Knowledge Base for a Disaster Management Dialogue System

Hao-Yung Chan[a], Cheng-Hsuan Yang[a], Meng-Han Tsai[a] and Shih-Chung Kang[a]

[a]Department of Civil Engineering, National Taiwan University, Taiwan
E-mail: hychan@caece.net, jason610155@gmail.com, menghan@caece.net, sckang@ntu.edu.tw

Abstract –

This research aims to develop a knowledge base for a disaster management question-answering dialogue system. The rapid growth of the amount of data has led to the variance of data in terms of their formats, sources, and attributes. Hence, the difficulties of decision makers to accomplish their missions accurately and efficiently have increased. To solve this problem, we developed a question-answering dialogue system for disaster management.

In our previous research, we found that the information most likely retrieved in response to a user's request can be determined by calculating the similarity between the keywords and the user's input in a handcrafted keyword–information mapping table. However, we also noticed that managing the mapping table was a tedious task. Moreover, for the inputs that had more than one keyword, the system was unable to provide integrated information. Therefore, we constructed a knowledge base to optimize the performance and maintainability of the system.

To build the knowledge base for disaster management, we designed the domain model by performing an abstraction on the knowledge of professional information providers and the required data on disaster management, while considering their source, certainty, and spatiotemporal features.

The query of requested information from the knowledge base is composed of mentioned entities in the user's input. For the dialogue system to recognize the entities, we applied entity recognition. The subtasks include segmentation, tagging, similarity calculation with the names of the entities in the knowledge base, and intent detection to determine the desired knowledge of the user.

Keywords –

Knowledge base; Disaster management; Entity Recognition

1 Introduction

With the advance of information technology, the capabilities of massive data storage and data productivity have triggered an evolution in the usage of information, especially in the area of disaster management. Decision making in disaster management relies on the gathered information and the knowledge of decision makers. Due to the rapid growth of the amount of data, more data and information are being consumed now than in the past. This has led to the variance of data in terms of their format, source, and attributes, which has in turn increased the difficulties of the decision makers to accomplish their mission accurately and efficiently.

To assist the decision makers in efficiently and accurately utilizing, selecting, and processing information, we developed a dialogue system for disaster management. The dialogue system is based on text input by the decision makers using their devices, such as smartphones or tablets. It provides the data, information, and corpora for the topic of interest by treating the user's inputs as queries.

In our previous research, for providing information and data to the user, we developed a handcrafted keyword–information mapping table and used a fuzzy search algorithm. By searching keywords in the user's input, the system provides the most suitable information from the table to the user. During the implementation, we noticed that it was tedious to manage the keywords and information in the mapping table. In addition, a user experience test revealed that for the inputs that had more than one keyword, instead of providing integrated information, the system could only retrieve the part that mapped to one of the keywords. Therefore, we constructed a knowledge base to optimize the performance and maintainability of the system.

To build the knowledge base for disaster management, we designed a domain model to describe the knowledge base. The domain model is often used in software engineering and ontology engineering. We performed an abstraction on the knowledge of professional information providers and the required data on disaster management, while considering their source, certainty, and spatiotemporal features. A diagram composed of the models, the slots of the models, and the relations between the models was developed. The entities of the knowledge base are later used to recognize entities in the user's inputs. By entity recognition, the intent of the user and the proper response to the user's query can be determined.

2 Previous Research

We developed a dialogue system to provide data to the decision makers of the Water Resources Agency in Taiwan. The language of the dialogue system is Chinese, and the system is implemented as a chatter bot via LINE, a commonly used messaging application in Taiwan.

2.1 Algorithm

To determine the proper responses to a user's request, we developed a handcrafted keyword–information mapping table on the basis of experiments and the needs of the staff and decision makers of the agency. By searching listed keywords in the user's input, the system provides the most suitable information from the table to the user.

The matching of keywords from the user inputs and mapping table are sorted by the similarity of strings. Unlike in English, there is no space in a sentence to segment words in Chinese; therefore, we simply regarded each character as a single token, instead of using a single word as a token.

The similarity is calculated by comparing the number of same characters in the input text and each keyword in the mapping table, using Equation (1).

$$sim(d_j, q) = \frac{\sum_{i=1}^{t} f_{i,j}}{t} \quad (1)$$

- d_j is the j-th text from the corpus of the system.
- q is the user's input.
- t is the number of tokens in the user's input. The tokens are the characters of the text in this case.
- $f_{i,j}$ is the count of appearance of the i-th token in the user's input in the j-th text from the corpus of the system.

For the keyword(s) with a similarity score of 1.0, the system returns the information related to the keyword(s) to the user. If none of the keywords has a similarity score of 1.0, the system returns a list of keywords with a similarity score that is higher than 0.5 to the user so that the user can improve the inputs.

2.2 User Test

The system was tested by the staff of the Water Resources Agency. The users pointed out the following flaws in the system performance:

- Lack of reaction to stickers.
- Requirement of adjusting the views of data presented by tables. Tables are often too small when displayed on smart phones.
- Poor performance in case of long inputs.
- Poor performance in realizing natural language.

In addition, during the user test, in some given tasks for testing that required integrating different types of data, the time taken for accomplishing those tasks was higher than for easier tasks. Instead of retrieving all of the requested information simultaneously, the users had to split the input into fundamental queries to retrieve one part of the information at one time.

2.3 System Management and Maintenance

The keyword–information mapping table was directly handcrafted in the system codes. Hence, it was tedious and difficult to manage the keywords and information in the mapping table. Moreover, it resulted in a lack of flexibility in extending the mapping table, making the task of improving the system performance challenging.

3 Objective

The objective of this research is to develop a knowledge base for improving the ability of the dialogue system to perform the following:

1. Recognize the intent and provide the information requested through user inputs.
2. Respond with proper information depending on the user's request.
3. Provide a better solution for managing data.

To meet these objectives, we developed a knowledge base for disaster management, a system for processing the user input to retrieve the information sought, and a user-friendly console for the system.

4 Methodology

The methodology includes two parts: knowledge base and named entity recognition. The knowledge base stores and provides information, data, and knowledge. Named entity recognition completes the task of recognizing the intent and given information in the user's input.

4.1 Knowledge Base

4.1.1 Overview

A knowledge base is a manually compiled knowledge collection [1]. Knowledge is extracted into entities, relations, and attributes in a knowledge base. A knowledge base can be used to enable intelligent applications such as question answering [2]. The successful integration of a question-answering system prompted us to develop a knowledge base of the required information for management decision making in water-resource-related disasters.

Information extraction techniques are used in the construction and update of knowledge bases. Two types of information extraction methods exist: rule-based and statistical methods. Specifically, rule-based methods are more useful in closed domains where human involvement is both essential and available [3]; hence, we chose a rule-base method for information extraction in our study.

4.1.2 Domain Model

We performed an abstraction on information and data by discussing with the experts from the staff and information providers of the Water Resources Agency. By considering the source, certainty, and spatiotemporal features of the required data on disaster management, we designed a domain model as the model of the knowledge base. The domain model describes the structure of the knowledge base. Models of the domain model are the abstraction of knowledge, data, and information. The entity-relationship model (ER model) diagram is applied as a visualization tool for the system engineer and the disaster management experts to discuss and verify the domain model. The model, slot of attributes, and relation between entities can thus be defined.

The entity in the knowledge base is the instance of the model in the domain model. The attribute of entities is the slot of models filled with values. The relation between entities in the knowledge base is the relation in the domain model. The roles of the from-end and to-end models/entities are defined in association with the relation in the domain model.

4.1.3 Challenges

One of the main challenges in the construction is that the requested information during disaster management may be in the form of streaming data [4]. Therefore, the knowledge base should be capable of updating data in a timely manner. Data can be categorized into two types:

- Static data: data that seldom change, such as the name and location of facilities.
- Streaming data: temporally updating or increasing data, such as observations of precipitation and water levels, logs of transportation resources, meetings, and operations during disaster events.

In the implementation, streaming data are collected from the application programming interface (API) of the information system of the Water Resources Agency. When the data are updated, we archive them into the database of our system in the originally provided format, and extract them using the related model of the domain model into the knowledge base.

The other challenge is that the source of different data and information varies. The requested data of the Water Resources Agency may be provided by other government institutions such as the Central Weather Bureau. In our research, we focus on the data provided by the Water Resources Agency alone.

4.2 Named Entity Recognition

4.2.1 Overview

Named entity recognition (NER) is one of the tasks in information extraction and its purpose is to recognize the entities mentioned in documents. In our research, NER is applied to recognize entities in the user's input in order to query requested information from the knowledge base.

4.2.2 Segmentation and Tagging

The input text is segmented for retrieving a list of words. In addition, position-of-speech (POS) tagging is used to determine the type of each word. Nouns are regarded as candidate entities for later comparisons with the entities in the knowledge base.

In addition, words or phrases tagged as time are extracted and transformed into date/time objects in the system using handcrafted rules. For example, if "yesterday" appears in the input text, it will be transformed into the absolute date of the previous day.

4.2.3 Similarity Calculation

After retrieving a list of nouns, by searching the names of entities and models, the system can recognize the entities or mentioned models in the input and retrieve the requested information from the knowledge base. The type (e.g., location, person, and not-categorized) and content (i.e., the exact text of the word) of the nouns will be compared with the following:

- Names of entities in the knowledge base.
- Names of slots for attributes of the entities.
- Data types of attributes of the entities.
- Values of attributes of the entities.
- Names of models in the domain model.
- Names of slots of the models.
- Data types of slots of the models.
- Names of relations in the knowledge base and the domain model.
- Names of roles of two related entities or models.

The texts mentioned above compose a corpus of the system. The similarity of words in the user's input and the texts from the corpus are calculated using the cosine similarity by treating the texts as vectors in a vector space.

Each character plays the role of a unit vector on different dimensions of the vector space. The vector representing each word is composed of unit vectors, and the weights of unit vectors are determined by text model term frequency–inverse document frequency (tf–idf). The similarity of a word in the user's input and the text from the corpus of the system is calculated using Equations (2), (3), and (4).

$$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \tag{2}$$

$$w_{i,q} = (1 + \log f_{i,q}) \times \log \frac{N}{n_i} \tag{3}$$

$$w_{i,j} = (1 + \log f_{i,j}) \times \log \frac{N}{n_i} \tag{4}$$

- d_j is the j-th text from the corpus of the system.
- q is the user's input.
- t is the number of tokens in the user's input. The tokens are the characters of the text in this case.
- $w_{i,j}$ is the weight of the i-th token in the user's input in the j-th text from the corpus of the system.
- $w_{i,q}$ is the weight of the i-th token in the user's input in the input itself.
- $f_{i,j}$ is the count of appearance of the i-th token in the user's input in the j-th text from the corpus of the system.
- $f_{i,q}$ is the count of appearance of the i-th token in the user's input in the input itself.
- N is the number of the texts in the corpus of the system.
- n_i is the number that contains the i-th token of the user's input.

The nouns from the user's input will be regarded as the entities or models whose related text has a higher similarity score than the given threshold.

4.2.4 Intent Detection

If the name of any model in the domain model is recognized in the user's input, the model(s) will be treated as the intent of the user's request. The desired instances of the intent model(s) will be retrieved from the knowledge base by filtering by other recognized entities and date/time objects extracted from the user's request.

On the other hand, if only entities are recognized, the system will return the entities and links to related entities to the user for further usage.

5 Implementation

5.1 Domain Model

There are, in total, 43 models, 214 attributes, and 126 relations in the domain model. The relations can be categorized into 53 types.

Table 1. Information of models in the domain model. Type 1 refers to static data while type 2 refers to streaming data.

Model name	Type	# of slots
Institution	1	2
WRA Affiliation	1	3
Region	1	2
Village	1	3
Town	1	7
County	1	3
Reservoir	1	13
Storage	1	4
Rain Station	1	4
Water Level Station	1	6
Response Event	1	4
Operation Log	1	2
Operation Standard	1	1
Flood Operation Condition	1	4
Typhoon Operation Condition	1	3
Drought Operation Condition	1	3
River Basin	1	2
River	1	2
Rain Warning	1	4
Water Level Warning	1	4
Reservoir Warning	1	8
Flood Disaster	2	5
Typhoon Disaster	2	12
Drought Disaster	2	4
Provider	1	6
Source	1	2
Equipment Kind	1	2
Equipment Model	1	3
Equipment Log	2	5
Time Interval	1	3
Rain Observation	2	6
Meeting Kind	1	3
Meeting Log	2	3
Meeting Reference	2	3
Pump Model	1	2
Pump	1	2
Pump Log	2	8
Reservoir Observation	2	17
Flood Situation Log	2	14
Facility Situation Log	2	14
Warning Kind	1	2
Warning Log	2	4
Water Level Observation	2	6

Table 2. Information of relations in the domain model.

Relation name	From role	To role	Count
supervised by	supervisee	supervisor	2
as	main	alias	1
follow	institution	standard	2
publish	institution	warning	2
supervise	supervisor	supervisee	2
supervise	institution	district	1
own	owner	object	2
supervise	institution	river basin	2
have	institution	log	2
as	alias	main	1
locate at	district	district	3
warned by	district	warning	3
contain	district	district	3
contain	district	facility	4
warned by	district	facility	1
have	district	log	3
supervised by	district	institution	1
watch	facility	warning	3
observe	facility	observation	3
locate at	facility	district	4
have	facility	log	1
owned by	object	institution	1
locate at	facility	river basin	3
have	instance	log	3
occur at	log	instance	1
originate from	log	standard	1
result in	standard	log	1
consider	standard	condition	3
follow by	standard	institution	1
considered by	condition	standard	3
contain	river basin	facility	3
contain	river basin	rivers	1
supervised by	river basin	institution	2
have	upstream	downstream	1
locate at	river	river basin	1
connect	downstream	upstream	1
watched by	warning	facility	3
warn	warning	district	3
provide	provider	resource	1
publish	source	data	8
provided by	resource	provider	1
categorize	type	instance	7
categorized by	instance	type	7
published by	data	source	9
occur at	log	storage	1
observed at	observation	facility	3
attached with	log	reference	1
occur at	log	event	2
attach to	reference	log	1
owned by	object	owner	1
occur at	log	district	3
of	log	institution	2
published by	kind	institution	1

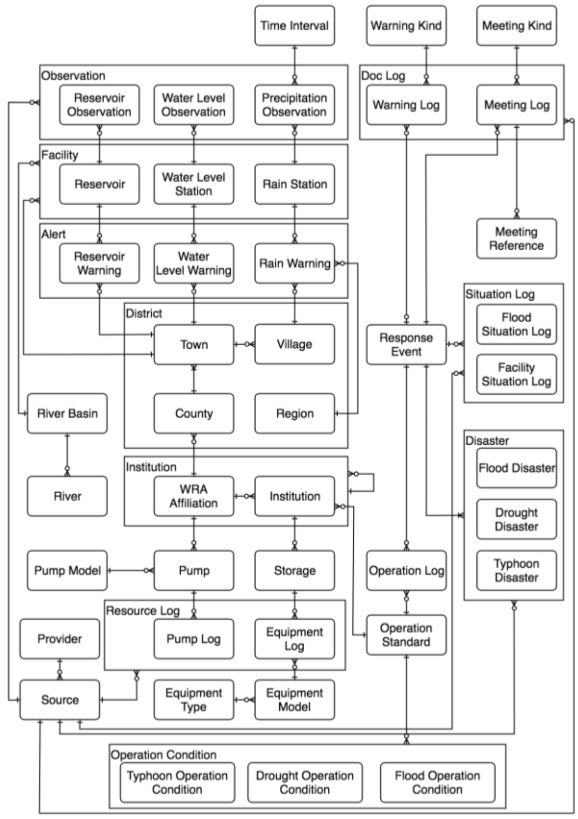

Figure 1. ER model diagram of the domain model.

5.2 System

Python is used in the development. Each part of the system is constructed using the various Python libraries or modules:

1. Jieba [5]: Chinese text segmentation module, used for the segmentation and POS tagging of the user's input. The algorithm of Jieba is based on a prefix dictionary structure to achieve efficient word graph scanning. The dictionary of Jieba can be extended.
2. Django [6]: The web framework consists of an object-relational mapper (ORM), which mediates between models of data and relational databases, used for implementing the domain model and the knowledge base. The domain model is implemented by the ORM, and the data of entities are stored in the database. By mediations using the ORM, the data can be transformed back to entities. It also provides an automatically generated graphical user interface (GUI) console for managing the data in the system.

In development, SQLite3 is chosen as the database for testing and validation.

First, static data and information are added directly to the knowledge base using the domain model. Second, the API listener of the system checks up the APIs of streaming data. If an update is detected, the system archives the newest data in original formats, transforms the data into entities using the models, and adds them to the knowledge base.

After the construction of the knowledge base, the Jieba dictionary is extended by adding texts from the corpus of the system, which is mentioned in 4.2.3. To increase the capability of referring the different names of the same entities, models, slots, and relations, aliases are attached to them, which are also added to the Jieba dictionary.

The transformation of words or phrases tagged as time is done by rules written in regular expressions and compiled in Python. They are transformed into native Python date/time objects and can be used as the attributes of entities or the slot values of models directly.

The data can be managed on the automatically generated console. Adding, modifying, and deleting data on the new system is much easier than on the original one.

The entire system runs as a web application. It receives the messages from LINE, following which the proper information is processed and returned to LINE.

6 Results and Discussion

In the current progress, we focus on recognizing and retrieving requested information by direct relation to other mentioned entities, such as facilities with their locations, in the user's input. By adjusting the threshold of similarity to 0.9 and adding handcrafted rules to improve the accuracy of recognition, the system can currently retrieve requested information from the knowledge base.

For example, when the system receives the text, "Tell me the Rain stations in Taipei City (告訴我臺北市雨量站)," by entity and model recognition, the system determines that the intent is to retrieve rain stations in Taipei City; therefore, it shows the list of rain stations in Taipei City. On the other hand, when the system receives the text, "Rain stations in the Da-an District of Taipei City (臺北市大安區雨量站)," it recognizes Taipei City and Da-an District as a county-level district and a town-level district, respectively, and then determines that the intent is to retrieve rain stations situated in only the Da-an District of Taipei City instead of the other district in Taichung City, which is also called Da-an; therefore, it shows the list of rain stations in the Da-an District of Taipei City.

However, the system can still not retrieve information that is not directly related to other entities provided in the user's input. In addition, setting the threshold of similarity to 0.9 might lead to missing relevant information. Setting the threshold too low, however, may reduce the retrieval speed, lower the accuracy of recognition, and cost more time and computation resources.

7 Conclusions

In this research, to improve the performance of the dialogue system for disaster management, we developed a knowledge base and designed a recognition process; we implemented the system to recognize the intent and provide the information mentioned in the user's input by replying with accurate information depending on the user's query.

By designing a domain model of the knowledge base through discussions with experts, we developed a system to provide and manage the information requested in disaster management. The knowledge base and the domain model increase the capability of the system in recognizing the user's intent and query.

In future work, to improve the accuracy of intent and entity recognition, the similarity between the keywords and the user's input by considering the slot and relation structure of models will be discussed. In addition, the feature of retrieving the requested data and information that do not directly relate to the mentioned entities in the user's input will be developed.

Acknowledgements

We thank Assistant Professor Yun-Cheng Tsai of the Center for General Education, National Taiwan University, for her useful suggestions.

This research was supported by Taiwan's Ministry of Science and Technology (MOST) under contract 107-2119-M-002-017.

References

[1] Suchanek F. and Weikum G. Knowledge harvesting from text and Web sources. In *2013 IEEE 29st International Conference on Data Engineering (ICDE)*, pages 1250–1253, Brisbane, Australia, 2013.

[2] Frank A. et al. Question answering from structured knowledge sources. *Journal of Applied Logic*, 5(8):20–48, 2007.

[3] Sarawagi S. *Information Extraction*. Now Publishers Inc., PO Box 179 2600 AD Delft, 2008.

[4] Hristidis V. et al. Survey of data management and analysis in disaster situations. *The Journal of Systems and Software*, 83:1701–1714, 2010.

[5] Django: The Web framework for perfectionists with deadlines. On-line: https://www.djangoproject.com, Accessed: 31/12/2017.

[6] fxsjy/jieba. On-line: https://github.com/fxsjy/jieba, Accessed: 31/12/2017.