

FNDS: a dialogue-based system for accessing digested financial news [☆]

Kwok Cheung Lan, Kei Shiu Ho ^{*}, Robert Wing Pong Luk, Daniel So Yeung

Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China

Received 4 November 2003; received in revised form 22 December 2004; accepted 22 December 2004

Available online 26 January 2005

Abstract

Electronic financial news available on the Internet contains a wealth of information useful for business decision-making. However, as this information is both qualitative and existent in huge volumes, it is very inefficient to digest manually. This paper presents a prototype system called FNDS, which automatically digests financial news by extracting important information from the articles and using this information to fill in pre-defined templates. A unique feature of FNDS is that users can access the extracted information through an interactive dialogue-based interface. This has the advantage that if users do not know exactly what information is required, the system will provide feedback to help them to formulate the information requirement incrementally.
© 2005 Elsevier Inc. All rights reserved.

Keywords: Information extraction; Financial news; Dialogue processing

1. Introduction

Electronic financial news available on the Internet (in online news websites, discussion groups, etc.) contains a wealth of information useful for business decision-making (e.g. stock price forecasting). However, as this information is both qualitative and existent in huge volumes, it is very inefficient to digest manually. Previously, various methods have been proposed to tackle this information overload problem. Among them, text filtering (Lang, 1995; Mostafa et al., 1997; Allan et al., 1998) aims at selecting the right documents for the user based on the user's preferences, as modeled by a profile. Text filtering allows the available documents to be matched with the profile so that only those documents of interest

to the user are returned. In general, the profile may be defined by user-specified keywords, or the system may infer the profile from documents that have been identified as interesting to the user. In the latter case, the system is usually adaptive: it can detect changes in the user's interests automatically and fine-tune itself accordingly to improve its performance. For example, SIFTER (Mostafa et al., 1997) employed a Bayesian-based shift detection model to track changes in the user's interests. When a shift occurred, the system would re-learn the user's profile automatically.

Taking one step further, a text summarization system digests documents on behalf of its user (Mani, 2001). Specifically, a document is analyzed (e.g. by using statistical or natural language processing techniques) to obtain a gist, which contains only the document's relevant information. The gist, commonly known as the summary, is returned to the user rather than the document itself. Recently, a series of Document Understanding Conferences (DUC) have provided forums for competitive evaluations of text summarization systems (NIST, 2003a). Early text summarization systems

[☆] A preliminary version of this article was presented at the 8th International Conference on Applications of Natural Language to Information Systems (NLDB 2003).

^{*} Corresponding author. Tel.: +852 2766 7286; fax: +852 2170 0116.

E-mail address: cksho@comp.polyu.edu.hk (K.S. Ho).

adopted the extraction-based approach in which a document was summarized by the selection of salient text spans, such as sentences (Goldstein et al., 1999) or passages (Strzalkowski et al., 1999). The salience of a text span was defined by criteria such as its position in the document, its length, or by other statistical measures (e.g. inverse document frequency Salton and McGill, 1983). Recently, we have seen employed more sophisticated methods which exploit the discourse structure of texts. For example, Marcu (2000a) used cue phrases and other linguistic features to derive the rhetorical structure tree of a document, from which an “importance score” was assigned to each text span. Text spans with a high importance score were included in the summary. Despite its simplicity, summaries produced by the extraction-based approach are often lengthy and incoherent, since the text spans they extract may actually be dispersed throughout the document and their contents may be unconnected. Various techniques have been proposed for revising the extracts in order to regenerate a more coherent summary. Such techniques range from simple repair methods applied at the sentence level (also called simple coherence smoothing) (Nanba and Okumura, 2000; Knight and Marcu, 2002) to approaches that involve full revision of the summary (Mani et al., 1999).

An alternative approach to the automatic digestion of documents is information extraction, where important items of information are extracted from a document, which are used to fill in pre-defined templates. The extracted information may then be used to generate the summary (e.g. by using pre-defined sentence patterns (Saggion and Lapalme, 2000)). Alternatively, the extracted information may be stored in a database for later access (Lam and Ho, 2001). The Message Understanding Conferences (Chinchor, 1998) reported a variety of approaches to information extraction. In general, the extraction templates can be generated in two ways, either being designed by human domain experts (Saggion and Lapalme, 2000; Lam and Ho, 2001) or being induced for a new domain automatically through some kind of learning mechanism, as demonstrated by systems like CRYSTAL (Soderland et al., 1995) and AutoSlog-TS (Riloff, 1996a). This second method reduces the considerable amount of knowledge engineering work otherwise involved in designing the extraction templates and increases the portability of the system. However, previous studies (Riloff, 1996b) have reported that the use of designed templates usually results in better extraction performance than does the use of learned templates.

This paper reports on a prototype system known as Financial News Dialogue System (FNDS). FNDS extracts important information from electronic financial news articles and uses this information to fill in designed templates. Users can access this information by posing questions to the system through a dialogue-based inter-

face. The questions can be posed in natural language, which allows the user to flexibly define the required information (Girardi and Ibrahim, 1995). The approach is similar to the task addressed by the question-answering (QA) track of the TREC conferences (NIST, 2003b), where answers to questions were provided by analyzing a large collection of articles, using a variety of techniques such as pattern-matching and shallow text analysis (Prager et al., 2000), information extraction (Srihari and Li, 1999), and natural language processing (Litkowski, 2000). Unlike in TREC’s QA, in FNDS, the answers are not retrieved directly from the articles, but from the database tables that have been filled in during the extraction stage by mapping the user’s questions to SQL queries (Hendrix et al., 1978; Sethi, 1986; Abreu et al., 2002). FNDS allows users and the system to communicate interactively. This has the advantage that if the user does not know exactly what information is required, such that a question cannot be answered, the system will provide feedback to help the user to formulate the information requirement incrementally. Compared with other similar approaches (Araki et al., 2001; Gatus and Rodríguez, 2002), in FNDS, the conversation between the user and the system is more flexible both because it is not constrained by a rigid *plan* (like the menus and forms in VoiceXML (Araki et al., 2001)) and because the user has the initiative during the interaction. This makes the system more robust and usable.

The rest of the paper is organized as follows. Section 2 provides an overview of the design of FNDS. Section 3 describes the information extraction sub-system. Section 4 describes the dialogue sub-system that is used to access the extracted information. Section 5 offers our conclusion, highlighting the future directions of our work.

2. Overview of FNDS

Fig. 1 shows the design of FNDS. It consists of two sub-systems: a back-end for processing online financial news articles and a front-end for accessing any financial information that has been digested.

FNDS digests a news article by extracting important information from it. The process involves several steps. First, the news article is divided into sentences, and each sentence is tokenized. Each token/word of the sentence is then assigned a part-of-speech (POS) tag. After that, shallow parsing is applied to identify the major syntactic constituents of the sentence, such as noun groups and verb groups. These constituents are further classified into one of several entity types, which carry semantic meanings related to the financial domain, e.g. company names and person names. Information is then extracted using templates. Intuitively, a template describes what information one can expect

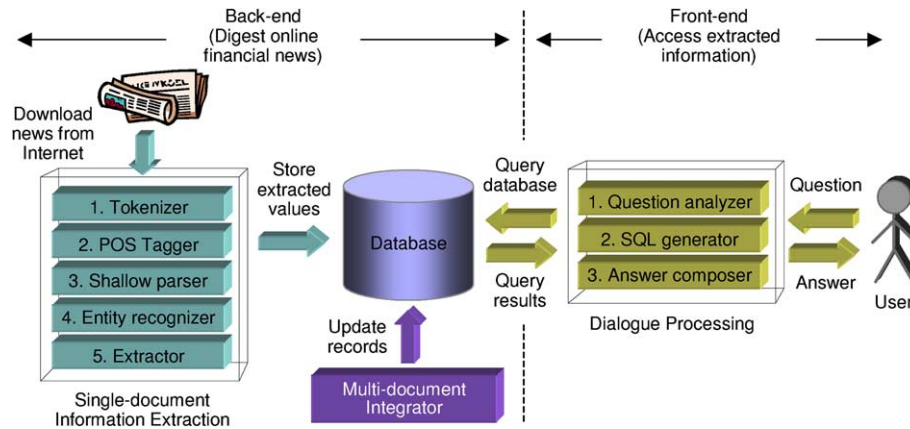


Fig. 1. System architecture of FNDS.

from an article and provides hints on how information can be located. During extraction, information is extracted from the article and is used to fill in the template. The information is then stored in a database for later access.¹ The templates are created by a human expert, based on a semantic network that captures the domain knowledge related to the financial domain. Later, users use natural language dialogues to access the extracted information. Specifically, the user and the system communicate interactively in a question-and-answer format: the user types in a question, and the system interprets the question and searches the database of extracted information to answer it. In case the question posed by the user is unclear, such that no answers can be found, the system will provide feedback to help the user re-formulate the question.

3. Digesting news articles

3.1. Preprocessing

Before information extraction begins, a series of preprocessing steps are carried out on each sentence of the article, including part-of-speech tagging, shallow parsing, and entity recognition.

3.1.1. Part-of-speech (POS) tagging

The lexical category of each word in the sentence is determined, using Brill's tagger program (Brill, 1995). The tagger uses the Penn TreeBank's tagset (Marcus et al., 1993), which consists of 36 part-of-speech (POS)

tags. For example, the sentence "The financial operator said Tom.com had a net profit of HK\$100 million." is tagged as follows:

The/DT financial/JJ operator/NN said/VBD Tom.com/
 NNP had/VBD a/DT net/JJ profit/NN of/
 IN HK\$100/\$ million/CD./.

(1)

where DT, JJ, NN, VBD, NNP, IN, \$, and CD are POS-tags corresponding to determiner, adjective, noun, verb, proper noun, preposition, dollar sign, and cardinal number, respectively.

The tagger is based on a supervised learning algorithm called transformation-based learning. This algorithm requires a large annotated corpus, referred to as the training corpus. Each word in the training corpus has already been tagged by a human expert. The algorithm first labels every word in the training corpus with its most probable tag (determined by the occurrence statistics of the word in the training corpus). For example, the word "book" can be tagged either as a noun or a verb. However, it is more often used as a noun than as a verb. Therefore, all occurrences of the word "book" in the training corpus are tagged as noun (i.e., NN) initially. Intuitively, this serves as a default rule for tagging the word "book".

As the default rules may tag some words incorrectly, transformation rules are formed which can override the default rules to improve tagging accuracy. The formats of the transformation rules are restricted by a set of pre-defined templates. Consider again the word "book". By examining the training corpus, one can see that the word "book" should be a verb and not a noun if it occurs after the word "to", as in "to book a room" (the POS-tag of "to" is TO). A transformation rule may thus be formed which will tag the word "book" as a verb (i.e., VB) if the previous word is tagged as TO, overriding the default rule which always tags "book" as a noun (see also Brill, 1995). This improves the overall

¹ Alternatively, the extracted information items may be fed as inputs to a natural language generator, whereby a summary of the original article (in the form of a passage) is generated (Dale et al., 1998). A summary may even be generated using information items extracted from multiple articles, thus achieving multi-document summarization (McKeown and Radev, 1995), which helps the user to integrate information from diversified sources.

tagging accuracy. In general, more than one transformation rule may be formed. However, only that rule which most improves the tagging accuracy at each step is adopted. The training corpus is then re-tagged, taking into account the newly found transformation rule. After that, another transformation rule is selected. This process is repeated until no further significant improvement in tagging accuracy is achieved. The resulting set of rules may then be used to tag unseen sentences.

3.1.2. Shallow parsing

After labeling each word with its POS-tag, the sentence is parsed syntactically. The purpose of this is to identify hierarchical relationships between the words in the sentence. For example, Fig. 2 shows the full parse tree of the sentence “The financial operator said Tom.com had a net profit of HK\$100 million”. Here, S, NP, CNP, VP, and PP are syntactic categories corresponding to sentence, noun phrase, common noun phrase, verb phrase, and prepositional phrase respectively.

Full parsing, however, is not suitable for FNDS largely because the construction of a syntactic parser requires a grammar of the underlying language. Since the news articles are written in free text, finding a gram-

mar that can “cover” all the sentences in the corpus is very difficult, if not infeasible. Even if such a grammar could be found, the number of grammar rules would be large, complicating the construction of the parser. More importantly, the use of such a large grammar could compromise the efficiency of parsing (it is well-known that full parsing is computationally expensive) and would seriously affect the overall processing efficiency of FNDS. Further, the output of full parsing is too elaborate and detailed for the purpose of FNDS, with some of the syntactic categories that it identifies being of little use to FNDS.

In view of these issues, the FNDS approach performs shallow parsing on the tagged sentence rather than a deep syntactic analysis. This approach divides the sentence into syntactically related groups of words, also called chunks (Abney, 1991; Zechner and Waibel, 1998). By definition, a word group or chunk is a linear group of neighbouring words in a sentence. Unlike syntactic categories (as used in full parsing), word groups are non-overlapping and non-recursive. Four types of word groups are recognized: noun groups (NG), verb groups (VG), proper name groups (PG), and cardinal groups (CG), as shown in Table 1. For example, the following word groups are found in the POS-tagged sentence (1):

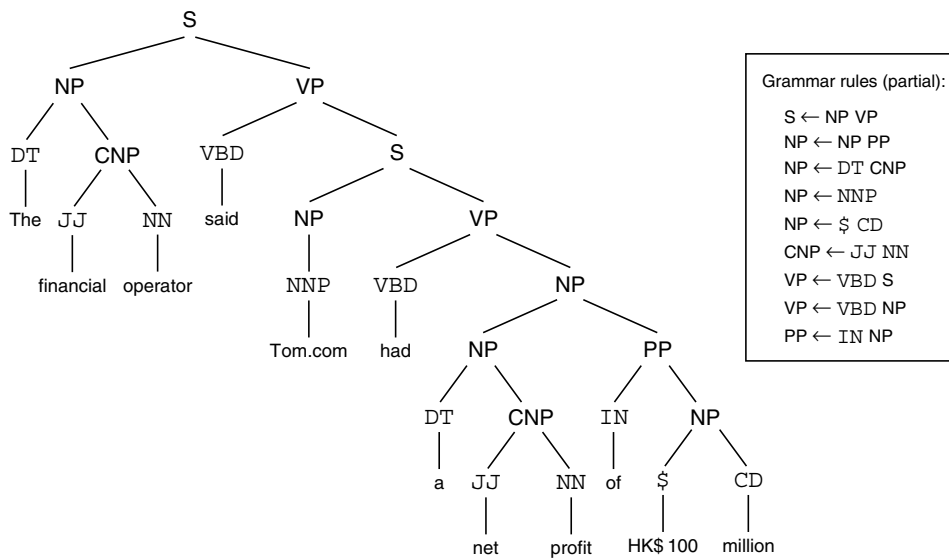


Fig. 2. Full parse tree of the sentence “The financial operator said Tom.com had a net profit of HK\$100 million”.

Table 1
Types of word groups recognized

Word group types	No. of rules	Examples of rules	Examples of word groups
Noun groups	25	‘DT’? ‘JJ’? ‘NN’? (‘NN’ ‘NNS’)	a/DT target/NN price/NN
Verb groups	12	‘MD’ ‘VB’	could/MD return/VB
Proper groups	2	‘DT’? ‘NNP’+	HSBC/NNP USA/NNP
Cardinal groups	9	\$ ‘CD’+	HK\$0.46/\$ billion/CD

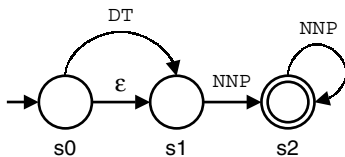


Fig. 3. Recognizer implementing the rule ‘DT’? ‘NNP’⁺ for identifying proper name groups.

[NG The financial operator] [VG said] [PG Tom.com]
 [VG had] [NG a net profit] of [CG HK\$100 million] (2)

In FNDS, shallow parsing is achieved by using a set of handcrafted rules (see Table 1) to identify the word groups, similar to the finite state parsing method of FASTUS (Hobbs et al., 1997). The rules are specified in the form of regular expressions from which a recognizer program is implemented.² The inputs to the recognizer are the POS-tagged sentences. Totally, there are 48 rules.

As an illustration, consider the rule ‘DT’? ‘NNP’⁺ as shown in Table 1. During shallow parsing, the recognizer scans the POS-tagged input sentence sequentially and tries to identify the proper name groups by performing state transitions, resembling the non-deterministic finite automata depicted in Fig. 3. For example, consider the sentence “HSBC USA announced interim results”. After POS-tagging, the sequence “HSBC/NNP USA/NNP announced/VBD interim/JJ results/NNS” is obtained. Initially, the recognizer is in s₀ (the start state). Since the sequence does not start with a determiner (i.e., DT), the recognizer transits to s₁. The first element of the sequence, “HSBC/NNP”, is read, which causes the recognizer to transit to s₂. Then, the second element “USA/NNP” is read, and the recognizer transits to s₂ (i.e. it remains in the same state). After that, there is no transition for the next tag (i.e., VBD). But since s₂ is designated as a final state, the recognizer declares that a proper name group [PG HSBC USA] is identified. The recognizer then returns to the state s₀ and the recognition process restarts from “announced/VBD”, trying to identify another proper name group.

When identifying word groups, the recognizer uses a *maximal matching* approach, matching as long a sequence of words/tags as possible. As in the preceding example, after reading the first element “HSBC/NNP”, the recognizer is in s₂ which is already a final state. It will not, however, declare that the proper name group [PG HSBC] is identified since the next element “USA/NNP” can still cause a transition. The recognizer continues to perform state transitions until it encounters the

VBD tag. From this point, it cannot proceed further and as a result, the proper name group [PG HSBC USA] is identified rather than [PG HSBC].

3.1.3. Entity recognition

The parsed sentence is then subjected to entity recognition. In contrast to the previous steps, which are mainly domain-independent and syntax-directed, this step looks for semantic entities that are related to the financial domain, such as company names, person names, time expressions, and monetary expressions. Six types of entities are of interest: PERFORMANCE, COMPANY, POSITION, PERSON, TIME, and AMOUNT. Table 2 summarizes their definitions. For example, from the word groups identified in (2), four entities are located:

[The financial operator]_{PO} [VG said] [Tom.com]_{CO}
 [VG had] [a net profit]_{PE} of [HK\$100 million]_{AM} (3)

Syntactically, each type of entities belongs to a specific word group(s). For example, a PERSON entity must be a proper name group (PG). To identify the entities, a dictionary is maintained, where each entity type (except AMOUNT) is associated with a set of key words or phrases. For example, the set of keywords for the entity type PERFORMANCE includes words such as earnings, profits, and shares. During operation, each word group located is matched against the dictionary, to determine its entity type. A word group X is classified as an entity of type Y if X contains a key word/phrase of Y. Since a cardinal group (CG) must be an AMOUNT entity, no keywords are needed for the entity type AMOUNT.

3.2. Information extraction

FNDS digests an article by extracting important information items to fill in handmade templates. Fig. 4 shows a template for extracting information related to the profit of a company. By definition, each template consists of two parts: attribute–value pairs and extraction patterns. The attribute–value pairs list the important information that one can expect to find in a so-called *standard* article. For example, when one reads a news article that talks about the profits of a company, one expects to see, among other things, information like the company’s name, the specific type of profit (e.g. gross or net), and the amount of profit or loss. Intuitively, articles of this type will share these features. Such regularities are captured by the attribute–value pairs.

Extraction patterns specify how information can be found in the article. Instead of relying on a simple keyword-matching approach (Lam and Ho, 2001), an extraction pattern involves lexical, syntactic, and semantic constraints which have to be satisfied before the

² The recognizer can in fact be automatically generated using tools such as *lex* (Levine et al., 1992).

Table 2
Types of entities recognized

Entity type	Word group type	No. of keywords	Examples
PERFORMANCE (PE)	NG	9	Earnings per share, net profit
COMPANY (CO)	PG	677	Samsung, HSBC USA
POSITION (PO)	PG or NG	7	CEO, chairman
PERSON (PN)	PG	234	Richard Li
TIME (TI)	NG	19	Last year, the first quarter
AMOUNT (AM)	CG	N.A.	1 million, 15%

Note: Since a CG must be an AMOUNT, no keywords are needed.

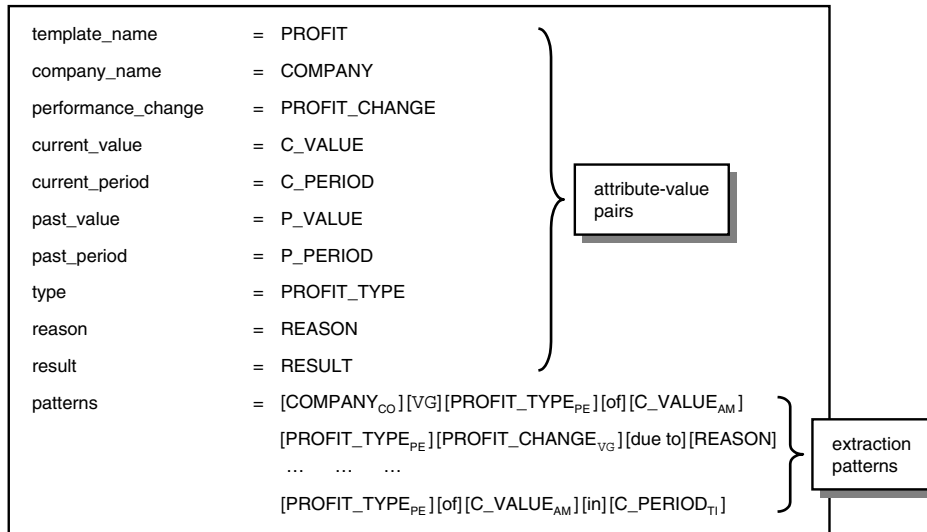


Fig. 4. Example of a template for extracting information related to the profit of a company.

pattern can be applied. Consider the example shown in Fig. 5. The extraction pattern consists of several parts. First, there are three slots: COMPANY, PROFIT_TYPE, and C_VALUE. They are exactly the attribute-value pairs in the template in Fig. 4. During extraction, the slots are to be filled using text excerpts. As depicted, each slot is associated with an entity type, indicating that it can be filled by an entity of the designated entity type only. Besides the slots, there are two triggers: the verb group (VG) and the word “of”. Recall that after the previous processing stages, each sentence is transformed into a sequence consisting of syntactic and

semantic constituents, namely, word groups and domain-specific entities. An extraction pattern is said to be successfully matched with a sentence if the designated word groups and entity types as well as the required trigger words and phrases as specified in the pattern can be found in the sentence.

Consider the example in Fig. 5. After the initial processing steps, the sentence “Tom.com has a net profit of HK\$100 million”. is transformed into the sequence [Tom.com]_{CO} [VG has] [a net profit]_{PE} [of] [HK\$100 million]_{AM}, which can be matched with the pattern. Extraction is then carried out, in which the entities

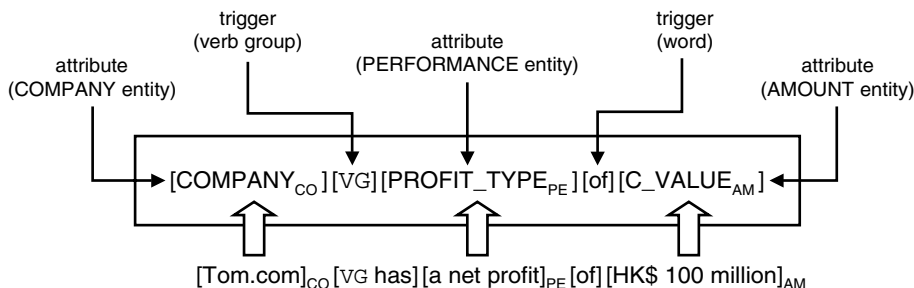


Fig. 5. Example of an extraction pattern.

-
1. repeat for each sentence S
 2. repeat for each template T
 3. match S against the extraction patterns of T to extract the relevant information from S
-

Fig. 6. Algorithm for extracting relevant information from a news article.

“Tom.com”, “a net profit”, and “HK\$100 million” are used to fill in the slots COMPANY, PROFIT_TYPE, and C_VALUE respectively. The extracted information is then recorded in the template, which will be stored in the database for later access.

Fig. 6 shows the algorithm for extracting relevant information from the sentences of a news article, thereby allowing the attributes of a template to be completed.

3.3. Extraction performance

The extraction performance of FNDS was evaluated using 109 unseen articles collected from the Web sites of two local newspaper agencies. The articles were written by different authors and the articles had an average length of about 500 words.³ Each article was manually examined to identify the set of relevant items that should be extracted, commonly referred to as the *answer key*. Each article was then processed by FNDS. Performance was measured by comparing the items extracted by FNDS against the answer key. The results are shown in Fig. 7(a). Line “I” denotes the number of items in the answer key of the article, whereas “E” represents the number of items extracted by FNDS. Line “C” shows the number of items, among those extracted, that belong to the answer key (i.e. items that were correctly extracted). The precision and recall of extraction of each article (Salton and McGill, 1983) were computed as follows:

$$\text{Precision} = \frac{\text{Number of correctly extracted items}}{\text{Number of extracted items}} = \frac{C}{E} \quad (4)$$

$$\text{Recall} = \frac{\text{Number of correctly extracted items}}{\text{Number of items in the article}} = \frac{C}{I} \quad (5)$$

Fig. 7(b) shows the precision and recall of each article. Overall, FNDS achieved an average precision of 0.84 across all articles, which is comparable to that of related systems. However, the average recall was only 0.54, which is relatively low. In general, the recall performance may be improved by using more extraction

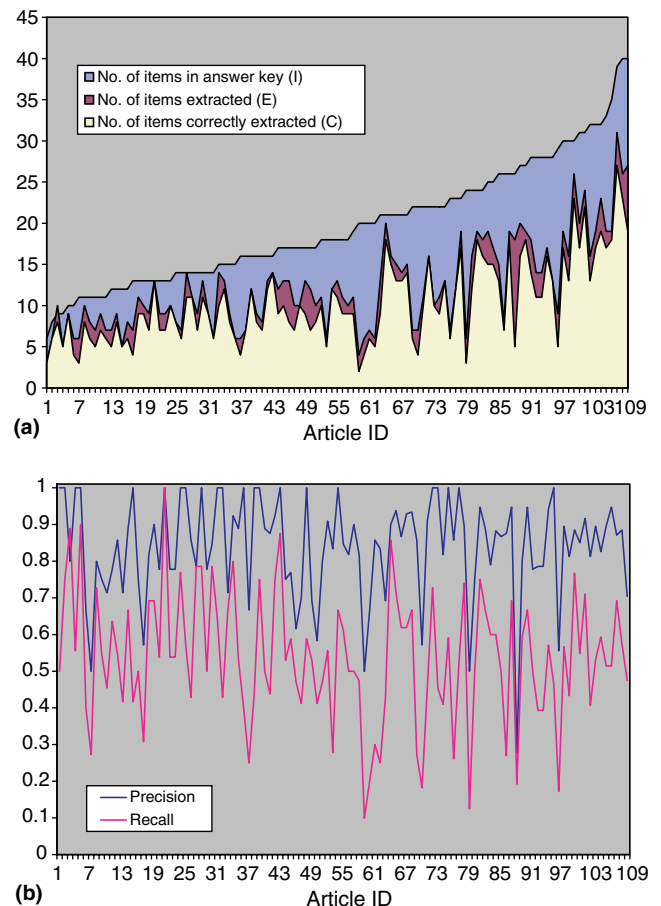


Fig. 7. Extraction performance of FNDS.

patterns, allowing the extraction of more target information items, but this is usually achieved at the expense of lower precision.

Further examination of the results reveals that in many of the failures, the sentence containing the missed information item was long. This meant that the triggers and the target information item were distantly separated. Since the extraction patterns were of limited length, they failed to match with the target sentence in those cases. Yet, there were also cases where the missed information item could actually be located in a sentence neighbouring the one containing the triggers. Since, however, the extraction patterns were defined only over a single sentence, they failed to spot the target information items. Although the performance of FNDS may be slightly improved by designing more extraction patterns

³ A self-created corpus was used because existing benchmark collections of articles for evaluating information extraction systems (like those used in the MUC conferences) do not exactly match with our target domain (i.e. financial news).

and/or by extending the scope of application of the patterns to multiple sentences, the fundamental cause of the failures is the inflexibility of the pattern-matching approach. In order to achieve more significant performance improvement, what is required is some kind of *deeper* understanding of the news articles, for example, discourse processing (Marcu, 2000b).

4. Accessing extracted information

After extraction, stored information can be accessed through natural language dialogues with the user and the system interacting in a question-and-answer format: the user types in a question, grammatical or ungrammatical, complete or incomplete, and the system analyzes the question and searches the database of extracted information to answer it (see Fig. 8).

The question-and-answer format provides two benefits. First, it enables the users to locate information more flexibly, by allowing them to specify the information needed in a virtually unlimited number of ways. This can make the system more robust and usable. Second, users may not always know exactly what information they require. FNDS allows a user to start with an imprecise question and by interacting with the system, clarify the question incrementally.

Consider the following examples:

- (a) What is the profit of HSBC? (6)
 (b) How well is HSBC doing?

In both cases, the system is unable to answer the question. Instead of simply rejecting the question, the system

provides feedback to help the user to define the required information. For example, in (a), the meaning of the word “profit” is too general, since there are different types of profit, such as net or gross profit. The system may respond to such a query by displaying a message like “Do you mean net profit, gross profit, or something else?”. The user can then clarify the question. The question in example (b), however, is too open-ended and its meaning is not well-defined. The system may only be able to provide a response such as “Would you clarify your question?”. Note that it may take several iterations before the system can clearly identify a question, of a type that it can then proceed to find an answer to.

Fig. 9 shows the state diagram for dialogue processing in FNDS. In general, there are several steps in answering a question. First, the question is analyzed. The purpose is to find the meaning of the question. If the question is ill-posed such that it cannot be interpreted precisely, the system will notify the user, possibly providing advice to help the user “refine” the question (response 1 in Fig. 9). On the other hand, if it is possible to interpret the question precisely, the system will formulate an SQL query to access the database of extracted information. If the target information can be accessed, it will be used to formulate an answer to be returned to the user. If it cannot be accessed, a response will be displayed to notify the user (response 2 in Fig. 9).

4.1. Question analysis

The purpose of question analysis is to determine the meaning of the question. First, the question is classified into one of three types: yes–no questions, wh-questions,

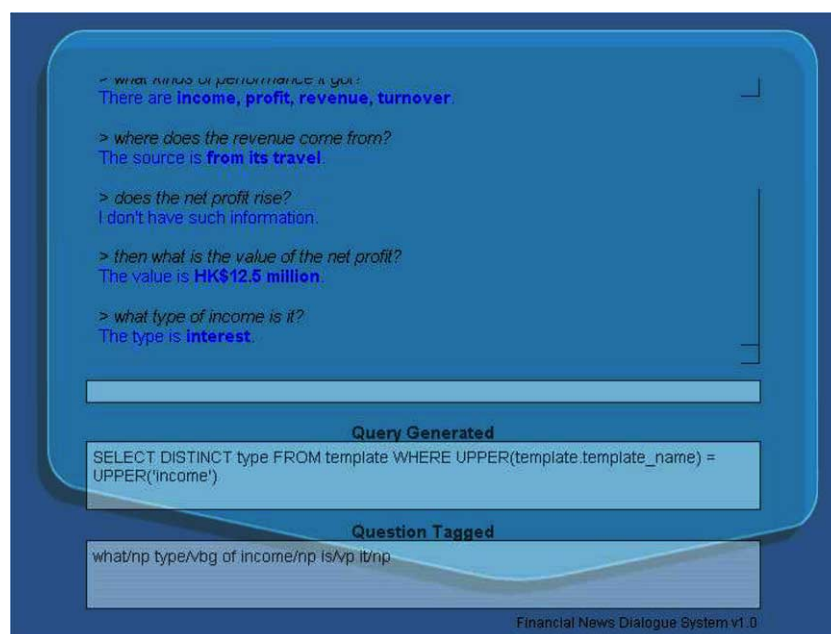


Fig. 8. Sample screen: accessing financial information using dialogues.

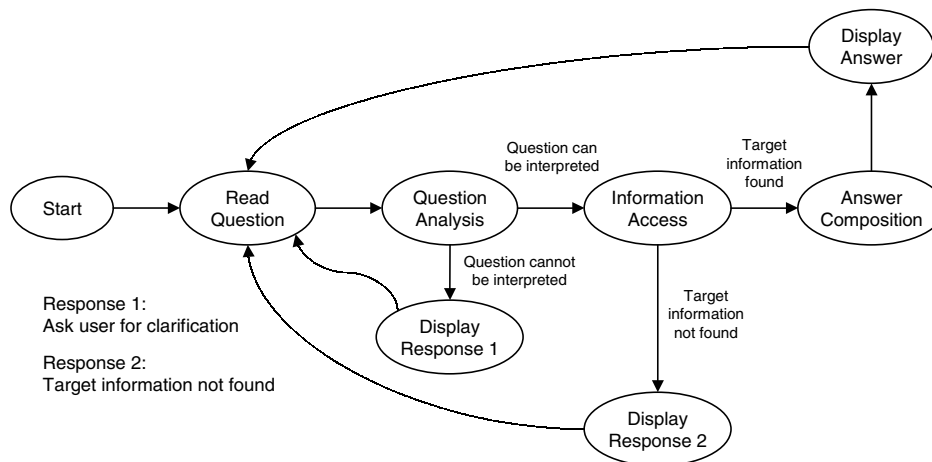


Fig. 9. State diagram: dialogue processing in FNDS.

and unclassified. A yes–no question such as “Did the net profit increase?” calls for a yes or no answer from the system, depending on whether the fact stated in the question is true or not. In FNDS, a yes–no question is identified by checking whether the question starts with an auxiliary (i.e. a word with the POS-tag AUX), e.g. *can*, *is*.

The second category of question, a wh-question, such as “What is the net profit of HSBC?”, usually starts with an wh-word (e.g. *which*, *who*, *why*) or a question phrase (e.g. *which person*). In general, a user poses a wh-question to request information about an entity mentioned in the question. For example, in the preceding question, users would expect the system to state the value of the net profit of HSBC.

Unclassified questions are those that the system fails to classify as one of the preceding two types. In such a situation, the system will generate a response asking the user to clarify the question.

Having classified a question, the system will proceed to determine the focus of the question, that is, the target information the user is looking for. Towards this aim, entity recognition is carried out on the question. Let us consider again the question “What is the net profit of HSBC?”. Two entities are identified, namely, “net profit” which is a PERFORMANCE entity, and “HSBC” which is a COMPANY entity. Both are potentially the focus of the question. Further examination, however, reveals that the entity “HSBC”, being a proper name group, is an item extracted from a certain news article and has been used to fill in a slot of a template. Logically, it cannot be the focus since the user is assumed not to have read the news article before raising the question. On the other hand, the entity “net profit”, being a common noun phrase, is an attribute in a template to be filled in during extraction. Further, semantically, it *agrees* with the wh-word “what” in the question, since “what” may refer to a PERFORMANCE, COMPANY, or POSITION entity. Hence, the entity “net profit” should be

the focus, and the user is taken to be asking for the net profit of a company called HSBC.

4.2. Information access

If the focus of the question can be determined, an SQL query will be formulated based on the result obtained in the question analysis stage. The query is then submitted to the database of extracted information items to access the target information. For example, the question “What is the net profit of HSBC?” gives rise to the following query:

```

SELECT value
FROM template
WHERE template_name = 'profit'
AND type = 'net'
AND company_name = 'HSBC'
  
```

Occasionally, the word or phrase used to describe the focus of a question may be literally different from the attribute name used by the system. For example, the user may ask the system about the net profit of HSBC by posing the question “How much does HSBC earn?” instead of “What is the net profit of HSBC?”. In that case, the system may have difficulty in understanding the question. To tackle this problem, a set of key words or phrases is associated with each attribute in a template. For example, the attribute “net profit” will be associated with words like “earn” and “gain”. The meanings of these words are closely related to the meaning of the attribute “net profit”. In this way, given the question “How much does HSBC earn?”, the system will return a response such as “Do you mean net profit?” to help the user re-formulate the required target information.

The result obtained by executing the SQL query is returned to the user using simple sentences generated by sentence patterns. If no matching results can be found from the database, meaning that the target information

has not been extracted, a standard response, such as “I don’t have such information”, will be returned to the user.

4.3. Performance evaluation

The performance of the dialogue processing sub-system was evaluated as follows. A group of 15 human subjects were invited to use the system. Each subject was given the same list of 10 target items to be filled in using information accessed from FNDS. The news articles in which these target items appeared had been processed by FNDS beforehand, and we ensured that they had been successfully extracted by the system during the extraction stage. In other words, each target item had been stored in the database, which could be accessed by posing an *appropriate* question to the system. These questions are shown in Table 3.

The evaluation was monitored by an assistant, who would describe the target items to the subject in Chinese. The standard Chinese translation of each item as it appears in local Chinese newspapers and finance dictionaries was used. The subject was then required to formulate a question in English for submission to the system. If an answer was returned and it was the target item, the subject would proceed to the next item. Otherwise, the subject would be asked to refine the question and re-submit it to the system. This process continued until the correct answer was returned, or until the subject had asked five questions, the maximum allowed number.

Note that when formulating a question, one must use the standard English translation of the target item, but as described in Section 4.2, FNDS associates each target item with a set of words or phrases in addition to the standard translation. The meanings or usages of these words or phrases are closely related to the target item. For example, the target item “net profit” (which is the standard English translation) is associated with “earning” and “gain”. If the subject uses one of these alternative words or phrases to access the target item, the system will return a response (e.g. “Do you mean net profit?”) to help the subject re-formulate the question.

For each subject, we counted the number of target items obtained as well as the average number of questions that had to be submitted in order to get a correct answer. The results are summarized in Table 4, where each entry represents the number of questions a subject posed before a particular item of information could be found. Overall, subjects were able to access an average of 8.87 items and for each item, an average of 13.3 subjects were able to obtain the correct answer within five trials. Access to a target item required an average of about 1.73 questions. In other words, the majority of subjects were able to obtain the target information in one or two trials. Considering that the subjects’ questions used a variety of wordings, the results reflect that the dialogue sub-system effectively facilitates user-access to information from FNDS.

One can observe that the results for accessing item 7 and item 1 are the worst. Item 7 could be accessed by only nine subjects (60%). Here, many of the subjects who had difficulty in accessing the target item used the terms “profit” or “revenue” instead of the *expected* term “earn”. In finance, these terms represent different concepts, although their literal meanings or general usages are close. As a result, the system was unable to return the required information. Item 1 could be accessed by only 10 subjects (66.7%) and it required as many as 2.9 trials in order to obtain the correct answer. The poor performance on item 1 may have been because it was the first item in the list and the subjects were inexperienced in using the system at the beginning of the evaluation.

Obviously, for the case of item 7, it will be helpful if the system can feedback a response such as “Do you mean earn?” to the user. In order to achieve that, links have to be established between the term “earn” and the other semantically similar terms, like “revenue” and “profit”. In the current implementation of FNDS, however, such kind of links must be set up manually. Unavoidably, some useful links, including those between “earn”, “profit”, and “revenue”, will be missed. A better approach is to make use of tools such as

Table 3

The set of information items to be filled in by the subjects for evaluating the dialogue sub-system of FNDS

Target items	Sample questions	Correct answers
1. Whether or not the loss of <i>Telecoms</i> has increased	Does the loss of <i>Telecoms</i> increase?	Yes
2. Turnover of <i>Henderson Cyber</i>	What is the turnover of <i>Henderson Cyber</i> ?	HK\$15.72 million
3. Previous turnover of <i>Henderson Cyber</i>	What was the past value of turnover of <i>Henderson Cyber</i> ?	HK\$3.41 million
4. Expected net profit of <i>China Eastern</i> in the last year	How much net profit did <i>China Eastern</i> expect to gain last year?	HK\$129.5 million
5. Net profit of <i>Hongkong.com</i> in this year	What is the net profit of <i>Hongkong.com</i> this year?	HK\$15.93 million
6. Whether or not the shares of <i>Samsung</i> has increased	Is there any growth of shares of <i>Samsung</i> ?	Decrease
7. Earnings of <i>Samsung</i> in 2001	How much did <i>Samsung</i> earn in 2001?	3.3 trillion won
8. Revenue of <i>Tom.com</i> in the second quarter	What was the revenue of <i>Tom.com</i> in the second quarter?	HK\$77 million
9. Net loss of <i>Sunday</i> in the last period	How much did <i>Sunday</i> lose in the past?	HK\$114.5 million
10. Any changes of shares of <i>TVB.com</i>	Are there any changes of shares of <i>TVB.com</i> ?	Increase

Note: Company names are shown in *italics*.

Table 4
Evaluation results of the dialogue sub-system of FNDS

Subjects	Target information items										No. of items obtained
	1	2	3	4	5	6	7	8	9	10	
Subject #1	×	1	2	1	1	5	×	2	2	2	8
Subject #2	3	1	1	1	2	1	1	4	2	1	10
Subject #3	4	1	2	1	1	2	2	2	1	1	10
Subject #4	2	1	1	×	1	1	3	2	5	1	9
Subject #5	×	1	1	1	1	×	×	5	2	1	7
Subject #6	4	1	1	1	1	3	×	1	3	2	9
Subject #7	1	1	1	1	2	1	3	1	1	1	10
Subject #8	5	1	1	4	1	×	1	5	1	1	9
Subject #9	1	1	1	1	1	1	5	1	1	1	10
Subject #10	4	1	1	3	1	1	×	2	2	1	9
Subject #11	×	2	3	1	×	4	×	2	2	1	7
Subject #12	4	1	1	1	1	1	3	1	1	1	10
Subject #13	1	1	1	1	1	1	2	1	1	1	10
Subject #14	×	1	2	1	2	1	×	×	1	3	7
Subject #15	×	1	2	1	1	2	1	×	1	2	8
No. of subjects obtaining the item	10	15	15	14	14	13	9	13	15	15	
Average no. of trials to obtain the item	2.9	1.1	1.4	1.4	1.2	1.8	2.3	2.2	1.7	1.3	

Note: × denotes that the subject failed to obtain the answer within five trials.

WordNet (Fellbaum and Miller, 1998) or concept hierarchies (Sanderson and Croft, 1999), so that more complete linking between related concepts can be in place. In this way, more useful feedback can be provided to users to help them access information from the system. This will be part of our future work.

5. Conclusion and discussion

This paper has described a system called FNDS that can provide information related to the financial domain using items of information extracted from online news articles. FNDS is built using a set of proven information retrieval and natural language processing techniques. A unique feature of FNDS is that the users can access the extracted information through a dialogue-based interface, by posing questions written in natural language. This feature helps the users in two ways. First, it allows users to access the information more flexibly, by specifying the required information in an unlimited number of ways. Second, where users may be uncertain of their information requirements, the system will provide feedback to help the user revise the question, more precisely specifying the target information. Experimental evaluation reveals that the performance of the extraction sub-system and the performance of the dialogue sub-system are both satisfactory.

Currently, FNDS is only a prototype. Our future work will focus on three areas. First, we will seek to improve the extraction performance of the prototype system, especially the recall rate. Presently, it is possible to apply extraction patterns to single sentences only.

As a result, FNDS misses some target items. Consider the following example:

“ABC announced interim results yesterday. In the past six months, the net profit of the company was HK\$10.2 million.”

The current design of FNDS is unable to extract the target item “HK\$10.2 million” from the second sentence as the net profit of “ABC” because it is unaware that “the company” actually refers to “ABC”. A simple way to alleviate the problem is to perform coreference resolution (Soon et al., 2001). In this example, after substituting “ABC” for “the company”, it is possible to directly apply extraction patterns to the second sentence, extracting the target item “HK\$10.2 million”. More comprehensive solutions, such as discourse analysis (Marcu, 2000b) and cross-sentence semantic role labeling (Gildea and Jurafsky, 2002; Lan et al., 2004), may also be applied. These methods allow related sentences to be grouped together for analysis, whereby important events are identified from the article. The participants of the events are filled using information extracted from different sentences. To implement these methods, the extraction patterns have to be re-designed. The new extraction patterns will be applied to events rather than to individual sentences. In our future work, we will explore the effectiveness of these approaches in improving the extraction performance of FNDS.

A second area of potential improvement is in the dialogue-based interface, which we believe can be extended such that the user may request the system to generate a summary of the information accessed. Fig. 10 shows an example of a dialogue summary, which was produced using a prototype of the extended FNDS system. We

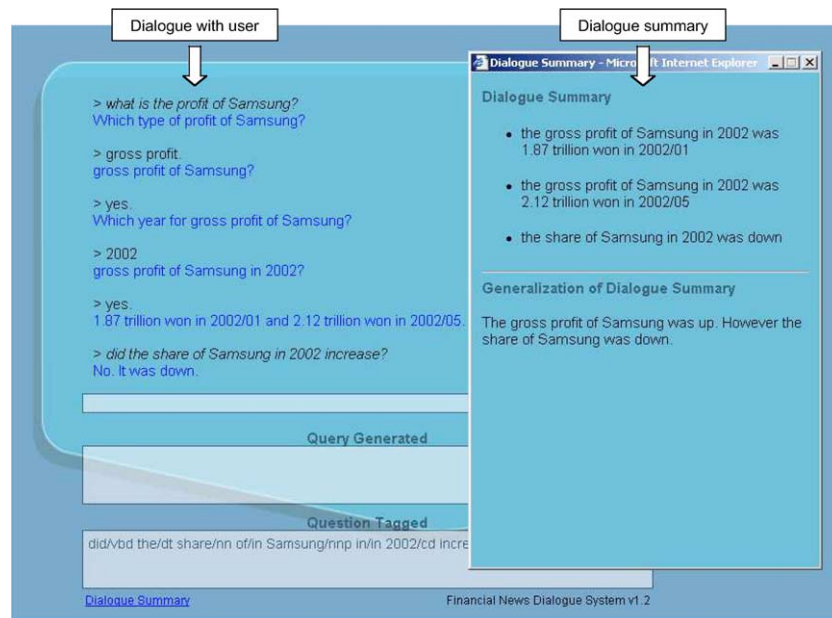


Fig. 10. Sample screen of a dialogue summary.

believe that such a summary is useful to the user in several ways. First, the summary can help the user to filter irrelevant answers. Second, the summary can organize the information accessed in a logical manner in cases where the number of items of information that have been accessed is large. For example, items of information about a particular company or person may be grouped together, or events may be presented in chronological order (Fig. 10).

In cases where the information spans several topics or domains, by employing context knowledge and/or ontological information, the system may report the hidden or potential associations between the accessed items of information, making the original information even more useful. This may also motivate the user to explore other information that has been overlooked before.

Our third area of research will be to apply our approach to other domains. Currently, FNDS can provide only financial information, but as many of its techniques, including part-of-speech tagging, shallow parsing, and dialogue processing, are domain-independent, its approach can readily be applied to other domains. All that is required is the redefinition of the extraction templates and the dictionary used for entity recognition. Currently, the extraction templates of FNDS are hand-made, since previous studies (e.g. Riloff (1996b)) have reported that templates created by domain experts usually lead to better extraction performance than learned templates. In our future work, the system will be extended to carry out domain-independent information extraction by inducing the extraction templates automatically through some kind of learning mechanisms. This can definitely broaden the applicability of the system.

We envision that the techniques underlying the design of FNDS will ultimately be applied to knowledge grid research (Berman, 2001; Zhuge, 2004). Assuming that heterogeneous sources of information are described using natural language-like metadata (similar to that proposed in (Di Felice and Fonzi, 1998)), it may be possible to automatically integrate and digest information/knowledge by applying various information extraction techniques (with the aid of an ontology of the relevant domain). A representation may thus be generated for specializing the digested knowledge, which can facilitate the retrieval of the knowledge (Zhuge and Liu, 2004). This synthesized knowledge may then be accessed through a dialogue-based user-interface (Zhuge, 2004) similar to FNDS. We believe that the interactive nature and robustness of the dialogue model of FNDS can enable the easy access to and discovery of knowledge, as well as facilitating communication between users for the purpose of knowledge exchange and management.

Acknowledgement

The work described in this article was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (PolyU 5085/99E).

References

- Abney, S., 1991. Parsing by chunks. In: Berwick, R., Abney, S., Tenny, C. (Eds.), *Principle-Based Parsing: Computation and Psycholinguistics*. Kluwer Academic Publishers, Dordrecht, pp. 257–278.

- Abreu, S., Quaresma, P., Quintano, L., Rodrigues, I., 2002. A natural language dialogue manager for accessing databases. In: Ranchhod, E., Mamede, N.J. (Eds.), *Proceedings of 3rd International Conference on Portugal for Natural Language Processing (PORTAL 2002)*. Springer-Verlag, Berlin, pp. 161–170.
- Allan, J., Rapka, R., Lavrenko, V., 1998. On-line new event detection and tracking. In: *Proceedings of 21st International ACM SIGIR Conference Research and Development in Information Retrieval*. pp. 37–45.
- Araki, M., Fujisawa, M., Nishimoto, T., Niimi, Y., 2001. Extracting domain knowledge for dialogue systems from unstructured web pages. In: *Proceedings of Pacific Association for Computational Linguistics 2001 (PACLING 2001)*.
- Berman, F., 2001. From TeraGrid to knowledge grid. *Commun. ACM* 44 (11), 27–28.
- Brill, E., 1995. Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging. *Comput. Linguist.* 21 (4), 675–686.
- Chinchor, N., 1998. Overview of MUC-7. In: *Proceedings of Seventh Message Understanding Conference (MUC-7)*.
- Dale, R., Eugenio, B.D., Scott, D., 1998. Introduction to the special issue on natural language generation. *Comput. Linguist.* 24 (3), 345–353.
- Di Felice, P., Fonzi, G., 1998. How to write comments suitable for automatic software indexing. *J. Syst. Software* 42 (1), 17–28.
- Fellbaum, C., Miller, G., 1998. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge.
- Gatius, M., Rodriguez, H., 2002. Natural language guided dialogues for accessing the web. In: Sojika, P., Kopeček, I., Pala, K. (Eds.), *Proceedings of International Conference Text, Speech and Dialogue (TSD 2002)*. Springer-Verlag, Berlin, pp. 373–380.
- Gildea, D., Jurafsky, D., 2002. Automatic labeling of semantic roles. *Comput. Linguist.* 28 (3), 245–288.
- Girardi, M.R., Ibrahim, B., 1995. Using English to retrieve software. *J. Syst. Software* 30 (3), 249–270.
- Goldstein, J., Kantrowitz, M., Mittal, V., Carbonell, J., 1999. Summarizing text documents: sentence selection and evaluation metrics. In: *Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 121–128.
- Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D., Slocum, J., 1978. Developing a natural language interface to complex data. *ACM Trans. Database Syst.* 3 (2), 105–147.
- Hobbs, J.B., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., Tyson, M., 1997. FASTUS: a cascaded finite-state transducer for extracting information from natural language text. In: Roche, E., Schabes, Y. (Eds.), *Finite-State Language Processing*. The MIT Press, Cambridge, pp. 383–406.
- Knight, K., Marcu, D., 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artif. Intell.* 139, 91–107.
- Lam, W., Ho, K.S., 2001. FIDS: an intelligent financial web news articles digest system. *IEEE Trans. Syst. Man Cybern. A* 31 (6), 753–762.
- Lan, K.C., Ho, K.S., Luk, R.W.P., Leong, H.V., 2004. Semantic role labeling using maximum entropy. In: *Proceedings of International Symposium on Computation and Information Sciences (CIS'04)*.
- Lang, K., 1995. Newsweeder: learning to filter netnews. In: *Proceedings of 12th International Conference on Mach. Learn. (ICML-95)*, pp. 221–339.
- Levine, J.R., Mason, R., Brown, D., 1992. *Lex & Yacc*. O'Reilly & Associates.
- Litkowski, K.C., 2000. Syntactic clues and lexical resources in question–answering. In: *Proceedings of 9th TExt Retrieval Conference (TREC-9)*. pp. 157–166.
- Mani, I., 2001. *Automatic Summarization*. John Benjamins, Amsterdam.
- Mani, I., Gates, B., Boledorn, E., 1999. Improving summaries by revising them. In: *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics*. pp. 558–565.
- Marcu, D., 2000a. The rhetorical parsing of unrestricted texts: a surface-based approach. *Comput. Linguist.* 26 (3), 395–448.
- Marcu, D., 2000b. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press, Cambridge.
- Marcus, M., Santorini, S., Marcinkiewicz, M., 1993. Building a large annotated corpus of English: the penn treebank. *Comput. Linguist.* 19 (2), 313–330.
- McKeown, K., Radev, D.R., 1995. Generating summaries of multiple news articles. In: *Proceedings of 18th International ACM SIGIR Conference Research and Development in Information Retrieval*. pp. 74–82.
- Mostafa, J., Mukhopadhyay, S., Lam, W., Palakal, M., 1997. A multilevel approach to intelligent information filtering: model, system, and evaluation. *ACM Trans. Inform. Syst.* 15 (4), 368–399.
- Nanba, H., Okumura, M., 2000. Producing more readable extracts by revising them. In: *Proceedings of 18th International Conference on Computational Linguistics (COLING-2000)*. pp. 1071–1075.
- NIST, 2003a. Document Understanding Conferences. Available from: <http://www.nlpir.nist.gov/projects/duc>.
- NIST, 2003b. TREC Question Answering Track. Available from: <http://trec.nist.gov/data/qa.html>.
- Prager, J., Brown, E., Coden, A., Radev, D., 2000. Question–answering by predictive annotation. In: *Proceedings of 23rd International ACM SIGIR Conference Research and Development in Information Retrieval*. pp. 184–191.
- Riloff, E., 1996a. Automatically generating extraction patterns from untagged text. In: *Proceedings of 13th National Conference on Artificial Intelligence (AAAI-96)*. pp. 1044–1049.
- Riloff, E., 1996b. An empirical study of automated dictionary construction for information extraction in three domains. *Artif. Intell.* 85, 101–134.
- Saggion, H., Lapalme, G., 2000. Summary generation and evaluation in SumUM. In: Monard, M., Sichman, J. (Eds.), *Proceedings of International Joint Conference: 7th Ibero-American Conference: on AI and 15th Brazilian Symposium on AI (IBERAMIA-SBIA 2000)*. Springer-Verlag, Berlin, pp. 329–338.
- Salton, G., McGill, M., 1983. *Introduction to Modern Information Retrieval*. McGraw Hill, New York.
- Sanderson, M., Croft, W.B., 1999. Deriving concept hierarchies from text. In: *Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 206–213.
- Sethi, V., 1986. Natural language interfaces to databases: MIS impact, and a survey of their use and importance. In: *Proceedings of 22nd Annual Computer Personnel Research Conference on Computer Personnel Research Conference (CPR'86)*. pp. 12–26.
- Soderland, S., Fisher, D., Aseltine, J., Lehnert, W., 1995. Crystal: inducing a conceptual dictionary. In: *Proceedings of 14th International Joint Conference on Artificial Intelligence*. pp. 1314–1319.
- Soon, W.M., Ng, H.T., Lim, D.C.Y., 2001. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.* 27 (4), 521–544.
- Srihari, R., Li, W., 1999. Information extraction supported question answering. In: *Proceedings of 8th TExt Retrieval Conference (TREC-8)*. pp. 186–195.
- Strzalkowski, T., Stein, G., Wang, J., Wise, B., 1999. A robust practical text summarizer. In: Mani, I., Maybury, M. (Eds.), *Advances in Automatic Text Summarization*. The MIT Press, Cambridge, pp. 137–154.
- Zechner, K., Waibel, A., 1998. Using chunk based partial parsing of spontaneous speech in unrestricted domains for reducing word error rate in speech recognition. In: *Proceedings of 17th International Conference on Computational Linguistics and 36th Annual*

Meeting of the Association for Computational Linguistics (COLING-ACL'98). pp. 1453–1459.

Zhuge, H., 2004. China's E-Science knowledge grid environment. *IEEE Intell. Syst.* 19 (1), 13–17.

Zhuge, H., Liu, J., 2004. Flexible retrieval of web services. *J. Syst. Software* 70 (1–2), 107–116.

Kwok Cheung Lan is a graduate student of the Department of Computing, The Hong Kong Polytechnic University. His research interests include dialogue processing and natural language understanding.

Kei Shiu Ho received his degrees of B.Sc., M.Phil. and Ph.D., all in computer science, from the Chinese University of Hong Kong. He joined the Department of Computing of The Hong Kong Polytechnic University in 1998, where he is now an assistant professor. His research interests are in collaborative computing, middleware, distributed systems, natural language processing, and neural networks.

Robert Wing Pong Luk is an IEEE senior member and an associate professor of the Department of Computing, The Hong Kong Polytechnic University. He serves as a program committee member for various conferences (e.g. ACM SIGIR, IRAL, IEEE NLPKE and NLDB), and he is a co-inventor of two US patent applications. He was a consultant for the bilingual law search system of the Hong Kong

judiciary, which used XML for mark up and XSL for parallel text rendering. His research is in the broad area of information retrieval, including indexing data structures and strategies, retrieval models, query expansion techniques and signal processing.

Daniel So Yeung (M'89-SM'99-F'04) received the Ph.D. degree in applied mathematics from Case Western Reserve University in 1974. In the past, he has worked as an Assistant Professor of Mathematics and Computer Science at Rochester Institute of Technology, as a Research Scientist in the General Electric Corporate Research Center, and as a System Integration Engineer at TRW. He was the chairman of the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, where now he is a Chair Professor. His current research interests include neural-network sensitivity analysis, data mining, Chinese computing, and fuzzy systems. He was the President of IEEE Hong Kong Computer Chapter, an associate editor for both *IEEE Transactions on Neural Networks* and *IEEE Transactions on SMC (Part B)*. He is a member of the Board of Governor for the IEEE SMC Society, and he has been elected the Vice President for Technical Activities for the same Society. He served as a General Co-Chair of the 2002–2004 International Conference on Machine Learning and Cybernetics held annually in China, and a keynote speaker for the same Conference. He leads a group of researchers in Hong Kong and China who are actively engaging in research works on computational intelligence and data mining.

His IEEE Fellow citation makes reference to his “contribution in the area of sensitivity analysis of neural networks and fuzzy expert systems”.