier. In this study, each BP classifier is a 3-layer feed-forward neural network structure. The three layers are the input layer, the output layer and a single hidden layer (shown in Fig. 5(a)). As we mentioned above, each BP is responsible for classifying the categories that succeed from a single parent category. The number of input nodes is determined by feature selection and the number of output nodes is determined by the number of output categories. We set the node number of the hidden layer to equal the average of node number of input and output layers for various hidden layers by the heuristic [26]. Our experiments demonstrate that a single hidden layer is sufficient to recognize the interactions between input features and output categories. More hidden layers would complicate the classification process, and make it hard for the learning procedure to converge [8, 26].

Actually, experimental results given later also show that the average performance of the proposed hierarchical BP neural network classifier with a single hidden layer is superior to the proposed neural network's architecture with two hidden layers in our tested dataset.

4. Experimental Results

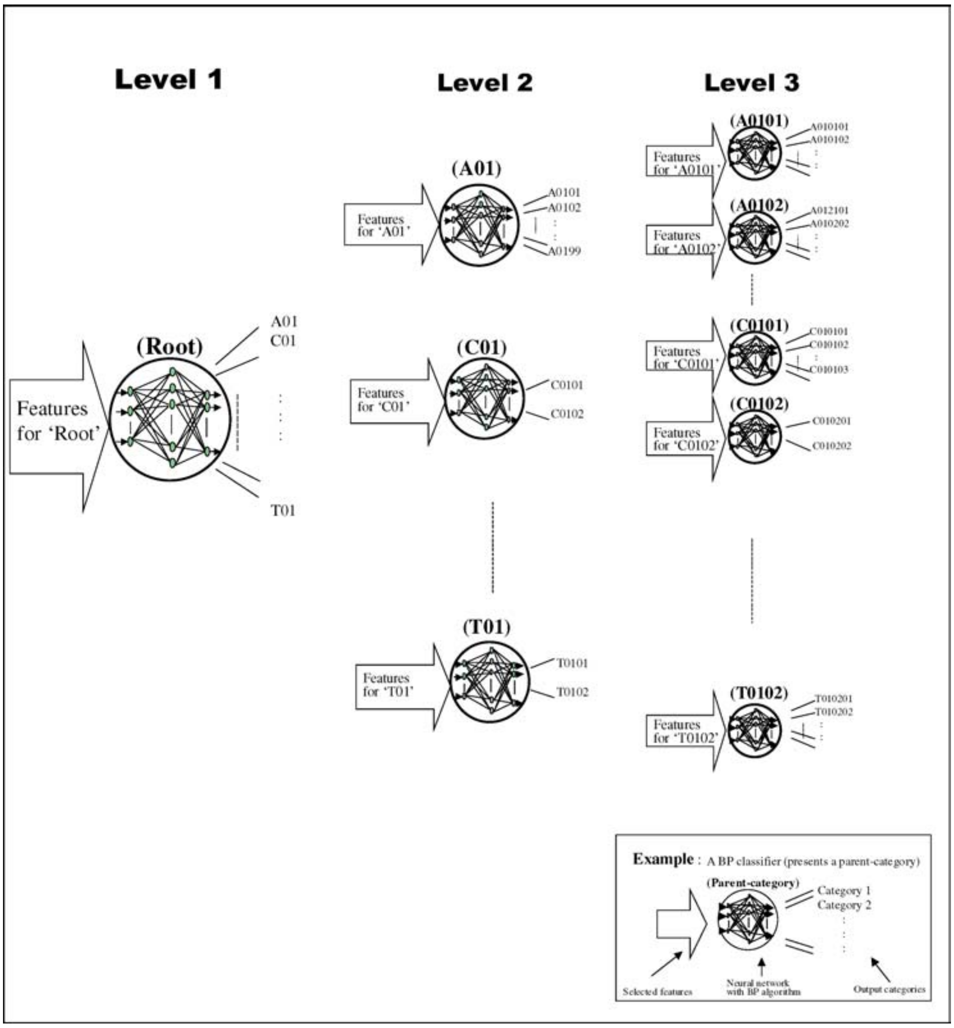
In our experiments, training and testing data sets come from All Products Online Corporation (www.allproducts.com). In this database, a document corresponds to a product and describes the product's properties. The description of a product is organized into fields, such as product name, specification, main features and keywords for indexing. Also, each product is manually classified into one or more categories according to the code book. In our experiments, two data sets with different amounts of documents are used to evaluate the performance of the proposed classification system. Set 1 contains 500 training documents



(a)

Figure 5. (a) The structure of the 3-level code book. (b) Hierarchical neural network classification unit with BP classifiers corresponding to the 3-level code book in Fig. 5(a)

(*Continued on next page.*)



(b)

Figure 5. (*Continued*).

and 500 testing documents; set 2 contains 3000 training documents and 3000 testing documents. The documents in set 1 are mapped to 9 levels and 1 category, 39 levels and 2 categories and 128 levels and 3 categories. In set 2, the hierarchy of categories is extended and contains 22 levels and 1 category, 69 levels and 2 categories, and 313 levels and 3 categories. Note that set 2 includes the documents from set 1.

4.1. Dimensionality Reduction by Feature Selection

In this section, the experimental result of dimensionality reduction by feature selection process is presented. More than three thousand words are extracted from 500 training documents in set 1 and more than seven thousand words are extracted from 3000 training documents in set 2. After text processing, many non-textural "words" are removed. We then select significant terms according to the measure of conformity (ICF) and uniformity (U). Therefore, almost 1000 terms and over 3000 terms are eliminated from sets 1 and 2, respectively. Table 1 shows the input dimensions before and after feature selection. We assign the thresholds of ICF and U as a proportion of $\mathrm{ICF}_{\max}$ and $U_{\max}$, where $\mathrm{ICF}_{\max}$ is the maximum ICF of all the terms and $U_{\max}$ is the maximum U among terms that belong to the same category. For both sets of documents, we find that the criterions of the two threshold values are determined as "less than 85% $\mathrm{ICF}_{\max}$" and "greater than

Table 1. Dimensionality reduction by feature selection.

	1	2
Testing set	(500 documents)	(3000 documents)
Original words	3013	7123
Original terms (after text processing)	2469	5390
Seleted terms (after selection by ICF and U)	1528	1854
Selected features (after synonym merging)	**1360**	**1722**

Threshold of ICF : <85% $\mathrm{ICF}_{\max}$.
Thresholds of U : >25% $U_{\max}$.
Threshold of similarity (S) : >25% $\mathrm{S}_{\max}$
Number of synonym groups : 500 documents → 14 groups, 3000 documents → 27 groups.

25% $U_{\max}$" which can appropriately identify significant feature terms and obtain a satisfactory classification accuracy rate.

In the next step, synonyms are merged. Similar to the selection of ICF and U, we heuristically assign a proportion of the maximum similarity ($S_{\max}$) as the threshold for synonym merging. We use "greater than 25% $S_{\max}$", so terms are merged if their similarities are greater than 25% $S_{\max}$. After synonym merging, some synonym groups are produced and the number of selected features falls to 1360 in document set 1 and 1722 in document set 2.

4.2. *Accuracy Measurements*

In the study, two measures, i.e. precision and recall rates, are applied to evaluate the performance of document classification for each classified document, and defined as follows:

$$\text{Precision Rate} = \frac{N_{\text{correct}}}{N_{\text{actual}}} \tag{7}$$

$$\text{Recall Rate} = \frac{N_{\text{correct}}}{N_{\text{desired}}} \tag{8}$$

where N_{correct} is the number of elements of output category set that are correctly categorized for the classified documents, N_{actual} is the number of elements of actual output category set for the classified documents, and N_{desired} is the number of elements of desired output category set for classified documents.

Since the proposed hierarchical BP classifier must handle multi-category output, this study uses the desired output category set and actual output category set to show various classification results. The desired output category set is the set of categories into which a document should be correctly classified. The actual output category set is the set of categories into which a document is actually classified by the proposed hierarchical BP classifier. In the tested electronic catalogs classification problem, both desired and actual output category sets for each document generally are not to be classified into a single category but a set of categories. Figure 6 illustrates the relationship between the desired output category set and the actual output category set. In this figure, we summarize five possible categorized situations, i.e. exact-match, coverage-match, subset-match, overlap-match, and no-match. These situations are described below:

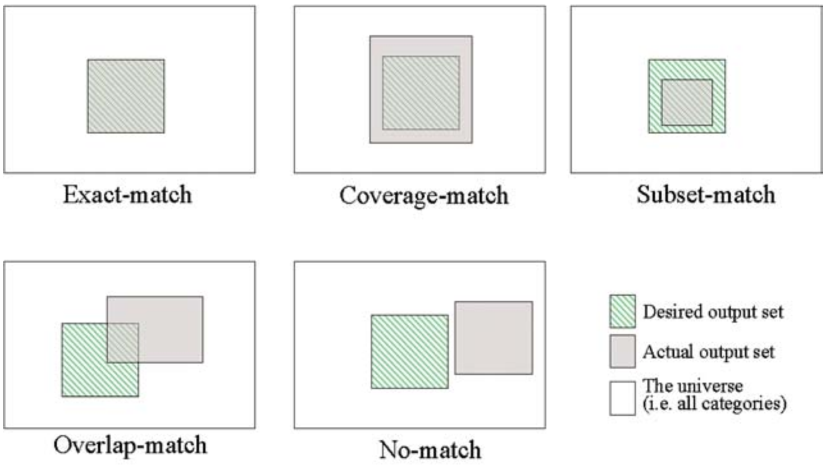


Figure 6. Five categorized situations showing the relationship between the actual output category set and the desired output category set.

- *Exact-match*
  The actual output category set of the classified documents is exactly same with the desired output category set. That is, all categories in the actual output category set equal the categories in the desired output category set.
- *Coverage-match*
  The number of categories in the actual output category set is more than that of the desired output category set, and the desired output category set is included in the actual output category set.
- *Subset-match*
  The number of categories in the actual output category set is less than that of the desired output category set, and the actual output category set belongs to desired output category set.
- *Overlap-match*
  Part of actual output category is identical to the desired output category set. Some categories in the actual output category set also belong to the desired output category set; others do not.
- *No-match*
  The actual output category set is disjoint with the desired output category set. This includes the case where the actual output category set is an empty set.

In what follows, we present classification results in term of the precision and recall rates under five categorized situations.

4.3. *Classification Results*

4.3.1. Training Phase. Table 2 shows the comparison results of the average exact-match accuracy of each layer for the proposed hierarchical BP neural network classifier with two hierarchical classifiers implemented by the vector space model (VSM) [15] and K nearest neighbors [27] in the training phase. Note that we only measure the case of exact-match in a training phase because the training results will be acceptable if and only if the exact-match results are satisfactory. Besides, the K nearest neighbors classifier is extremely sensitive to the value of K. A rule of thumb is that $K \leq \sqrt{\text{number of training data}}$ [27] is used to determine the value of K. In our experiments, the valuse of K has been optimized according to the rule of thumb.

4.3.2. Testing Phase. Table 3 shows comparison results of the classification accuracy rate of each level for the proposed hierarchical BP neural network classifier with two existing hierarchical classifiers respectively implemented by VSM and K nearest neighbors models in the testing phase. We measure precision and recall rates of document classification for the presented five different output situations discussed in Section 4.2. Obviously, the average recall rate is better than the average precision rate of each level for all tested classifiers. This phenomenon is predictable and reasonable. Since our training documents are multi-category and manually pre-classified, it is not easy to produce classification results that exactly match the desired outputs. From Table 3, we also find that the accuracy in level 1 is acceptable, but the accuracy rate of coverage-match situations decreases in levels 2 and 3 for all tested classifiers. Meanwhile, the overlap-match situation becomes more frequent. This result may be caused by noise in the training documents. For instance, many categories are named ‘Others’ in levels 2 and 3, and the meaning is that these categories are ambiguous. It is difficult to select significant features to represent such categories. Moreover, venders sometimes use some

Table 2. Comparison of the average exact-match accuracy of each layer for the proposed hierarchical BP neural network classifier with a single hidden layer and various hierarchical classifiers in the training phase.

Level	Hierarchical VSM classifier (%)	Hierarchical K nearest neighbors classifier (%)	Hierarchical BP classifier (%)
		(a) Testing set 1 (500 documents)	
1	100	100	100
2	100	100	100
3	97.3	100	100
		(b) Testing set 2 (3000 documents)	
1	88.2	90.3	90.1
2	85	86.5	87.8
3	80.3	84.5	85.2

Table 3. Comparison of the classification accuracy rate of each level for the proposed hierarchical BP neural network classifier with a single hidden layer and various hierarchical classifiers in the testing phase.

	Accuracy						
Level	Average precision rate (%)	Average recall rate (%)	Exact-match (%)	Coverage-match (%)	Subset-match (%)	Overlap-match (%)	No-match (%)
			Hierarchical VSM classifier				
(a) Testing set 1 (500 documents)							
1	35	88	2	84	0	12	0
2	47	82	8	69	0	20	0
3	43	73	10	46	0	42	0
			Hierarchical K nearest neighbors classifier				
(b) Testing set 1 (500 documents)							
1	36	91	3	85	0	10	0
2	50	88	9	70	3	17	0
3	45	75	13	48	0	37	0
			Hierarchical BP classifier				
(c) Testing set 1 (500 documents)							
1	37	91	3	86	0	9	0
2	52	87	10	69	4	14	0
3	48	77	14	51	0	33	0
			Hierarchical VSM classifier				
(d) Testing set 2 (3000 documents)							
1	19	74	1	68	0	30	0
2	21	70	0	59	0	40	0
3	13	59	0	44	0	55	0
			Hierarchical K nearest neighbors classifier				
(e) Testing set 2 (3000 documents)							
1	20	75	1	68	0	29	0
2	22	75	1	61	0	37	0
3	15	64	0	48	0	51	0
			Hierarchical BP classifier				
(f) Testing set 2 (3000 documents)							
1	21	76	1	72	0	26	0
2	24	74	1	63	0	34	0
3	17	66	0	54	0	45	0

attractive words (“excellent”, “perfect”, etc.) in their product descriptions. Although such words have no intended relevance for classification, they nonetheless influence feature selection. Besides, this phenomenon also shows the categories are getting similar while they are far away from the root category, thus leading to the overlap-match situation occurring more frequent. Based on this observation, we find that the linear classifier VSM performs the poorest classification ability due to the highest accuracy of overlap-match situation. Therefore, to solve hierarchical document classification problems, a non-linear classification mechanism is needed, especially on the categories which are far away from the root category. Our experimental results also show that the proposed hierarchical BP neural network classifier is superior to the two tested hierarchical classifiers implemented by VSM and K nearest neighbors models in terms of performance of document

Table 4. The classification accuracy rate of each level for the proposed hierarchical BP neural network classifier with two hidden layers in the testing phase (symbol↑denotes that classification accuracy rate is promoted compared with the result of Table 3, vice versa).

	Accuracy						
Level	Average precision rate (%)	Average recall rate (%)	Exact-match (%)	Coverage-match (%)	Subset-match (%)	Overlap-match (%)	No-match (%)
			Testing set 1 (500 documents)				
1	↑ 42	↓ 77	↑ 10	↓ 62	↑ 2	↑ 23	0
2	↓ 47	↑ 88	↓ 9	↑ 73	↓ 0	↑ 16	0
3	↓ 39	↓ 71	↓ 9	↓ 47	↑ 2	↑ 40	0
			Testing set 2 (3000 documents)				
1	↑ 27	↓ 58	↑ 4	↓ 51	0	↑ 43	0
2	↑ 32	↓ 56	↑ 6	↓ 41	↑ 2	↑ 48	0
3	↑ 38	↓ 60	↑ 7	↓ 40	↑ 7	45	0

classification in the tested document classification problem.

Furthermore, to compare the classification accuracy rates for the proposed hierarchical BP neural network classifiers with various hidden layers in the testing phase, the proposed hierarchical BP neural network classifier with two hidden layers is performed to discuss their testing generalization abilities. Table 4 shows the accuracy of each level for the hierarchical BP neural network classifier with two hidden layers in the testing phase. Experimental results show that the average performance of the proposed hierarchical BP neural network classifier with a single hidden layer is superior to the proposed neural network's architecture with two hidden layers in our tested dataset. Actually, we find that using BP neural network classifier with a single hidden layer is sufficient to solve document classification problems well. This is because a nonlinear document classification problem will gradually degenerate as a linearly separable pattern classification problem when its feature dimensions increase. Additionally, the upper categories in a hierarchical document classification structure are closer to being linearly separable. Thus, an approximate linear problem is not suited to solve it utilizing a complicated BP neural network architecture. On the contrary, our study also shows that a non-linear classification mechanism is needed, specially on the categories which are far away from the root category.

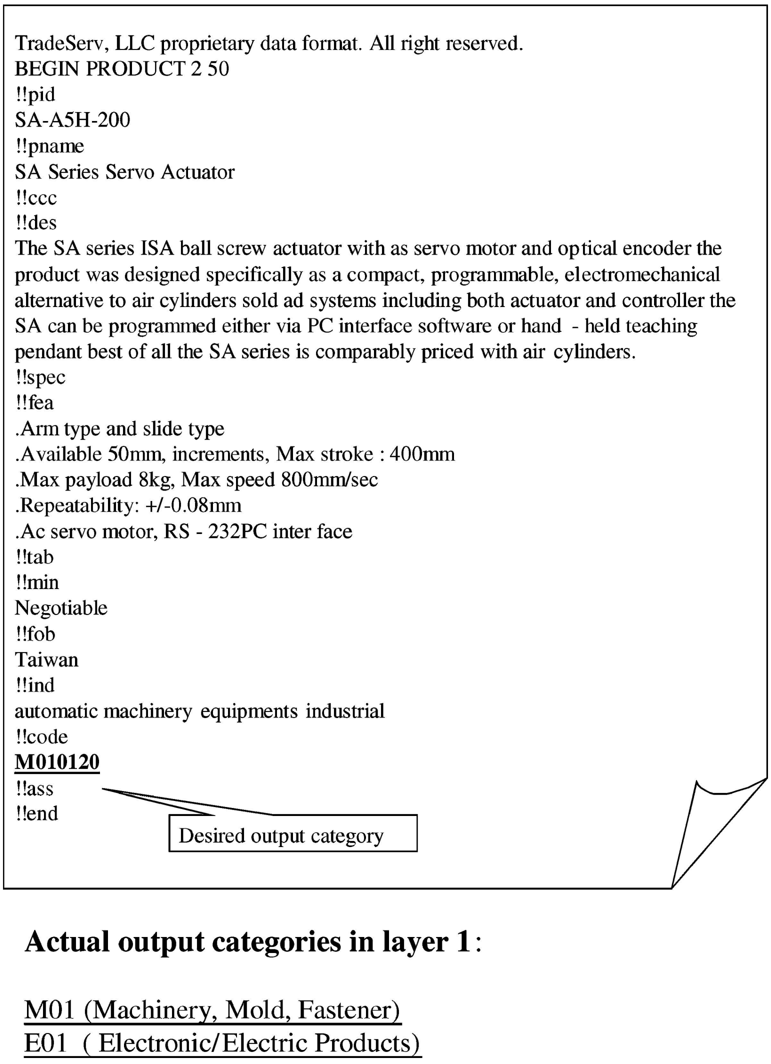
Furthermore, some output categories that are appropriate for a document but not included in the desired output set are discovered by our system. Figure 7 is an example of discovering appropriate categories. This product is manually classified to only one category M01 (Machinery, Mold, Fastener). From our system, we find that this product is also an electric product according to the description. That is, our system discovers the appropriate category E01 (Electronic/Electric product). Because the coverage-match situation occurs frequently, the number of actual output categories is usually more than that of desired output categories. This phenomenon causes the average recall to be better than the average precision. In this multi-category case, we emphasize discovering more potentially appropriate categories that may not have been discovered by manual classification (i.e., we promote greater recall) although the precision is merely adequate.

5. Discussion

In this section, we analyze the characteristics of our proposed model on the strength of the experimental results and discuss some possible enhancements that should be investigated in the future.

5.1. *Learning Performance*

To simplify the learning scheme, the well-known back-propagation learning model is used to build proper hierarchical classification units in this study. In our learning architecture, both the time consumed and number of features selected increase as the number of training



```
TradeServ, LLC proprietary data format. All right reserved.
BEGIN PRODUCT 2 50
!!pid
SA-A5H-200
!!pname
SA Series Servo Actuator
!!ccc
!!des
The SA series ISA ball screw actuator with as servo motor and optical encoder the
product was designed specifically as a compact, programmable, electromechanical
alternative to air cylinders sold ad systems including both actuator and controller the
SA can be programmed either via PC interface software or hand - held teaching
pendant best of all the SA series is comparably priced with air cylinders.
!!spec
!!fea
.Arm type and slide type
.Available 50mm, increments, Max stroke : 400mm
.Max payload 8kg, Max speed 800mm/sec
.Repeatability: +/-0.08mm
.Ac servo motor, RS - 232PC inter face
!!tab
!!min
Negotiable
!!fob
Taiwan
!!ind
automatic machinery equipments industrial
!!code
M010120
!!ass
!!end
```

Desired output category

Actual output categories in layer 1:

M01 (Machinery, Mold, Fastener)

E01 (Electronic/Electric Products)

Figure 7. An example of discovering appropriate category.

documents increases. Therefore, the learning performance can be improved by using the other enhanced learning algorithms if the proposed model is applied to commercial applications. Including a momentum term in the learning function is the most widely used improvement [26]. This method can accelerate the speed of convergence, but it might occur the problem of mean-square-error oscillation [28]. Besides, using the conjugate-gradient [29] and Levenberg-Marquardt [30] learning schemes to build proper hierarchical classification unit can deliver an improvement in learning speed to two orders of magnitude.

Moreover, supplementary learning [31] may be another effective improvement. Since most of the execution time of the BP learning algorithm is spent on backward error propagation, supplementary learning tries to economize on execution time by ignoring the errors of completely trained samples. Moreover, the learning rate can be decreased if the number of training samples that require backward error propagation is decreased.

Furthermore, neural networks likely occur overfitting learning during a training process so that the testing accuracy rate will obviously degrade. In our experiments, we find that this phenomenon can be detected or be avoided by using the difference of the present MSE and the previous MSE as the stop criterion to terminate the training process.

5.2. Thresholds for Feature Selection

Hou and Yang [32, 33] proposed a procedure to select the optimal threshold for feature selection by using the genetic algorithm (GA). The GA iterates three operations (reproduction, crossover and mutation) to figure out the appropriate solution for a given problem [24, 34]. However, to optimize the threshold parameters for feature selection is a time-consumed job for the proposed hierarchical BP classifier because the classification accuracy rate must be repeatedly evaluated to test various combinations of threshold parameters.

5.3. Discovery Learning

When classifying a new kind of document whose features have not been discovered by our feature selection before, we have to generate a new category and discover new features. Namely, in the future, we will investigate discovery learning approaches [35] to automatically analyze every input pattern and output category of our proposed BP classifiers. If any new patterns are found, we then add new nodes to proper BP classifiers and discover new features. Thus, knowledge can be accumulated automatically.

5.4. Noise Detection

It is possible that the training documents contain noise which will reduce the classification accuracy. That is, since the classification of the training documents is made by humans, the classification may have some mistakes or be subjective. In addition, in our commercial case, vendors may add many words to attract buyers, but those words are not meaningfully related to the specified product.

Those human factors make our classifiers learn the wrong knowledge and produce output with errors. Therefore, our further investigation will consider how to detect or correct the noise contained in the training documents.

5.5. Negative Words

The meaning of a sentence in natural language may have an exactly opposite effect if there are negative words in this sentence. Therefore, it is necessary to deal with negatives in the future. To solve this problem, an inference process might be needed. The inference process examines each word if there are negatives prior or posterior to it. If a word is negative, we assign a negative weight to it or regard it as an antonym during feature selection.

5.6. Human Interactions

Although our goal is to construct an automatic document classification system, the classification results may be improved if some processes are performed by humans at this early experimental stage. Human interaction is practical in three of the processes in our system: feature selection based on conformity and uniformity, synonym thesaurus construction and output expression.

As we mentioned in previous sections, feature selection based on conformity and uniformity is a key factor for selecting significant features. A flexible strategy is to refine the selected features by experts. Experts can verify the selected features and insert or delete features if necessary.

The synonym thesaurus is also an important factor influencing the classification results. The characteristics of categories will be ambiguous or misleading if we treat too many terms, or the wrong terms, as synonyms. To avoid these mistakes, the entire synonym thesaurus could be examined by experts after the synonyms are grouped by our system. Experts could modify the synonym thesaurus using their own knowledge.

Finally, the output could also be adjusted by humans. In actual applications, all of the output categories whose relational degrees are greater than 0 are considered, but only the top N ranked output categories are shown to users (N being a positive integer). These top N output categories are displayed to users with their relational degrees, and users then choose the most appropriate categories from these top N output categories.

6. Conclusion

In this paper, a 3-level hierarchical neural network classification system is proposed for automatic linguistic document classification. This classification system contains an effective feature selection procedure to analyze linguistic terms and select significant terms based on the analysis of conformity and uniformity. In the

3-level hierarchical neural network classification system, the hierarchy corresponds to the given category structure. Each BP classifier represents a parent category and classifies documents into the child categories that succeed it. The effectiveness of our classifier was tested by employing a product description document database. The proposed system was found to be adequate to aid the manual classification of product description documents.

Furthermore, there are some issues which need to be further investigated. With a large number of training documents, the memory requirements become a bottleneck in our system during feature selection and learning. For this reason, we will investigate a strategy to modify the matrix operations to break through this limitation. Determining appropriate thresholds for feature selection will also require a considerable further work. Finally, improvements in discovery learning, noise detection and processing negative words should be considered in the future.

References

1. R.E. Filman and Sangam Pant, "Searching the Internet," *IEEE Internet Computing*, July–August, pp. 21–23, 1998.
2. A. Sun, Ee-Peng Lim, "Hierarchical text classification and evaluation," in *Proceedings IEEE International Conference on Data Mining*, 2001, pp. 521–528.
3. M.E. Ruiz and P. Srinivasan, "Hierarchical text categorization using neural networks," *Information Retrieval*, vol. 5, no. 1, pp. 87–118, 2002.
4. M. Sasaki and K. Kita, "Rule-based text categorization using hierarchical categories," *IEEE International Conference on Systems, Man, and Cybernetics*, 1998, vol. 3, pp. 2827–2830.
5. R. Schettin, C. Brambilla, G. Ciocca, A. Valsasna, and M. De Ponti, "A hierarchical classification strategy for digital documents," *Pattern Recognition*, vol. 35, pp. 1759–1769, 2002.
6. C.H. Caldas and L. Soibelman, "Automating hierarchical document classification for construction management information systems," *Automation in Construction*, vol. 12, pp. 395–406, 2003.
7. G.P. Zhang, "Neural networks for classification: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, 2000.
8. P. Picton, *Neural Networks*, Palgrave: New York, 2000.
9. Hsinchun Chen, Chris Schuffels, and Richard Orwig, "Internet categorization and search: A self-organizing approach," *Journal of Visual Communication and Image Representation*, vol. 7, no. 1, pp. 88–102, 1996.
10. W.B. Frakes and R. Baeza-Yates, *Information Retrieval: Data Structures & Algorithms*, Prentice Hall PTR, 1992.
11. C. Jenkins, M. Jackson, P. Burden, and J. Wallis, "Searching the World Wide Web: An evaluation of available tools and methodologies," *Information and Software Technology,* vol. 39, pp. 985–994, 1998.
12. V.N. Gudivada, V.V. Raghavan, W.I. Grosky, and R. Kasanagottu, "Information retrieval on the World Wide Web," *IEEE Internet Computing*, September–October, pp. 58–68, 1997.
13. Ron Sun and L.A. Bookman, *Computational Architectures Integrating Neural and Symbolic Processes: A Perspective of the State of the Art*, Kluwer Academic Publishers, 1995.
14. Stefan Wermter and Ron Sun, *Hybrid Neural Systems*, Springer-Verlag Telos, 2000.
15. G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison Wesley, 1989.
16. M.F. Wyle and H.P. Frei, "Retrieval algorithm effectiveness in a Wide Area network information filter," in *Proc. of the 14th ACM SIGIR Conf. on R&D in Information Retrieval*, ACM, Chicago IL, 1991, pp. 114–122.
17. M.E. Porter, *Competitive Strategy: Techniques for Analyzing Industries and Competitors*, Free Press: New York, 1980.
18. W. Francis and H. Kucera, *Frequency Analysis of English Usage*, New York, 1982.
19. C.E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 279–423, 1948.
20. Y. Yang, C.G. Chute, and M. Clinic, "An example-based mapping method for text categorization and retrieval," *ACM Transaction on Information Systems*, vol. 12, no. 3, pp. 252–277, 1994.
21. M.T. Hagan, B. Demuth Howard, and H. Beale Mark, *Neural Network Design*, Martin HaganJan: Stillwater, 2002.
22. G.J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall: NJ, 1995.
23. Y.-L. Huang, "A theoretic research of cluster indexing for mandarin chinese full text document—the construction of vector space model," *Journal of Library and Information*, vol. 24, pp. 44–68, 1998.
24. Rothlauf and Franz, *Representations for Genetic and Evolutionary Algorithms*, Heidelberg, Physica-Verlag, 2002.
25. M.S. Bazarara, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming Theory and Algorithms*, John Wiley & Sons: New York, 1993.
26. Yi-Cheng Ye, *Applications and Implementation of Neural Network Models*, Ru-Lin, 1998.
27. Dunham Margaret H., *Data Mining Introductory and Advanced Topics*, N.J.: Prentice Hall/Pearson Education, 2003.
28. M. Torii and M.T. Hagan, "Stability of steepest descent with momentum for quadratic functions," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 752–756, 2002.
29. L. Mohan Saini and M. Kumar Soni, "Artificial neural network-based peak load forecasting using conjugate gradient methods," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 907–912, 2002.
30. G. Lera and M. Pinzolas, "Neighborhood based levenberg-marquardt algorithm for neural network training," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1200–1203, 2002.
31. T. Kimoto and K. Asakawa, "Stock market predication system with modular networks," in *IJCNN-90*, 1990, vol. 1, pp. 1–6.

32. Y.-C. Hou and S.-H. Yang, "A study on automatic document classification by combine fuzzy theory and genetic algorithms," *Journal of Fuzzy Systems*, vol. 4, no. 1, pp. 45–57, 1998.
33. Y.-Y. Yang, "Document Automatic Classification and Ranking," Master Thesis, Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, June 1993.
34. R.M. Friedberg, "A learning machine: Part I," *IBM Journal*, vol. 2, pp. 2–23, 1958.
35. F. Limin, *Neural Networks in Computer Intelligence*, McGraw Hill, 1994.