reference robust control scheme can improve the performance of induction motor drive with time delay and reduce its sensitivity to parameter variations and load disturbances. Moreover, the proposed new approach is presented in a manner that will contribute to a better understanding of NN applications to speed-tracking control and robust control in induction motors.

REFERENCES

[1] B. K. Bose, "Motion control technology—Present and future," *IEEE Trans. Ind. Applicat.*, vol. IA-21, pp. 1337–1342, Nov./Dec. 1985.

[2] C. W. Liaw, F. J. Lin, and Y. S. Kung, "Design and implementation of a high-performance induction motor servo drive," *Proc. Inst. Elect. Eng., Electric Power Applications*, vol. 140, pp. 241–248, July 1993.

[3] R. Balasubramanian and K. H. Wong, "A microcomputer-based self-tuning IP controller for DC machines," *IEEE Trans. Ind. Applicat.*, vol. IA-22, pp. 989–999, Nov./Dec. 1986.

[4] J. K. Ji and S. K. Sul, "DSP-based self-tuning IP speed controller with load torque compensation for rolling mill dc drive," *IEEE Trans. Ind. Electron.*, vol. 42, pp. 382–386, Aug. 1995.

[5] M. Iwasaki and N. Matsui, "Robust speed control of IM with torque feedforward control," *IEEE Trans. Ind. Electron.*, vol. 40, pp. 553–560, Dec. 1993.

[6] T. C. Chen and J. S. Chen, "A microprocessor-based induction motor drives with a novel PWM control," *J. Control Syst. Technol.*, vol. 3, no. 1, pp. 71–78, 1995.

[7] K. J. Astrom, C. C. Hang, and B. C. Lim, "A new Smith predictor for controlling a process with an integrator and long dead time," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 343–345, Feb. 1994.

[8] F. J. Lin and C. W. Liaw, "Control of indirect field-oriented induction motor drives considering the effects of dead time and parameter variations," *IEEE Trans. Ind. Electron.*, vol. 40, pp. 486–495, Oct. 1993.

[9] T. Fukuda and T. Shibata, "Theory and applications of neural networks for industrial control system," *IEEE Trans. Ind. Electron.*, vol. 39, pp. 472–489, Dec. 1992.

[10] V. Etxebarria, "Adaptive control discrete systems using neural networks," *Proc. Inst. Elect. Eng., Control Theory Appl.*, vol. 141, pp. 209–215, July 1994.

[11] T. C. Chen and C. Y. Liaw, "Design of a neural fuzzy with PI controller for induction motor speed control," *Int. J. Knowledge-Based Intell. Eng. Syst.*, vol. 2, pp. 9–16, Jan. 1998.

[12] C. Forgez, B. Semail, and J. P. Hautier, "Adaptive speed control of an AC machine coupled to a nonlinear load torque using neural network," in *Proc. ICEM*, vol. 1, Istanbul, Turkey, Sept. 1998, pp. 608–613.

[13] B. K. Bose, *Power Electronics and AC Drives*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[14] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[15] A. T. Bahill, "A simple adaptive Smith predictor for controlling time-delay systems," *IEEE Control Syst. Mag.*, vol. 3, pp. 16–32, May 1983.

[16] Texas Instruments Incorporated, *TMS320C3X User's Guide*, Mar. 1996. Ver. 1.00.

[17] Texas Instruments Incorporated, *TMS320C32 Floating-Point DSP Optimizing C Compiler User's Guide*, Feb. 1995. Rev. B.

[18] B. C. Kuo, *Digital Control Systems*. Philadelphia, PA: Saunders, 1992.

[19] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control," *IEEE Trans. Neural Networks*, vol. 6, pp. 144–156, Jan. 1995.

FIDS: An Intelligent Financial Web News Articles Digest System

Wai Lam and Kei Shiu Ho

***Abstract*—In this paper, we present a system, called FIDS (Financial Information Digest System), which can digest online financial news automatically. Compared to the previous approaches, FIDS is unique in the way that it can understand news articles in different domains simultaneously. These domains are all concerned with financial news. The system is able to integrate the information from different articles by conducting automatic content-based classification and information item extraction. Moreover, it allows us to perform cross-validation on their contents. As a result, users can have access to more complete information which otherwise would be scattered in different articles.**

***Index Terms*—Classification, text summarization, web financial news.**

I. INTRODUCTION

We are now living in an information explosion era. There is no denying that the wealth of information provided by the Internet (e.g., online news, discussion groups, etc) is a valuable resource to its users. But at the same time, the enormous volume of information available has also prohibited an easy access to the right information. Users are faced with the problem of information overload—much time is spent in order to sift out useful or relevant information manually.

Recently, this problem has aroused the interests of many researchers. Generally speaking, previous work in this area can be grouped into several streams. Among them, text filtering [1], [20], [25], [31], [34] aims at selecting the right documents or information for the user based on the user's preference, which is modeled by a profile. During operation, the profile is compared to the documents available to select those articles that may be of interest to the user. Usually, the profile is formed by keywords that are specified by the user. In other cases, it may be inferred by the text filtering system from documents that have been identified as "interesting" to the user. In other words, the system is adaptive in nature—it can detect the specific interests of the user automatically and fine-tune itself accordingly to improve its future performance. For example, SIFTER [33] employed a Bayesian-based shift detection model that could track changes in the user's interests. When a shift occurred, the system would perform a relearning of the user's profile automatically.

A related problem to information filtering is text classification, in which an article is being assigned to one of a number of classes based on its contents. This problem has been well studied previously and various techniques have been discussed in the literature, including machine learning methods (e.g., decision tree [2]), statistical classifiers (such as LMS [27], the Rocchio algorithm [21], and support vector machine [22]), and neural network approaches (e.g., multilayer perceptrons [13], learning vector quantizer [24]). In addition, work has been done to combine multiple classifiers together to improve classification accuracy (see, e.g., [28]). In some sense, information filtering can be

Manuscript received March 15, 1999; revised May 1, 2000 and September 21, 2001. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, Grant PolyU5088/99E.

W. Lam is with the Department of Systems Engineering & Engineering Management, Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: wlam@se.cuhk.edu.hk).

K. S. Ho is with the Department of Computing, Hong Kong Polytechnic University, Hung Hom, Hong Kong (e-mail: csksho@comp.polyu.edu.hk).

Publisher Item Identifier S 1083-4427(01)11292-0.

perceived as a special form of text classification, where the number of classes is restricted to two.

On the other hand, a more challenging task is tackled by the text understanding approach which aims at summarizing a document by extracting the relevant information only. It differs from text categorization and information filtering in two major aspects. First, in order to locate the target information, a more in-depth analysis of the document is usually required which will unavoidably involve the application of natural language processing techniques. Definitely, this makes the task more difficult. Second, due to its inherent difficulty, a text summarization system usually focuses on a single domain only. For example, SCISOR [19] and the system reported in [10] were both dedicated for analyzing financial news. Recently, a series of conferences, known as the Message Understanding Conferences (MUC), have been held which were competitive evaluations for text summarization systems (see, e.g., [7]). The impressive performances of many of the MUC participants have demonstrated the potential application of simple natural language processing techniques in digesting textual information.

Many of the early text summarization systems produced the summary of a document by selecting salient text spans, e.g., sentences [4], [14] or passages [44], from the document. Usually, the salience of a text span was defined by criteria such as its position in the document, its length, or other statistical measures (e.g., term frequency or inverse document frequency [39]). Despite the simplicity of the approach, summaries produced by these systems are often not coherent, since the text spans selected may actually be located dispersedly in the original document (so their contents may be unconnected at all). Besides, they tend to be lengthy and redundant in meaning.

In view of this, more sophisticated techniques have been applied to improve the qualities of the summaries generated. For example, SUMMARIST [17] made use of the thesaurus WordNet and various concept fusion/generalization techniques to identify the salient concepts discussed in the document, i.e., the topic. The summary was then produced by selecting sentences that were pertinent to the topic. On the other hand, various systems have been proposed that exploited the discourse structure of text [8], [29], [38], [44]. Among them, Marcu [29] used cue phrases and other linguistic features to derive the rhetorical structure tree of the whole document, from which an "importance score" was assigned to each text span as a measure of its salience. Text spans with a high importance score could then be output as the summary. Another work along a similar direction was reported in [8], in which sentence segmentation was first performed using cue phrases. The rhetorical relations among the segments in the same sentence were then identified which, together with other statistical features, would be used to encode the segments. Finally, a learning algorithm was applied to select salient segments for inclusion in the summary.

Recently, researchers have been working on a different approach to text summarization. Instead of excerpting text spans, the summary is produced by filling in predefined templates (implemented in one form or another) using information items extracted from the document. This approach is commonly called information extraction. In general, the design of an extraction template requires much domain knowledge, and certain syntactic and/or semantic constraints have to be satisfied before it can be applied. Effectively, a deeper understanding of the text's meaning is undergone (as compared with the excerption-based methods). For example, in WHISK [43], syntactic analysis was first performed on the document. Extraction rules consisting of regular expressions were then applied. They were formed using syntactic and semantic tags which could be flexibly matched with the parsed text to extract the target information.

Other implementations of the extraction templates have also been proposed previously. For instance, in CRYSTAL [42], extraction templates were implemented as a dictionary of concept nodes, whereas in

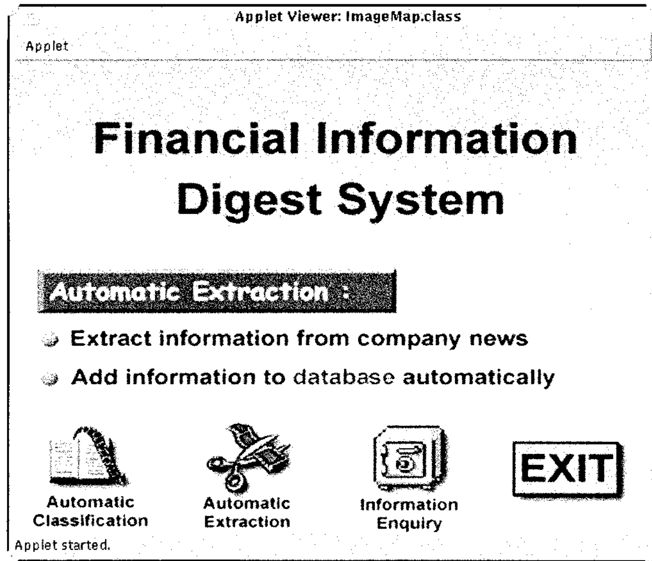

Fig. 1. Starting screen of the prototype FIDS.

FASTUS [16], they were modeled as finite-state transducers. In order to relieve the cumbersome knowledge engineering work involved in designing the extraction templates, some systems such as AutoSlog [36], LIEP [18], and CRYSTAL [42] have also provided a learning mechanism for inducing the templates for a new domain automatically. Experimental results (e.g., [37]) revealed that the learned templates had a comparable performance as the hand-crafted ones. Doubtlessly, the portability of the systems can be promoted.

Recently, with the advent of information agents (e.g., [23], [46]), the fruits of the effort in the above three research areas have been further enriched. Generally speaking, an information agent is a piece of intelligent software capable of collecting, filtering, and integrating information from multiple sources. For example, CiteSeer [3] could autonomously submit queries to different Internet search engines to instruct them to look for relevant research articles on the Web that were of interest to the user. Some agents, so called mobile agents, can even move from one Web site to another physically to look for interested information [5]. In this way, analysis and filtering of information can be done remotely, and "assimilated" information instead of raw information will be returned to the users. Communication costs can thus be reduced. On the other hand, in some approaches, multiple agents are deployed that can work collaboratively to exchange information with one another (e.g., Gossip [15]). This can definitely enhance the effectiveness and efficiency of the information searching process.

In this paper, we present a text summarization system for online financial news. The system, known as FIDS (Financial Information Digest System), basically performs three tasks: classification, information extraction, and information enquiry. News articles are periodically downloaded from online news servers. Based on its contents, each article is first classified into one of the five categories, namely, company performance, economy, merger and acquisition, services and products, and securities. For each category, a template is predefined which is basically a list of items of information that one can expect from a standard article in that category. Information is extracted from the article to fill in the template, which can then be presented as a summary to the user or be stored in a database to support user queries. A prototype of FIDS has been built, and preliminary experiments reveal that it has very encouraging performance. Fig. 1 shows the starting screen of the system. The significance of FIDS is threefold. First, as we have mentioned earlier, the volume of information available on the Web has

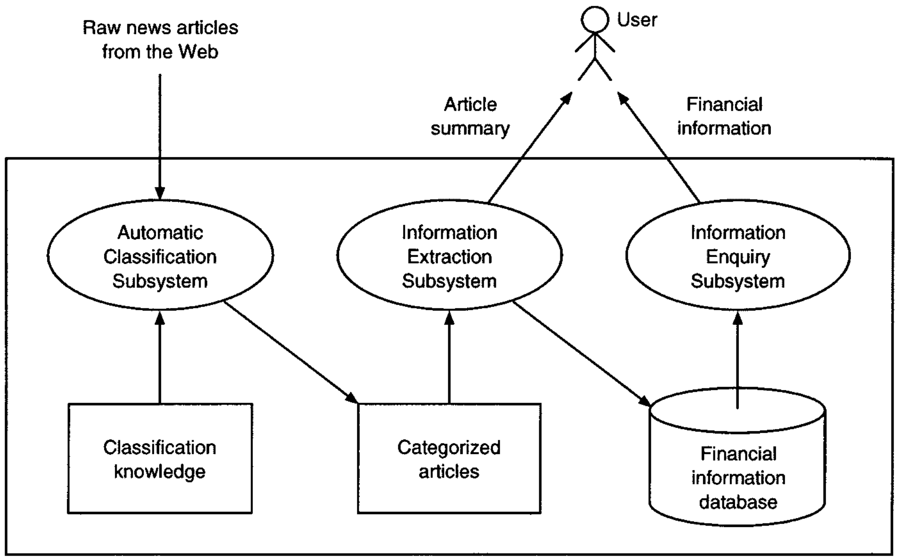

Fig. 2. An overview of FIDS.

already grown to an extent which is nonmanageable by human users. FIDS can help users filter out all irrelevant information and present a quick but yet condensed summary which preserves the gist of the reported stories. This advantage becomes more apparent when an instant access to information is demanded to support rapid decision-making (as in the financial markets).

Second, most of the existing text understanding systems focus on a single domain only (e.g., joint ventures). And for those that process more than one domain (such as [10]), few attempt to address the issue on relating the information as found in the different domains. In contrast to the previous approaches, FIDS is unique in the way that it can address multiple domains simultaneously. More importantly, these domains are all concerned with financial news. In other words, they are inter-related.

In FIDS, relevant information of an article is extracted as items of a schema. Each item is then stored in a database as a record. In this way, linkages can be maintained between the information items of related articles in different domains by, say, indexing all records using the same field such as the company name. Effectively, the information extracted from different articles is integrated implicitly. Then, by querying the database using the company name as the key, all records concerning that company can be retrieved. Hence, missing information or supplementary materials for an article in one domain may possibly be found in a related article of a different domain [30]. Or, articles from different sources describing the same story (which may even be written in different languages) can be cross-validated. In case they are inconsistent or even contradictory, the system may provide different viewpoints to the user. This creates a leverage effect on the utility of the information extracted.

Finally, FIDS can be adapted to serve as a kind of "Internet watchdog" for individuals or companies. By continuously monitoring information appearing on the Internet (e.g., newsgroups, online news, etc), the system can spontaneously report any information which may be critical to one's interest and guard against any online materials which may hurt one's goodwill.

II. Overview of FIDS

FIDS consists of three main subsystems, namely, *Automatic Classification Subsystem*, *Information Extraction Subsystem*, and *Information Enquiry Subsystem*. Fig. 2 illustrates an overview of FIDS. First,

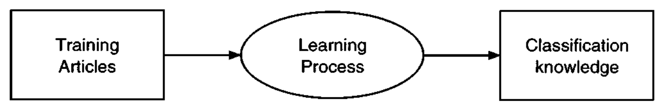

Fig. 3. Classification knowledge learning subsystem.

TABLE I
SAMPLE FEATURES

Account
Achief
Acquir
Action
Badly
Balanc
Bank
Bankcard
Benchmark
Billion
Blow
Bond
Bondhold

raw financial Web news articles are automatically downloaded from the Internet periodically. The source Web sites are mainly Internet edition of newspaper agencies. The Automatic Classification Subsystem analyzes the contents of these articles and classifies them into appropriate categories. The system can currently recognize five categories, namely, *company performance*, *economy*, *merger and acquisition*, *financial services and products*, and *securities*. To classify a news article, the system employs some classification knowledge to analyze the text contents of the article and assigns it with a suitable category. The classification knowledge is represented in rule format and automatically learned by the *Classification Knowledge Learning Subsystem*.

The Information Extraction Subsystem then extracts useful financial information from a categorized article as produced from the classification process. Currently we focus on the *company performance* category for this extraction process. The schema-based approach is employed for this task. In particular, a schema for the company performance category is built. This subsystem will attempt to extract major facts and information from the article contents. It then generates an article summary and displays the summary to the user. In addition, the summary and the

1. let All_Tests be the set of all possible feature tests;
   Best_Rule = nil; Conditions_Set = $\emptyset$
2. repeat
3. specialize all conditions in Conditions_Set by constructing New_Conditions_Set as $\{c \wedge t \mid c \in \text{Conditions\_Set}, t \in \text{All\_Tests}\}$
4. examine each condition R in New_Conditions_Set;
   if R is better than Best_Rule according to the rule strength metric, replace Best_Rule by R
5. discard the worst conditions from New_Conditions_Set
   until the size of New_Conditions_Set is less than a threshold
6. Conditions_Set = New_Conditions_Set
7. until Conditions_Set = $\emptyset$

Fig. 4. Module for searching a good rule during the induction of the classification rules.

1. company name
2: date of period
3. company performance (good, fair, poor)
4. balance sheet data
   - 4.1 revenue
   - 4.2 net income (net income, income after tax)
   - 4.3 asset
   - 4.4 turnover (turnover, sales)
   - 4.5 earning per share (earning per share, value per share)
   - 4.6 sales (sales, sales revenue, amount of sales)
   - 4.7 loss (loss in sales, decrease in sales)
   - 4.8 delinquency (mistake decision, wrong decision, poor investment)
   - 4.9 income (sales before tax, total sales, gross revenue)
   - 4.10 liability/loan (liability, loan, debt, borrowing)
   - 4.11 expectation gain
5. new product/service (good sales, optimistic sales value)
   - 5.1 selling (good)
   - 5.2 growth (good)
   - 5.3 improvement in market position (gain the market share, leading)
   - 5.4 lower cost (reduce the cost, decrease in cost)
   - 5.5 company restructuring (new branch, expand size of company)
6. performance in the last period (compare to)
   - 6.1 overall performance (excellent, good, fair, poor, very poor)
   - 6.2 reason for change in performance (wrong investment, lack of planning, poor planning, inexperience, market change, change in key people)
7. financial issue:
   - 7.1 no. of share issue:
   - 7.2 amount get from public
8. major business acitivities:
   - 8.1 amount earning
   - 8.2 other income/loss (other investments, business)

Fig. 5. Extraction schema for the category *company performance*.

extracted information can be stored in the financial information database for future enquiry.

The Information Enquiry Subsystem supports user queries on company performance stored in the financial information database. The enquiry process is performed in an interactive manner. A user may select or enter a company name, the system will then retrieve all the summaries related to the given company from the database and display them on the screen. A user can also query about a specific item related to company performance such as *profit*, *loss*, *dividend*, *earnings per share*, or *turnover*.

As mentioned above, classification knowledge is needed for the Automatic Classification Subsystem. The classification knowledge is produced in advance by the Classification Knowledge Learning Subsystem. To prepare for the training process, a set of articles in each of the five different categories are manually collected. The subsystem makes use of a machine learning technique on the training articles to

```
1. repeat for each sentence S
2.     repeat for each item I of the schema
3.         if S contains one of the keywords of I then
4.             match S against the extraction patterns of I to extract
               the relevant information from S
```

Fig. 6. Algorithm for extracting relevant information from a document.

TABLE II
CLASSIFICATION PERFORMANCE WHEN THE TF SCHEME IS USED

PREDICTED CATEGORY	ACTUAL CATEGORY Company Performance	Economy	Merger and Acquisition	Services and Products	Securities
Company Performance	37	1	3	4	3
Economy	2	31	2	1	2
Merger and Acquisition	7	4	45	10	12
Services and Products	1	4	0	37	1
Securities	0	6	2	1	35
ACCURACY	78.7%	67.4%	86.5%	69.8%	66.0%

TABLE III
EXAMPLE OF A CLASSIFICATION RULE FOR THE CATEGORY *SECURITIES* WHEN THE TF SCHEME IS USED

IF	cent = Y
	AND currenc = N
	AND dividend = N
	AND expend = N
	AND grow = N
	AND index = Y
THEN	category = securities

discover the classification knowledge automatically, as illustrated in Fig. 3. Note that this learning is only performed once.

III. Representing News Articles

A. Feature Selection in Text Processing

To process a news article, we need to represent it in some way. Traditionally, text documents have been encoded using features [41]. These features, also called terms, were in general formed by words appearing in the documents. Each article was thus transformed into a set or vector of features based on its text contents. In order to process unrestricted texts, we need to develop an automated technique for extracting content features based on information retrieval methods. Once these features are obtained, they can be used for subsequent processing.

However, if one simply defined a feature for every word contained in the documents, the number of features collected would be enormous. Processing efficiency and even accuracy could then be compromised as a result. In view of this, some scoring measure was often applied which assigned a weight to each feature identified. This weight was supposed to reflect the relevance or significance of the feature with respect to the articles. All the features were then ranked according to their weights, and a set of "important" features (so called keywords) were selected for representing the news articles. Previously, various weighting schemes have been proposed, including document frequency [45], inverse document frequency [40], information gain [45], and frequency odds ratio [31], to name but a few. In FIDS, two weighting schemes, namely, term frequency (TF) and inverse document frequency (IDF), are used to form a combined weighting scheme. Experimental results (see Section VI) reveal that this combined scheme is effective in capturing the text contents of the articles and it can lead to good performance in the classification task.

TABLE IV
THE CLASSIFICATION PERFORMANCE WHEN THE TF-IDF SCHEME IS USED

PREDICTED CATEGORY	ACTUAL CATEGORY Company Performance	Economy	Merger and Acquisition	Services and Products	Securities
Company Performance	40	2	0	4	0
Economy	1	34	1	2	3
Merger and Acquisition	1	1	48	6	1
Services and Products	3	3	1	38	2
Securities	2	6	2	3	47
ACCURACY	85.1%	73.9%	92.3%	71.7%	88.6%

TABLE V
CLASSIFICATION RULES INDUCED FOR THE CATEGORY SECURITIES WHEN THE TF-IDF SCHEME IS USED

	Rule 1		Rule 2
IF	cent = Y	IF	affair = N
	AND currenc = N		AND econom = N
	AND dividend = N		AND index = Y
	AND expend = N		AND new = N
	AND grow = N		AND remain = N
	AND index = Y	THEN	category = securities
THEN	category = securities		Rule 4
	Rule 3	IF	asset = N
IF	acquir = N		AND billion = Y
	AND brand = N		AND investig = Y
	AND chip = Y	THEN	category = securities
	AND custom = N		Rule 5
THEN	category = securities	IF	bank = N
			AND jump = Y
			AND operat = N
		THEN	category = securities

TABLE VI
CLASSIFICATION ACCURACY COMPARISON BETWEEN FIDS AND THE DECISION TREE METHOD (C4.5)

	FIDS	Decision Tree (C4.5)
Company Performance	85.1%	83.1%
Economy	73.9%	72.4%
Merger and Acquisition	92.3%	89.5%
Services and Products	71.7%	75.4%
Securities	88.6%	82.0%

B. Encoding Documents by the TF-IDF Scheme

In FIDS, encoding of news articles is performed as follows. First, common stop words such as "the", "of", and "is" are eliminated from the texts since these words do not carry useful information to characterize the document (e.g., see [11]). After removing stop words, we select those words from a dictionary that commonly appear in financial news articles. These words, after being stemmed (using a stemming algorithm such as [12]), form the basis of the terms for generating features. Totally, there are 251 terms selected. Table I shows some of these terms.

HKE profits dip 8.5pc

By Stephen Seawright

Hong Kong Electric (HKE) reported an 8.5 per cent fall in its profits for the first half of the year mostly due to the small contribution from its property arm.

Interim net profit to June reached \$1.909 billion compared with \$2.087 billion in the same period last year.

Despite the fall in profits the company announced an interim dividend of 53.5 cents per share, an increase of 5.9 per cent over the 1997 interim payment.

Earnings per share, however, fell to 94 cents from 103 cents in the first half of last year.

Total profits fell because the group's property development associate, Secan, only made a small contribution to profits this year.

Gross profit from the associate company was only \$3 million in the first half of this year as opposed to a contribution of \$281 million last year.

Excluding associate companies, operating profit rose 6.8 per cent to \$2.18 billion in the first half of the year.

Turnover jumped to \$4.03 billion from \$3.59 billion last year.

⋮ ⋮ ⋮

Fig. 7. Example article which is classified under the category *company performance* by FIDS.

We treat each term as a potential feature. Specifically, each feature is associated with a weight which forms an element for document representation. To obtain the weight g_{ij} for term T_j in document D_i, we first compute the *term frequency* t_{ij} which is basically the occurrence frequency of term T_j in document D_i. Higher occurrence frequency will contribute to larger weight. This scheme is referred to as the TF scheme.

In addition to the term frequency, we also compute the *inverse document frequency* (IDF) which is commonly used in information retrieval research [21], [40]. The IDF of a term is defined as $\log(N/p_j)$ where N is the total number of documents in a collection and p_j is the number of documents in which term T_j appears. This factor captures the usefulness of a term since it gives a lower weight for a term which appears in too many documents in the collection. Using the term frequency and the inverse document frequency, we obtain a document representation, commonly referred to as the TF-IDF scheme [21], [40], [45], incorporating these two frequency elements. Specifically, the weight g_{ij} is given as

$$g_{ij} = t_{ij} \log \frac{N}{p_j}. \tag{1}$$

Consequently, we can represent a document D_i by a vector $(g_{i1}, \ldots, g_{ip})$ where p is the total number of terms extracted as features.

IV. Automatic News Classification

The classification knowledge is represented in rule form. The process of induction of classification rules is based on a learning algorithm known as CN2 [9]. Specifically, the rules are in the form of:

IF⟨conditions⟩THEN the document is in class C.

The conditions in the antecedent is basically a conjunction of feature tests. A feature test is in the form of (⟨feature⟩ ⟨operator⟩ ⟨value⟩), where ⟨operator⟩ denotes a comparison operator such as ⟨ or ⟩. For example, a classification rule can be

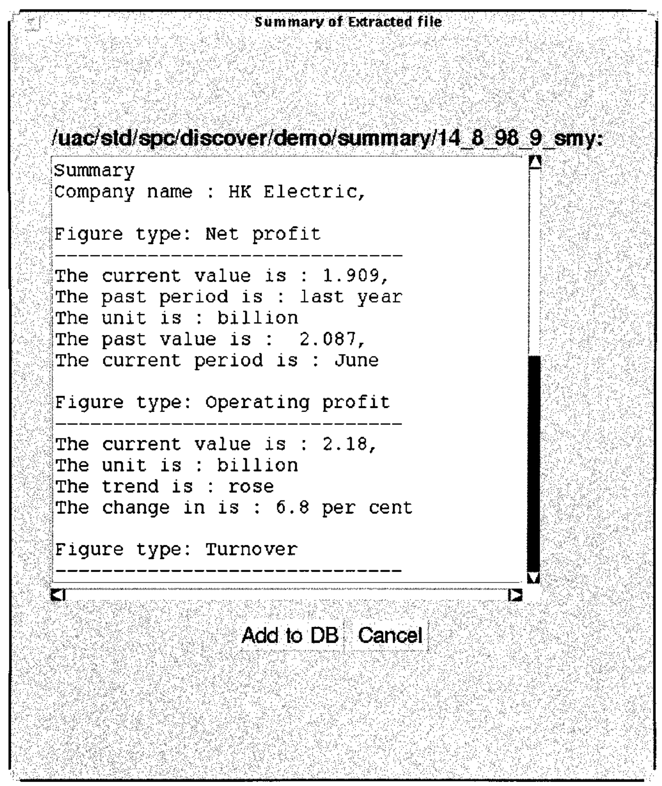

Fig. 8. Items of relevant information (partial) as extracted from the news article in Fig. 7 by FIDS.

IF (internet > 0.0)and (program> 0.0)
THEN the document is relevant.

The rules are learned in an iterative fashion. In each iteration, the algorithm invokes a module for searching for a good rule as explained below. Each learned rule is associated with a rule strength metric. We use the Laplace expected error estimate [6] for this metric which is given by

TABLE VII
EXTRACTION PERFORMANCE OF FIDS IN PROCESSING THE 47 ARTICLES BELONGING TO THE CATEGORY COMPANY PERFORMANCE

	Items	Extracted	Correct	Incorrect	Precision	Recall
11_4_98_6	25	14	12	2	0.8571	0.4800
11_5_98_2	9	3	2	1	0.6667	0.2222
11_5_98_7	15	12	12	0	1.0000	0.8000
11_8_98_18	7	7	7	0	1.0000	1.0000
14_8_98_14	10	6	6	0	1.0000	0.6000
15_5_98_11	20	13	13	0	1.0000	0.6500
15_7_98_7	15	7	4	3	0.5714	0.2667
16_3_98_5	8	1	1	0	1.0000	0.1250
16_5_98_12	18	7	6	1	0.8571	0.3333
16_5_98_18	21	13	13	0	1.0000	0.6190
16_5_98_9	11	5	5	0	1.0000	0.4545
16_7_98_10	27	15	14	1	0.9333	0.5185
17_3_98_11	18	14	14	0	1.0000	0.7778
17_5_98_12	18	7	7	0	1.0000	0.3889
17_5_98_18	21	13	13	0	1.0000	0.6190
17_5_98_9	11	5	5	0	1.0000	0.4545
17_8_98_13	1	1	1	0	1.0000	1.0000
18_2_98_2	6	5	5	0	1.0000	0.8333
18_3_98_9	10	9	7	2	0.7778	0.7000
18_4_98_7	16	8	7	1	0.8750	0.4375
19_2_98_20	11	9	9	0	1.0000	0.8182
19_4_98_3	24	21	21	0	1.0000	0.8750
19_8_98_19	9	11	9	2	0.8182	1.0000
19_8_98_6	14	13	13	0	1.0000	0.9286
20_8_98_10	16	16	16	0	1.0000	1.0000
21_4_98_12	17	12	11	1	0.9167	0.6471
21_4_98_16	26	15	13	2	0.8667	0.5000
21_4_98_9	32	17	13	4	0.7647	0.4063
21_8_98_2	10	5	3	2	0.6000	0.3000
22_4_98_13	18	7	7	0	1.0000	0.3889
22_4_98_7	19	8	7	1	0.8750	0.3684
22_5_98_3	21	7	7	0	1.0000	0.3333
22_8_98_7	20	8	8	0	1.0000	0.4000
23_4_98_10	11	7	7	0	1.0000	0.6364
23_4_98_14	26	20	18	2	0.9000	0.6923
23_5_98_8	17	15	15	0	1.0000	0.8824
24_2_98_2	17	11	11	0	1.0000	0.6471
24_5_98_15	18	15	15	0	1.0000	0.8333
25_3_98_14	17	9	9	0	1.0000	0.5294
27_3_98_0	24	22	17	5	0.7727	0.7083
27_3_98_1	14	5	5	0	1.0000	0.3571
27_5_98_24	6	6	3	3	0.5000	0.5000
28_2_98_6	9	1	1	0	1.0000	0.1111
28_5_98_12	19	6	6	0	1.0000	0.3158
2_4_98_11	20	20	20	0	1.0000	1.0000
8_4_98_15	16	9	8	1	0.8889	0.5000
8_4_98_64	11	5	3	2	0.6000	0.2727
Overall:	749	465	429	36	0.9158	0.5794

$(N_P+1)/(N_R+2)$ where N_P is the number of relevant documents covered by the rule and N_R is the total number of documents covered by the rule. After a rule is found and added to the classification rule set, those positive examples it covers are removed from the training set. The algorithm will terminate if no more positive examples remain.

The module for searching for a good rule is given in Fig. 4. To generate a possible feature test for step 1, we first divide the range of weights of each feature into discrete intervals. A test on a feature, say G_j, can be constructed in the form of testing whether G_j is larger or smaller than a certain interval boundary. Beam search is applied to find a good rule. The heuristic function used for evaluating the merit of a rule is the rule strength metric defined above.

V. SCHEMA-BASED INFORMATION EXTRACTION

A. Extraction Patterns

Generally speaking, an information extraction system fills in predefined templates by looking for relevant information from the text.

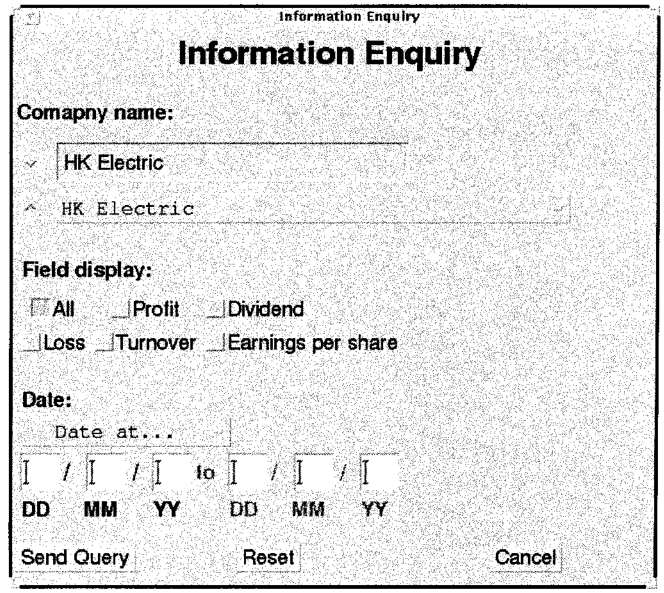

Fig. 9. Sample screen of the information enquiry subsystem.

Intuitively, each template, commonly called an extraction pattern, represents an event pertinent to the domain. Usually, it is associated with certain syntactic and semantic constraints which have to be satisfied before the pattern can be activated (e.g., the occurrence of a keyword). During extraction, each extraction pattern is applied to a text at the sentence or clause level to extract specific types of information.

Previously, various methods have been proposed for implementing the extraction patterns. For example, in the CIRCUS system [26], the extraction patterns were implemented as a dictionary of concept nodes. Each concept node specified an event of interest to the target domain, and was triggered by the presence of a verb in a sentence or a clause. A concept node was basically a case frame consisting of a set of slots. Each slot expected a particular type of information and provided a clue of where that information could be located in the sentence or the clause (e.g., the direct object).

On the other hand, in the FASTUS system [16], extraction patterns were modeled using finite-state transducers. The specific phrases as expected by the pattern of a domain event were "hard-coded" in the edges of a finite-state transducer. To recognize an event, these phrases would have to appear in the exact order so as to trigger the correct sequence of state transitions necessary to bring the transducer to its final state (flexibility was thus constrained).

Still another approach has been put forth in the LIEP system [18], where each extraction pattern was implemented as a rule whose antecedent part consisted of a number of clauses or preconditions. When a rule was applied to a sentence, some of its clauses first identified certain target sentence elements (which were syntactic constituents representing domain-specific entities such as person or company names), The other clauses then verified if certain pre-defined syntactic relationships were plausible between the identified elements. If all the pre-conditions could be satisfied, the event described by the rule was logged.

B. Extraction Schema in FIDS

In FIDS, an extraction schema is hand-crafted for each of the five categories. Basically, it is a checklist summarizing the types of information that one can expect from an article of that category. Our rationale is that, although the style of writing may vary from one article to another, the types of materials appearing in different articles of the same category will be more or less the same. This similarity is cap-

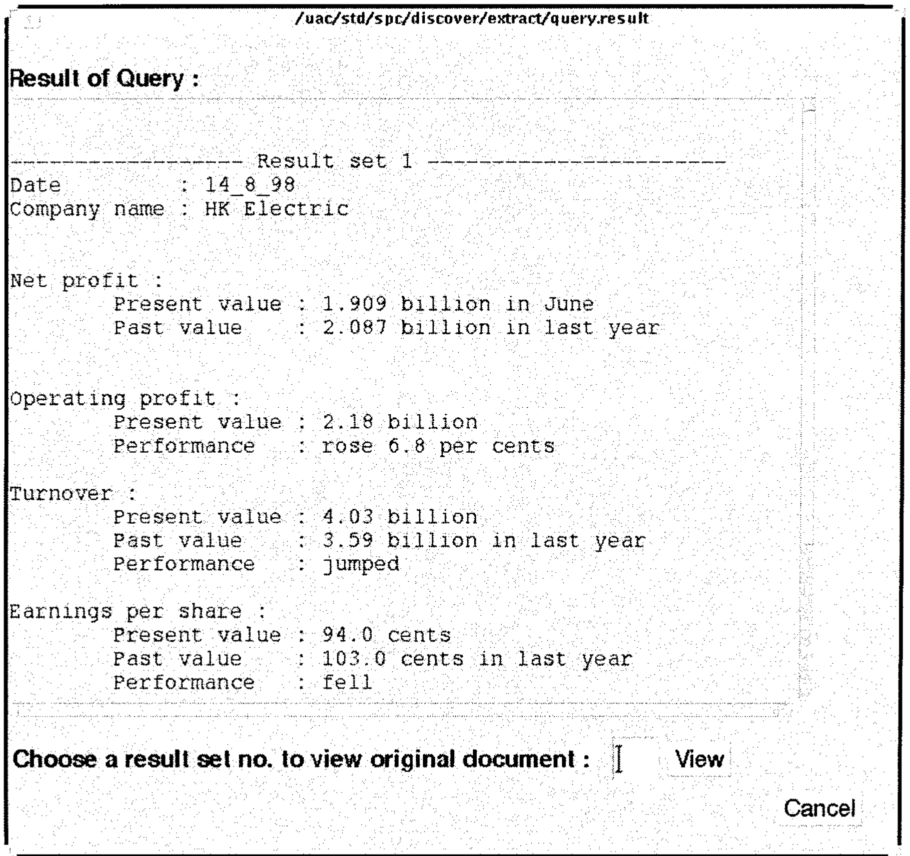

Fig. 10. Output of the information enquiry subsystem with respect to the enquiry as shown in Fig. 9.

tured by the schema (it thus resembles the case-based templates used in [10] and the text grammars proposed in [32], [38]) . For example, the schema for the *company performance* category is depicted in Fig. 5.

As shown, the schema is structured and it consists of a list of items (e.g., company name). Each item may further consist of one or more fields. Associated with each item or field is a set of keywords. For example, the field "turnover" of the item "balance sheet data" contains the two keywords "turnover" and "sales." Each item or field is also associated with some extraction patterns. For example, the following extraction pattern is defined for the item "date of period":

$$\text{from } \langle\text{start date}\rangle \text{ to } \langle\text{end date}\rangle$$

Intuitively, the extraction patterns summarize the common ways an item or field may be described in a news article. During extraction, the pattern above will serve as a template for matching against a sentence which is potentially pertinent to the item concerned (i.e., "date of period") and provide a guideline to isolate the information items of interest, which in this case are ⟨ *start date* ⟩ and ⟨ *end date* ⟩.

C. Extraction Algorithm

After classifying a news article into one of the five categories, the corresponding schema is retrieved and the information extraction is performed using the algorithm in Fig. 6. Note that if no matching items can be found in step 2 and step 3, the sentence S will be simply discarded.

VI. System Evaluation

To evaluate the performance of FIDS, a total of 1257 online news articles are collected from the Web sites of two local newspaper agencies, namely, South China Morning Post and Hong Kong Standard, which are the only two English newspapers in Hong Kong (they are at the addresses http://www.scmp.com and http://www.hkstandard.com respectively). These news articles are manually classified into the five categories: company performance, economy, merger and acquisition, financial services and products, and securities.

A. Classification Performance

We split the classified news articles into two sets, namely a training set and a testing set. The training set contains 1006 articles whereas the testing set contains 251 articles. The training news articles are first used to train the classification knowledge learning subsystem. After training, the system is used for conducting classification on the articles in the testing set. The two term weighting schemes for text representation as discussed in Section III are implemented, and the resulting classification performances are compared. In Table II, the results obtained by using the TF scheme are shown. The system has an overall precision of 73.7%. Table III shows the classification rule learned for the category *securities*.

Table IV depicts the classification performance when the TF-IDF scheme is used. As shown, the overall precision is now improved to 82.3%. With the TF-IDF scheme, term frequencies are normalized to prevent the keywords appearing in a long article from dominating the text representation. The features of an article can therefore be more accurately captured by the representation. As a result, classification performance is improved.

Table V depicts the five classification rules as induced for the category *securities*. As shown, the first rule is the same as the one induced when the TF-scheme is used (cf. Table III), while the others are newly learned. The four new rules provide a more precise classification concerning the category *securities*. For example, Rule 3 will assign an article to the category *securities* if the article possesses the feature *chip* (say, when the word *blue-chips* is present in the article), which when the TF-scheme was adopted, would have been classified wrongly into another category. In fact, by comparing Tables II and IV, one can see that the addition of these new rules has contributed to a substantial improvement in the classification performance of the category *securities*.

In order to gauge our classification performance under some existing machine learning methods, we conduct another experiment on classification using a well known decision tree learning algorithm known as C4.5 [35]. The same set of labeled news articles are used to train the classification knowledge learning subsystem. The TF-IDF weighting scheme is employed for the article representation. The classification accuracy for the same set of testing articles is shown in Table VI. The result indicates that our classification performance is slightly better than C4.5. All categories have a higher accuracy except for the category *services and products*.

B. Extraction Performance

The 47 testing articles in the category *company performance* are then selected for evaluating the performance of the information extraction subsystem. For example, given the article in Fig. 7 which appeared in a local newspaper, the system produces the output in Fig. 8 which shows some of the items of information extracted from the article.

To evaluate the extraction performance, each of the 47 articles is examined manually to identify the set of relevant items that should be extracted. This set is commonly called the key. Each article is then processed by FIDS. Performance is measured by comparing the items extracted by FIDS against the key of the article.

Two metrics are employed which are commonly used in the area of information retrieval [39]. The first one is known as *precision* which is defined as the ratio of the number of items correctly extracted to the number of items the system extracted. The second metric is known as *recall* which is defined as the ratio of items correctly extracted to the number of items in the article. We compute the precision and recall of each article as well as the average of these metrics across all articles.

Table VII depicts the extraction performance of each article and the average performance. In the table, the first column lists the unique identification code given to each article. The column labeled by "items" denotes the number of items in an article which are identified (manually) as pertinent to the category *company performance* (i.e., the size of the key), whereas the one labeled by "extracted" represents the number of items in the key which are actually extracted by FIDS. As shown, our extraction subsystem achieves an average precision of 0.92 and an average recall of 0.58.

C. Information Enquiry

The data extracted from the articles will be stored in a database. The Information Enquiry Subsystem provides an interface for users to retrieve the data from the database. Fig. 9 depicts a sample screen of this subsystem. The user can specify some selection criteria such as company name and date range. The system will retrieve all data records satisfying the criteria (note that these data records may be "contributed" by more than one article which may possibly belong to different categories). Besides, the user can specify a set of fields to be displayed in the output. This example demonstrates a scenario where a user wishes to retrieve the company performance data of the company Hong Kong Electric and display all fields in the output. Fig. 10 shows a sample output of this enquiry.

VII. Conclusion

In this paper, we have presented the design of FIDS which can summarize online financial news articles automatically. Compared with the previous approaches, the system is unique in the sense that it addresses multiple domains simultaneously, which are all concerned with financial news. In other words, they are interrelated. This has the advantage that related information appearing in articles of different domains can be cross-validated, integrated and cross-referenced. Preliminary evaluation reveals that FIDS has very encouraging performance.

For future work, we plan to extend the system along two directions. First, the extraction schema in the current implementation are manually designed. The process is tedious and demands much expert's knowledge. More importantly, it is difficult to justify the completeness of the schema produced in general. Alternatively, the extraction schema can in fact be learned from the articles automatically, as demonstrated by AutoSlog [36], the LIEP system [18], and CRYSTAL [42]. In this way, the portability of FIDS can be greatly enhanced which makes it easier to adapt the system to other new domains.

On the other hand, the experimental results have revealed that although the prototype of FIDS exhibits a very high precision rate of over 91%, there is still much room for improvement in the recall rate. A probable explanation is that, in a real article written in a natural language, the same item of information can be reported/described in virtually an infinite number of ways. Doubtlessly, the current extraction algorithm, employing mainly pattern-matching techniques, is simply too rigid to cope with this flexibility. As a result, much useful information is missed, leading to a low recall rate. In view of this, we are currently exploring the use of recurrent neural networks to implement the extraction subsystem. We hope that the robust processing characteristic of neural networks can to a certain extent help to alleviate the problem.

References

[1] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *Proc. 21st Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 1998, pp. 37–45.

[2] C. Apte, F. Damerau, and S. Weiss, "Automated learning of decision rules for text categorization," *ACM Trans. Inform. Syst.*, vol. 12, no. 3, pp. 233–251, 1994.

[3] K. D. Bollacker, S. Lawrence, and C. L. Giles, "CiteSeer: An autonomous Web agent for automatic retrieval and identification of interesting publications," in *Proc. 2nd Int. ACM Conf. Autonomous Agents*, 1998, pp. 116–123.

[4] R. Brandow, K. Mitze, and L. F. Rau, "Automatic condensation of electronic publications by sentence selection," *Informing Processing and Management*, vol. 31, no. 5, pp. 675–686, 1995.

[5] B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, and D. Rus, "Mobile agents for distributed information retrieval," in *Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet*, M. Klusch, Ed. Berlin, Germany: Springer-Verlag, 1999, pp. 355–395.

[6] B. Cestnik, "Estimating probabilities: A crucial task in machine learning," in *Proc. Eur. Conf. Artif. Intell.*, London, U.K., 1990.

[7] N. Chinchor, "Overview of MUC-7," in *Proc. Seventh Message Understanding Conf. (MUC-7)*, 1998.

[8] W. T. Chuang and J. Yang, "Extracting sentence segments for text summarization: A machine learning approach," in *Proc. 23rd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2000, pp. 152–159.

[9] P. Clark and T. Niblett, "The CN2 induction algorithm," *Mach. Learn.*, vol. 3, pp. 261–283, 1989.

[10] M. Costantino, R. G. Morgan, R. J. Collingham, and R. Garigliano, "Natural language processing and information extraction: Qualitative analysis of financial news articles," in *Proc. IEEE/IAFE Conf. Computat. Intell. for Financial Eng. (CIFEr)*, 1997, pp. 116–122.

[11] C. Fox, "Lexical analysis and stoplists," in *Information Retrieval: Data Structures & Algorithms*, W. Frakes and R. Baeza-Yates, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1992, pp. 102–130.

[12] W. B. Frakes, "Stemming algorithms," in *Information Retrieval: Data Structures & Algorithms*, W. B. Frakes and R. Baeza-Yates, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1992, pp. 131–160.

[13] G. L. Gentili, M. Marinilli, A. Micarelli, and F. Sciarrone, "Text categorization in an intelligent agent for filtering information on the Web," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 15, no. 3, pp. 527–549, 2001.

[14] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, "Summarizing text documents: Sentence selection and evaluation metrics," in *Proc. 22nd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 1999, pp. 121–128.

[15] Gossip. [Online]. Available: http://www.tryllian.com/index3.html.

[16] J. B. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson, "Fastus: A cascaded finite-state transducer for extracting information from natural language text," in *Finite-State Language Processing*, E. Roche and Y. Schabes, Eds. Cambridge, MA: MIT Press, 1997, pp. 383–406.

[17] E. Hovy and C. Lin, "Automated text summarization in SUMMARIST," in *Advances in Automatic Text Summarization*, I. Mani and M. T. Maybury, Eds. Cambridge, MA: MIT Press, 1999, pp. 81–94.

[18] S. B. Huffman, "Learning information extraction patterns from examples," in *Connectionist, Statistical and Symbolic Approaches to Learning in Natural Language Processing*, S. Wermter, E. Riloff, and G. Scheler, Eds. Berlin, Germany: Springer-Verlag, 1996, pp. 246–260.

[19] P. Jacobs and L. Rau, "SCISOR: Extracting information from on-line news," *Commun. ACM*, vol. 33, no. 11, pp. 88–97, 1990.

[20] A. Jennings and H. Higuchi, "A Personal News Service Based on a User Model Neural Network," *IEICE Trans. Inform. Syst.*, Mar. 1992.

[21] T. Joachims, "A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization," in *Proc. 14th Int. Conf. Mach. Learn. (ICML-97)*, 1997, pp. 143–151.

[22] ——, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. 10th Eur. Conf. Mach. Learn. (ECML-98)*, 1998, pp. 137–142.

[23] M. Klusch, *Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet*. Berlin, Germany: Springer-Verlag, 1999.

[24] T. Kohonen, *Self-Organizing Maps*. Berlin: Springer-Verlag, 1997.

[25] K. Lang, "Newsweeder: Learning to filter netnews," in *Proc. 12th Int. Conf. Mach. Learn. (ICML-95)*, 1995, pp. 221–339.

[26] W. Lehnert, J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan, and S. Goldman, "UMASS/HUGHES: Description of the CIRCUS system used for MUC-5," in *Proc. Fifth Message Understanding Conf. (MUC-5)*, San Francisco, CA, 1993, pp. 277–291.

[27] D. D. Lewis, R. E. Schapire, J. P. Callan, and R. Papka, "Training algorithms for linear text classifiers," in *Proc. 19th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 1996, pp. 298–315.

[28] Y. H. Li and A. K. Jain, "Classification of text documents," *Comput. J.*, vol. 41, pp. 537–546, 1998.

[29] D. Marcu, "The rhetorical parsing of unrestricted texts: A surface-based approach," *Comput. Linguist.*, vol. 26, no. 3, pp. 395–448, 2000.

[30] K. McKeown and D. R. Radev, "Generating summaries of multiple news articles," in *Proc. 18th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 1995, pp. 74–82.

[31] D. Mladenić, "Feature subset selection in text learning," in *Proc. 10th Eur. Conf. Mach. Learn. (ECML-98)*, 1998, pp. 95–100.

[32] M.-F. Moens and J. Dumortier, "Automatic abstracting of magazine articles: The creation of 'Highlight' abstracts," in *Proc. 21st Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 1998, pp. 359–360.

[33] J. Mostafa, S. Mukhopadhyay, W. Lam, and M. Palakal, "A multilevel approach to intelligent information filtering: Model, system, and evaluation," *ACM Trans. Inform. Syst.*, vol. 15, no. 4, pp. 368–399, 1997.

[34] M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Mach. Learn.*, vol. 27, pp. 313–331, 1997.

[35] J. R. Quinlan, "Constructing decision tree," in *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufman, 1993, pp. 17–26.

[36] E. Riloff, "An empirical study of automated dictionary construction for information extraction in three domains," *Artif. Intell.*, vol. 85, pp. 101–134, 1996.

[37] E. Riloff and W. Lehnert, "Information extraction as a basis for high-precision text classification," *ACM Trans. Inform. Syst.*, vol. 12, no. 3, pp. 296–333, 1994.

[38] A. Salminen, J. Tague-Sutcliffe, and C. McClellan, "From text to hypertext by indexing," *ACM Trans. Inform. Syst.*, vol. 13, no. 1, pp. 69–99, 1995.

[39] G. Salton and M. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw Hill, 1983.

[40] G. Salton and C. Bucklet, "Term weighting approaches in automatic text retrieval," *Inform. Process. Manag.*, vol. 24, pp. 512–523, 1988.

[41] R. F. E. Sutcliffe, "Representing meaning using microfeatures," in *Connectionist Approaches to Natural Language Processing*, R. G. Reilly and N. E. Sharkey, Eds. Hillsdale, NJ: Lawrence Erlbaum, 1992, pp. 49–73.

[42] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert, "CRYSTAL: Inducing a conceptual dictionary," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, 1995, pp. 1314–1319.

[43] S. Soderland, "Learning information extraction rules for semi-structured and free text," *Mach. Learn.*, vol. 34, pp. 233–272, 1999.

[44] T. Strzalkowski, G. Stein, J. Wang, and B. Wise, "A robust practical text summarizer," in *Advances in Automatic Text Summarization*, I. Mani and M. T. Maybury, Eds. Cambridge, MA: MIT Press, 1999, pp. 137–154.

[45] Y. Yang and J. Pedersen, "A comparative study on feature detection in text categorization," in *Proc. 14th Int. Conf. Mach. Learn. (ICML)*, 1997, pp. 412–420.

[46] M. Wooldridge and N. Jennings, "Intelligent agents: Theory and practice," *Knowl. Eng. Rev.*, vol. 10, no. 2, pp. 115–152, 1995.

An Intelligent System for Failure Detection and Control in an Autonomous Underwater Vehicle

N. Ranganathan, Minesh I. Patel, and R. Sathyamurthy

***Abstract*—Autonomous underwater vehicles (AUVs) have been used extensively in deep sea research. The AUVs are preferred to remote operated vehicles (ROVs) due to their low cost and efficiency. Failure detection and control is an important issue in maintaining the stability of an AUV. In most AUVs, the vehicle resurfaces in the event of minor failures such as in the depth sensor, the inclinometer, etc. This paper proposes an intelligent system for failure detection and control in AUVs where the vehicle could continue exploration in case of minor failures in the sensors and control surfaces. The intelligent system, based on the model proposed in [12], integrates the adaptability of an artificial neural network (ANN) and the inferencing ability of a fuzzy rule based expert system on a single VLSI chip. The associative function of the ANN is used to recognize and detect the failures by observing the various changing parameters of the dynamic vehicle. The inferencing ability of an expert system suggests ways to control the failure and indicates the subsequent status of the vehicle. The entire system could be used as a low level diagnoser in an overall control system for AUVs.**

***Index Terms*—ANN, automomus underwater vehicles (AVV), expert system, intelligent systems, remote operated vehicles (ROV).**

I. Introduction

Significant research is being conducted to increase the autonomy of underwater robotic vehicles. The vehicles remotely operated by human operators have several restrictions such as the limited operating range of the vehicle due to the physical limits of its communication cable and

Manuscript received January 8, 1998; revised August 21, 2000 and October 18, 2001. This paper was recommended by Associate Editor P. K. Willett.

N. Ranganathan and R. Sathyamurthy are with the Center for Microelectronics Research, Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620 USA (e-mail: ranganat@csee.usf.edu).

M. I. Patel is with Honeywell International, Clearwater, FL 33764-7290 USA.

Publisher Item Identifier S 1083-4427(01)11294-4.