EVENT DETECTION DRIVEN APPROACH FOR EXTRACTING INFORMATION FROM INTERNET DOCUMENTS

Heng-Hsou Chang and *Yau-Hwang Kuo*
Institute of Computer Science and Information Engineering,
National Cheng Kung University,
Tainan, Taiwan, R.O.C
Email:changht@ismp.iie.ncku.edu.tw
kuoyh@ ismp.iie.ncku.edu.tw

Jang-Pong Hsu
Department of Finance,
Tainan Woman's College of Art & Technology,
Tainan, Taiwan, R.O.C
Email: hsujp@ismp.iie.ncku.edu.tw

ABSTRACT

In this paper, we propose event detection driven intelligent information extraction by using neural network paradigm. A back-propagation (BP) approach learning algorithm is adopted in the proposed system to train the event detector. In order to detect the potential events in documents effectively, we apply an part-of-speech tagger to help select nouns as feature words. Furthermore, unrelated nouns are filtered by the analysis based on document frequency distribution. At last, selected nouns are conceptualized into concepts using Ontology. These concepts are thought to characterize documents mostly. In the experimental results, we got high accuracy both in the inside testing and outside testing of Internet documents. By means of the well-trained event detector, the information extraction task can be applied in wider domains by the aid of intelligent event detection. This event detection technology is introduced in this paper for the application of E-mail delivery.

Keywords:
Information extraction, feature selection, neural network, knowledge representation.

1. Introduction

Since the Internet has been popular in these years, a lot of information can be spread over the world more easily. It seems that nowadays people get used to gathering news or fresh technologies through World Wide Web or E-mails. However, with the fully connected network from nation to nation, an incredible amount of information is rapidly produced in all fields every day. It results that people are not easily capable of assimilating incoming information as they were before. Information anxiety describes this kind of exactly situation people are suffering today. Therefore, with the increasing advance in computer speed, people try to employ computer to extract or retrieve useful information for them.

Information extraction systems [1-3] analyze unrestricted text in order to extract specific types of information. They do not attempt to understand all of the text in all input documents, but they do analyze those portions of each document that contain relevant information. Relevance is determined by pre-defined domain guidelines, which must specify, as accurately as possible, exactly what types of information the system is expected to find.

Very recently a new orientation for research in natural language processing has emerged under the name of information extraction. This area addresses information processing [1-3] needs associated with large volumes of text containing information in some domain of interest. For example, a stock analyst might want to track news stories about corporate mergers [4-7]. Or an intelligence analyst might need to track descriptions of terrorist events in some geographic region. An insurance adjuster might want to compile data from text-based hospital records.

Understanding language is a process that comes quite naturally to most humans. Unfortunately, the naturalness with which we understand language makes it very difficult to model this process in a computer. This difficulty in modeling tasks that seem relatively easy for humans can be seen in other areas of artificial intelligence research, notably research into the task of enabling a computer to recognize faces or creating a two-legged robot that can walk like a human. In fact, understanding language is the final goal that people want computer to do. However, it is still in progress.

The goal of an information extraction system is to identify references to the concept of interest for a particular domain. A key knowledge source for this purpose is a set of text analysis rules based on the vocabulary, semantic classes, and writing style peculiar to the domain. However, information extraction is usually limited resulting from not enough domain knowledge or being unable to identify domain. Therefore, in this paper, information extraction is aided by the event detection from a different aspect. With the help of event detection, it is believed that information extraction can be applied in wider domains more easily.

2.System overview of E-mail delivery application

In this paper, we use an E-mail management system for enterprises as case study to explain our concept and application. Therefore, we take a brief introduction to this system and see where the event-driven and ontology based information extraction process takes effect.

Figure 1 is our system overview of business application. This architecture illustrates the flow of how E-mails are dealt with and so on.

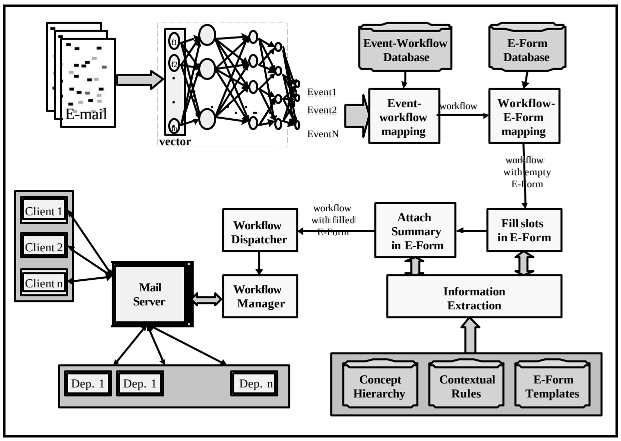


Figure 1. System overview of E-mail delivery.

In Figure 1, E-mails are taken as input into processing modules. We imagine that there are events, which represent some incoming motivation of business process, in each E-mail and these events should be detected in order to motivate some business workflow. Therefore, features selected from E-mails are fed into a neural network [8-10] module and then detected events become outputs. According to these identified events and event-workflow database in Figure 1, an event-workflow mapping process finds the appropriate workflow for an E-mail. Next, each workflow is responsible for a mission through several departments and should take special format of E-Form with itself, where E-Form is the missive in electronic form. Therefore, a workflow-E-From mapping process selects predefined E-Form template to the next process, primary information extraction process as shown in Figure 1.

The information extraction process provides important information to its upper sub-process in the figure to fill slots in the E-Form. The information extraction process relies on three domain knowledge databases. They are contextual rules, E-Form templates, and concept hierarchy. The contextual rules play an important role to our extraction task. The rules decide whether the information is important itself or even important related to others in some context.

When E-Form in the workflow is filled in all slots, a new E-mail will be duplicated and takes this E-Form as attachment. A workflow dispatcher will send the new E-mail to departments who are in the workflow. Meanwhile, the workflow manager will monitor whether the workflow is well run.

In this system, event detection using neural network is an important issue to our information extraction. It is well known that information extraction usually succeeds in domain-specific application and not suitable for a general-purpose domain. Hence, the events detected by the neural network restrict our domain by some related business processes. If the number of event increases, the domain of application is wider. These business processes are predetermined manually by human being towards huge number of incoming E-mails. Each E-mail is tagged with some desired events that appears in it and serves as a training sample to train a back-propagation neural network.

3. Event-driven Information Extraction

In this section, we attempt to propose an event-driven information extraction model to ease information extraction systems from the limitation that is only useful to specific domain.

The domain specific problem lies on the limited domains that an information extraction system can identify. Only under such limited domains, extracted information is thus valid and useful. Otherwise, it may be ineffective and less meaningful. Hence, to break the limitation, a domain classifier, which distinguishes one specific domain from others will help an information extraction system decide when an information extractor should be appropriate to be placed on. Hence, a domain classifier will tell which domain the information is in, and prepare an appropriate information extractor to extract information on that domain.

Traditionally, the issue of domain in information extraction problems is actually the issue of event. That is to say, to decide which domain an extractor is in, you must decide which event will be the target of extraction instead. Hereafter, the domain classifier is called an event detector. In addition, an event detector will detect which events are in an Internet document. Events are possibly coexistent in the same Internet document and the Internet documents can be classified into several domains. Consequently, our event detector receives an Internet document as input and results in several notifications about which events exist as output.

In Figure 2, event-driven information extraction model is illustrated. It is mainly divided into three phases.

- Feature selection.
- Training event detector.
- Creating extraction templates and contextual rules.

Since this model targets Internet documents, the Internet document access module is responsible for the access to Internet and for getting Internet documents back, such as web pages and E-mail messages. Therefore, this model will go through Internet protocols, such as POP3, HTTP, and GOPHER to acquire desired Internet documents for further purpose.

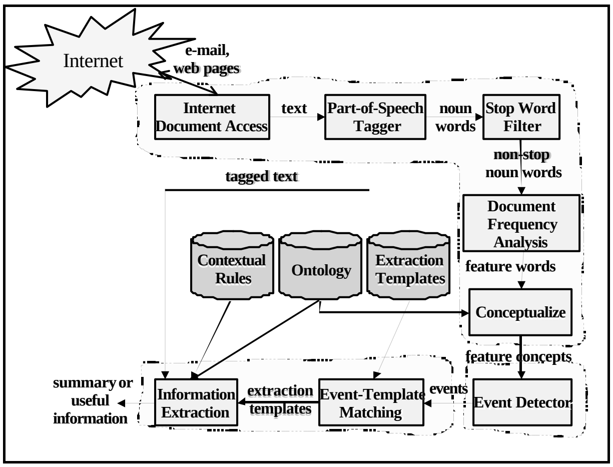


Figure 2. Event-driven information extraction model.

Internet documents will be previously tagged resulting in each word with its appropriate part-of-speech tag [11]. Generally speaking, the Internet documents are supposed to be composed by natural language text to express human's thought. It's believed in this paper that nouns will mostly decide concepts, which express human's thought. Therefore, the part-of-speech tagger is employed to select nouns as better features.

Stop words are also filtered to prevent noise from the further processing. After the stop words are filtered, the rest of non-stop noun words still can't be determined to be fully related to what human wants to express. According to human's writing habit, it is believed that too low or too high frequency of word's occurrence results from that the word itself is not important or representative.

In addition, many words may come from the same concept. While facing such text-based analysis, redundancy of these words of the same concept will bring foreseeable overload and redundancy to the final decision. Therefore, many words are conceptualized into concepts beforehand for further event detection.

An event is taken to represent each domain, which is concerned in the desired specific domain. An event detector thus plays the role to detect whether some events are in the document. It receives a vector of concepts from previous module, conceptualize module and makes an analysis towards these concepts. In the result, the detector will find which events are available in the document. It is allowable that more than one event can appear in the same document.

After the event detector detects potential events, those detected events can therefore direct the extraction script towards input document. Each event has its own extraction script and this kind of script is called extraction template. Extraction templates form strategies of how to extract information according to certain events. These strategies are contributed by contextual rules, which limit the distance and constraint of two information fields. In other words, each contextual rule specifies how far the contextual relationship takes effect and which concept the word should belong to.

The intelligent information extraction emphasizes the learning [9] of detecting occurrences of events. Before learning, Internet documents are randomly selected from each event on average to be training data. The training data is marked with desired outputs. These desired outputs are events that should be detected in the corresponding document. The whole flowchart of training phase is illustrated in Figure 3.

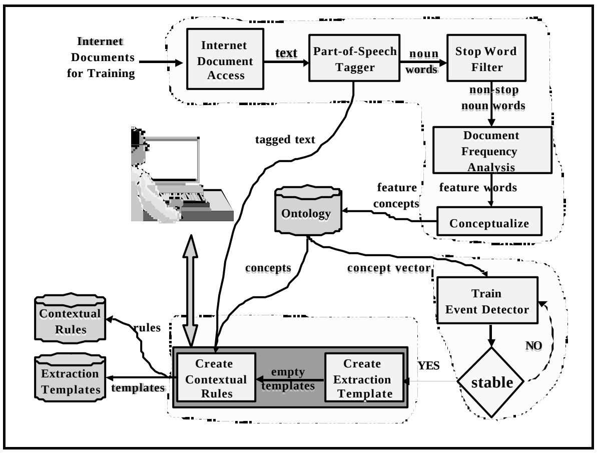


Figure 3. Training phase of event-driven information extraction model.

In this flowchart, the whole flow can be generally divided into three parts. The first part is feature selection, the second is training event detector and the last is manually creating extraction templates and contextual rules. These three parts listed below constitute the whole training phase.

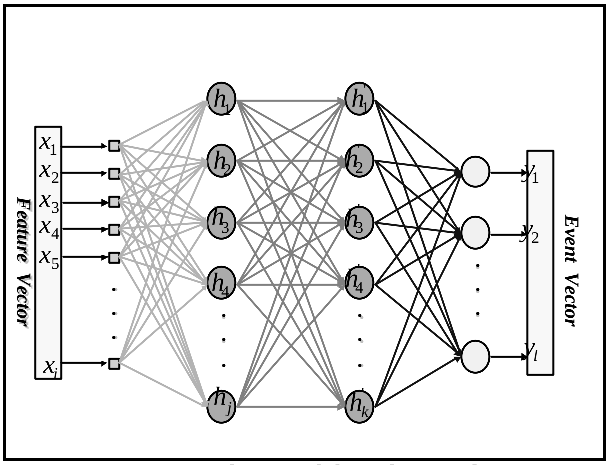


Figure 4. Neural network based event detector.

As to event detector, the back-propagation algorithm is proven to be a universal approximator. If event detector adopts back-propagation algorithm to learn about events, it is supposed to be successful, if the number of layers or number of hidden nodes is capable for the problem complexity. However, the more complicated the problem space is, the more neurons will be needed. It is well known that if the number of neurons is too large, the time spent for training will be awfully much. Therefore, there is one solution to avoid the disaster. That is, doing some reduction of inputs. In other words, features must be well selected to prevent redundancy from noise.

After features are selected using feature selection proposed in previous section, the neural network thus acquires a feature set, consisting of all feature concepts. This feature set is then fed into input nodes of neural network.

There are one layer of input nodes, two layers of hidden nodes, and one-layer output nodes in the designed network. The size of input layer (the size of feature set) is i, the size of hidden layers is j and k (j is designed to be equal to k), and the size of output layer is l as illustrated in Figure 4.

As in the forward pass, feature vector goes through the network and the network outputs an event vector. Meanwhile, desired output is compared with the event vector, producing an error vector. Error vector is thus, in the backward pass, used to calculate the weight correction. Each weight is adjusted according to the calculated weight correction.

While the total error energy is as small as possible, the neural network based event detector is supposed to be able to detect almost every potential events appeared in incoming E-mails.

4 .Ontology based conceptualization

It is said, "an ontology is a specification of a conceptualization." [12]

Many experienced knowledge engineers will agreed that an intelligent system must have not only a logic system, but also a good knowledge base. Without a knowledge base, a logic system can't know what's the background behind the problem. However knowledge representation is an important issue to construct the desired knowledge, a successful knowledge representation must be reasonable and be capable for solving how concepts should be expressed or connected. Ontology is a kind of knowledge representation, which is newly applied to knowledge engineering to help logic systems face more about the real world. Hereafter, a concept is considered a basic and primary entity in ontology. Generally speaking, ontology has several basic features as listed below.

- Generalization
- Specialization
- Thesaurus
- Conceptualization
- Classification

A concept hierarchy structure can characterize generalization, specialization, and classification. In fact, the well-known is-kind-of relation in ontology provides features of generalization and specialization and branches of inheritance provide classification. However, the feature of thesaurus will depends on additional relationship provided among a set of similar concepts in semantic distance. Eventually, the feature of conceptualization is based on the process of transforming real-world words into concepts.

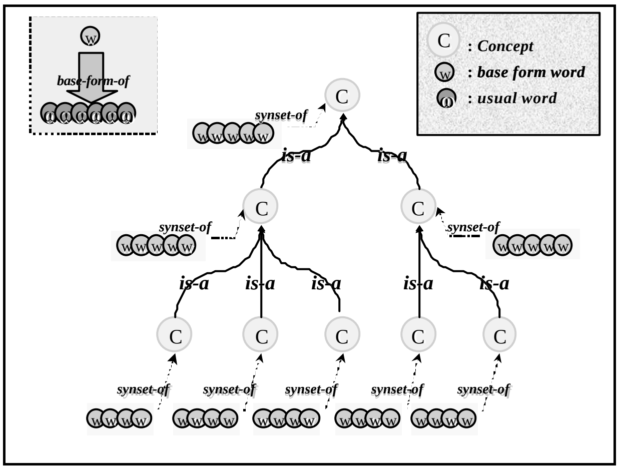


Figure 5. Ontology based concept hierarchy.

We adopt the ontology based concept hierarchy as structured in Figure 5. In the hierarchy, each node represents a concept. Concepts are organized with is-a (ISA) relation. Additionally, each concept has a set of base form words, which is a synonym set of (synset-of) the concept and each bas e form word has many variants. As dealing with the real words, each word is considered a variant of certain base form word.

The conceptualization algorithm below describes how to conceptualize a variant to a concept.

```
concept Conceptualize_Algorithm( word )
{
        find the word class of input word

        if( the word class is VERB )
        {
                base_form= VERB_BASE_FORM(word)
        } else if(word class is NOUN)
        {
                base_form= NOUN_BASE_FORM(word)
        } else if(word class is ADJ)
        {
                base_form= ADJ_BASE_FORM(word)
        }

        synset = SYNONYM ( base_form )
        concept = CONCEPTUALIZE ( synset )
        return concept;
}
```

In the conceptualization algorithm, part-of-speech tagger is employed to determine the word class of incoming word. Further each word goes to a base form word and each base form word belongs to a set of synonyms. Finally, a representative synonym is elected to be the concept of the synonym set (synset).

With the help of conceptualization algorithm, the number of features, which characterize the incoming E-mails can reduce in certain degree. Furthermore, the selected features are more meaningful and useful.

5. Experimental Results

In this section, experimental result is discussed in details.

In the beginning, the training data and testing data are listed in Table 1. There are five predefined events in Table 1. They are events of price, technology, visit, partner, and remove. Those events exist in training data of E-mails.

Table 1. Description of predefined events.

	Description
Price	Events, which talking about price, arguing about prices, or asking for product catalogue.
Technology	Events, which talking mostly in current technologies, especially that using lots of professional words.
Visit	Any event that talking about dating or asking about a date in the future.
Partner	Events, which talking about building the relationship between companies in business.
Remove	Events that senders asking to remove them from the mailing list.

In the experiment, there are totally 507 training documents and 413 testing documents from E-mails. The distribution of these E-mails is listed as in Table 2. These E-mails are handcrafted with desired events. Multiple events in E-mail are allowed.

Table 2. Distribution of both training and testing data.

	Price	Tech	Visit	Partner	Remove	Total
Training Data	211	64	86	54	8	507
Testing Data	121	16	45	170	3	413
Total	332	80	131	224	11	920

Once all documents are marked with desired events, an event detector can immediately be trained using these prepared E-mails. However, features must be well selected before training. As shown in Figure 6, features are selected step by step.

In Figure 6, the number of features goes from unknown amount to 3431 nouns. The nouns thus go from 3431 to 572 and finally there are 364 concepts left for characterizing those Internet documents.

At last, we need a method to evaluate the precision of event detection. Therefore, we define an evaluation function $EV(O_i, D_i)$ to score the result for ith document. The o_{ij} represents output value of jth output node in neural network and the d_{ij} represents desired output corresponding to o_{ij}..

When desired output d_{ij} is 0, the evaluation function is:

$$EV(o_{ij}, d_{ij}) = \begin{cases} 1 & , if\ -1.5 \le o_{ij} \le 0.4 \\ -0.1 & , otherwise \end{cases},$$

where 5 ≥ j ≥ 1.When desired output d_{ij} is 1, the evaluation function is:

$$EV(o_{ij}, d_{ij}) = \begin{cases} 1 & , if\ 0.6 \le o_{ij} \le 2.5 \\ -0.1 & , otherwise \end{cases},$$

$$PRE_i = \frac{1}{5}\sum_{j=1}^{5} EV(o_{ij}, d_{ij}),$$

$$PRE_{total} = \frac{1}{M}\sum_{i=1}^{M} PRE_i,$$

where M is the number of training or testing documents.

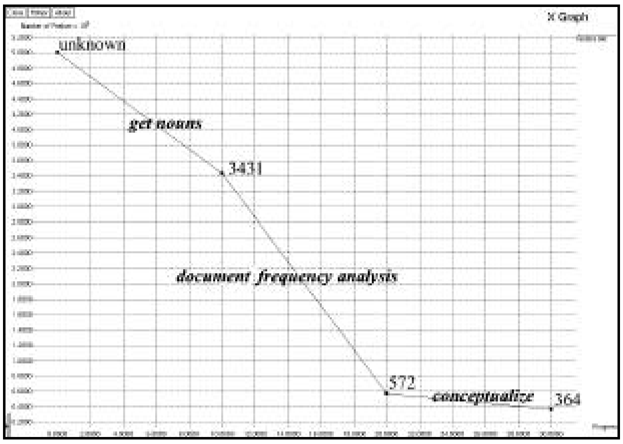


Figure 6. Feature selection step by step.

In Figure 7, the training doesn't go stable until 188,000 epochs. The detection precision in inside testing reaches 98.4% precision that time. Meanwhile, in recall phase, the detection precision for testing data reaches 87.2% precision.

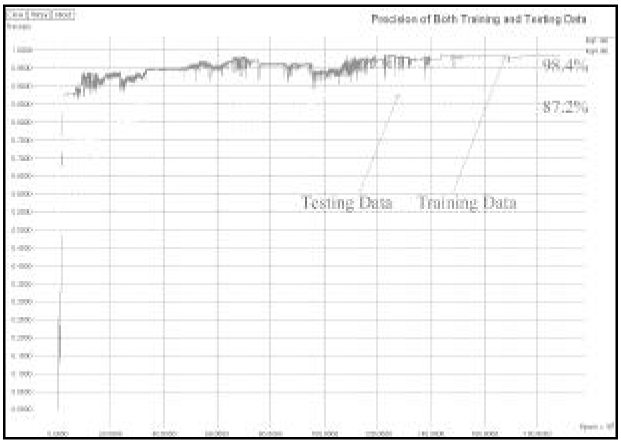


Figure 7.Precision of both inside and outside testing.

The Error Energy of back-propagation of training phase is shown in Figure 8. The error energy seems to struggle up and down for a long time. Put Figure 7 and 8 together in mind. It is straightforward that the error energy dangles between 0 and 160,000 epochs. It means that weights are regularly adjusted from time to time. It results that the precision also dangles in the period since the evaluation function defines a threshold for deciding whether an event occurs. Consequently, the precision doesn't converge smoothly in that moment. However, it is stable after about 16000 epochs of training. The final error energy reduces to 0.00787.

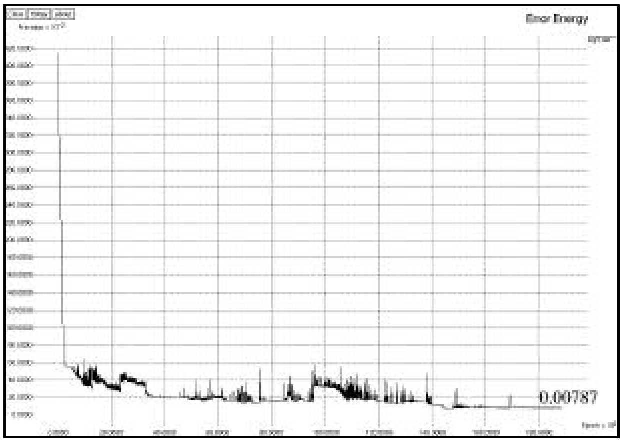
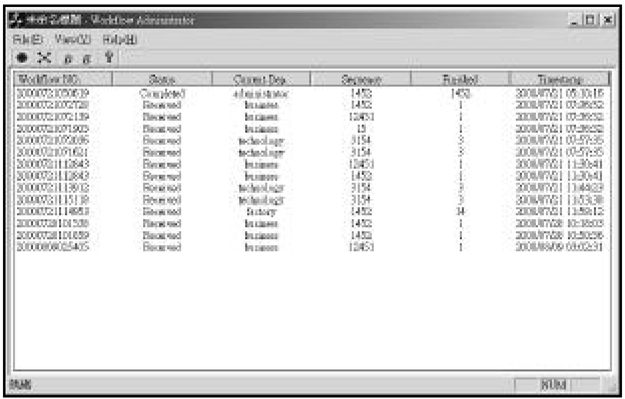


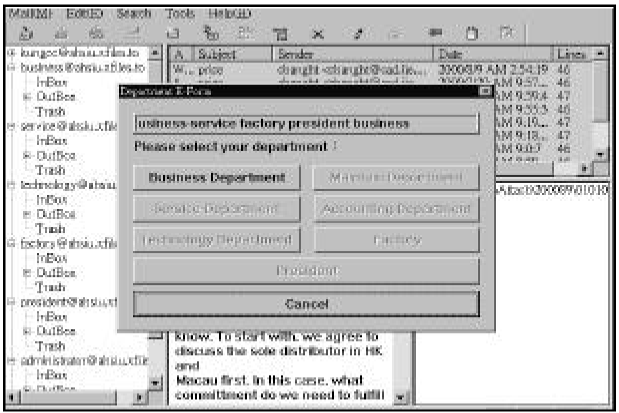
Figure 8. Error Energy of BP Training Phase.

With the high precision in both inside and outside testing, we also apply the event-detection driven information extraction to workflow control since each event can activate a predefined workflow. Therefore, in the Figure 9, a workflow manager is implemented to monitor every status of every active workflow. In this workflow manager, each row in the window represents an active workflow and administrator can see where each workflow has stepped forward so far. Each workflow is supposed to pass through several company departments. While certain department has finished its job, a workflow steps forward to the next assigned department.



Workflow NO.	Status	Current Dep.	Sequence	Finished	Timestamp
20000721050619	Completed	administrator	1452	1452	2000/07/21 05:10:16
20000721072720	Received	business	1452	1	2000/07/21 07:36:52
20000721072139	Received	business	12451	1	2000/07/21 07:36:52
20000721071905	Received	business	15	1	2000/07/21 07:36:52
20000721072006	Received	technology	3154	3	2000/07/21 07:57:35
20000721071621	Received	technology	3154	3	2000/07/21 07:57:35
20000721112643	Received	business	12451	1	2000/07/21 11:30:41
20000721112643	Received	business	1452	1	2000/07/21 11:30:41
20000721113912	Received	technology	3154	3	2000/07/21 11:44:23
20000721115119	Received	technology	3154	3	2000/07/21 11:53:38
20000721114853	Received	factory	1452	14	2000/07/21 11:59:12
20000728101538	Received	business	1452	1	2000/07/28 10:38:03
20000728101659	Received	business	1452	1	2000/07/28 10:50:56
20000809025405	Received	business	12451	1	2000/08/09 03:02:31

Figure 9. Workflow manager.



usiness service factory president business

Please select your department !

know. To start with, we agree to discuss the sole distributor in HK and Macau first. In this case, what commitment do we need to fulfill

Figure 10. E-mail agent with workflow handling.

In each department, we implement an E-mail agent, which sends or receives E-mails. This agent also handles the special task that is coming with some workflow notification specified in E-mail headers. While facing this workflow notification, the agent makes a popup window for dealing with the specified workflow.

Totally speaking, this system works fine and has been applied in commercial usage.

6. Conclusions

In this paper, we have proposed an intelligent event detector with automatic learning [8] capability for information extraction. This event detector has been tested for up to 98.4% precision in training phase and 87.2% precision in recall phase.

This event detector doesn't simply find keywords predefined manually. Instead, it employs neural network architecture and the learning algorithm is based on error correction of back-propagation algorithm. The experimental result shows that it really detects almost every potential event.

Generally speaking, the proposed model combines the technologies of information extraction and event detection to work effectively. Our event-driven information extraction model can be divided into information extraction module and event detection module. Typical information extraction doesn't take care of finding events. Therefore, it is usually limited for the use on specific domains resulting from being capable for few events. Many information extraction machines will sometimes claim that they only extract a limited number of events. This is the primary reason that contributes the domain-specific limitation. In summary, our event detector enhances the lack of typical IE machine towards wider domain.

7. Acknowledgment

This research was supported in part by the Development of Software Object Reuse Technology Project of Institute for Information Industry and sponsored by MOEA, R.O.C

8. References

[1] Jun-Tae Kim and Moldovan, D.I, "Acquisition of Linguistic Patterns for Knowledge-based Information Extraction," IEEE Trans. Knowledge and Data Engineering, vol. 7, no. 5, pp. 713 –724, 1995.

[2] Stephen G. Soderland, "Learning Text Analysis Rules for Domain-specific Natural Language Processing," PhD Thesis, CIIR Technical Report, 1996.

[3] Claire Cardie, "Domain-specific Knowledge Acquisition for Conceptual Sentence Analysis," PhD Thesis, CIIR Technical Report, 1994.

[4] James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan, " Detections, Bounds, and Timelines: UMass and TDT-3," in Proc. Topic Detection and Tracking Workshop (TDT-3), Vienna, Virginia, 2000.

[5] Yiming Yang, Jaime G. Carbonell, Ralf D. Brown, Thomas Pierce, Brian T. Archibald, and Xin Liu,

"Learning Approaches for Detecting and Tracking News Events," IEEE Intelligent Systems, vol. 14, no. 4, pp. 32 –43,1999.

[6] Ron Papka, "On-line New Event Detection, Clustering, and Tracking," Ph.D. dissertation, CIIR Technical Report TR99-45, 1999.

[7] James Allan, Victor Lavrenko, and Ron Papka, "Event Tracking," CIIR Technical Report, 1998.

[8] Tom M. Mitchell, "Machine Learning," MaGraw-Hill, 1997.

[9] Simon Haykin, "Neural Networks," Prentice-Hall, London, 1999.

[10] Erik Daniel Wiener, "A Neural Network Approach to Topic Spotting in Text," MS Thesis, 1995.

[11] Eric Brill, "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging Computational Linguistics, " 1995.

[12] http://www-ksl.stanford.edu/kst/what-is-an-ontology.html