Acquisition of Domain-specific Patterns for Single Document Summarization and Information Extraction

Mian Du & Roman Yangarber
University of Helsinki
Department of Computer Science, P.O. 68, FI-00014, Helsinki, Finland
du@cs.helsinki.fi & roman.yangarber@cs.helsinki.fi

ABSTRACT

Single-document summarization aims to reduce the size of a text document while preserving the most important information. Much work has been done on open-domain summarization. This paper presents an automatic way to mine domain-specific patterns from text documents. With a small amount of effort required for manual selection, these patterns can be used for domain-specific scenario-based document summarization and information extraction. Our evaluation shows that scenario-based document summarization can both filter irrelevant documents and create summaries for relevant documents within the specified domain.

KEYWORDS

Document Summarization, Information Retrieval, Information Extraction, Data Mining, Pattern Acquisition, Natural Language Processing

1 INTRODUCTION

Much work has been done for accessing relevant articles from a specific domain[1]. Information retrieval (IR) systems apply keyword-matching to filter out irrelevant documents and return the possibly relevant ones containing the keyword combinations. One can try to read the text of these potentially relevant documents to acquire relevant information, which requires additional time on the part of the user. Document summarization is often used to reduce the amount of text the user needs to scan and to reveal the most important information in the document.

[1]Domain refers to a broad subject area, such as medicine, security, business, etc.

However, for certain purposes, these methods may not satisfy the need of information users. For example, for monitoring disease outbreaks from news, the keywords could be disease names and their variations. With keyword matching alone the following two types of incorrect results could be returned:

- errors of commission: a document containing a disease name may describe something other than an outbreak event, such as historical overview, or scientific studies of the disease;

- errors of omission: adding more keywords like ”outbreak”, ”epidemic” or ”pandemic” to increase precision, on the other hand, would eliminate many relevant documents. Since the number of different ways to describe an outbreak in plain text is almost unlimited, relevant documents do not need to contain these specific keywords.

In addition, the goal of general-purpose document summarization is to select the most important passages from the document. Even if the document is irrelevant, it may still produce a summary. Therefore, using IR and general-purpose document summarization, may still suffer from irrelevant summaries and missing relevant summaries.

In this paper we propose an automatic way to extract domain-specific patterns from text documents. With minimal manual post-selection, we can use these patterns to generate domain-

specific scenario-based[2] single-document summaries. The method will generate summaries only for relevant documents and will filter out irrelevant documents. For the disease-outbreak scenario stated above, relevant documents must describe an actual epidemic. We generate the summary using sentences containing the patterns. Irrelevant documents do not describe a disease outbreak; they should not contain any relevant patterns and hence should not generate a summary. In this paper we present experiments for two domains—medical epidemics and business intelligence—in order to demonstrate the method. In the medical domain, we focus on patterns to describe the *disease outbreak* scenario. In the business domain, we extract patterns for major business activity scenarios, such as "investment", "new product launch", "management/leadership change", etc.

Beyond information retrieval and summarization, information extraction (IE) is used to automatically extract pre-specified kinds of facts from natural-language text, [1]. Finding events related to disease outbreaks or to business activities in news articles are typical use cases for IE. Many IE systems are pattern-based, where a core task in building the IE system is finding extraction patterns. The domain-specific patterns acquired by our method can be integrated into an IE system to extract more detailed information in for new domain.[3]

The paper is organized as follows: Section 2 introduces related work. In Section 3 we describe the methodology for finding the domain-specific patterns. In Section 4, we present experimental results of using these patterns to generate scenario-based document summaries in the medical and business domains. We conclude with a discussion of the results and plans for future work in Section 5.

[2]Scenario refers to a collection of certain types of events or actions typical within a domain, such as disease outbreaks in the medical domain, investment activities in business domain, etc.

[3]The evaluation of performance in IE is beyond the scope of the present paper; but the potential for this avenue of research should become clear.

2 RELATED WORK

2.1 Document Summarization

General, open-domain document summarization aims for the following characteristics [2]:

- summary can be produced from single or multiple documents;
- summary should convey the most important information;
- summary should be short.

There are two main approaches to summarization: extraction and abstraction. Extraction identifies and keeps the most important sections of the documents; abstraction uses natural-language generation techniques to create the summary using new text, not necessarily explicit in the document(s). We explore *domain-specific* scenario-based summarization which focuses on single-document summarization using extraction. Thus, it has one more characteristic—summary should be relevant to *pre-defined scenarios* in the domain. For example, in the business domain, if we pre-define the target scenarios in the domain as "investment" and "new product launch" by a company, then a document describing either of these scenarios will generate summary with the scenario label, while a document describing something else about a company (e.g., the company's structure) will generate no summary.

During the last half century, much work has been done for single-document summarization using extraction. Early work focuses on word or phrase frequency, [3], position of the phrase, [4], and key phrases, [5]. Later researchers began to apply machine-learning methods to produce document extracts, such as Naive-Bayes methods [6], decision trees [7], Hidden Markov Models, [8] and Neural Networks, [9]. Recently, deep natural-language analysis methods have been used. These methods are more closely related to this paper. For example, [10] use strong lexical chains[4] to find sentences suit-

[4]A lexical chain is a sequence of linked words in the text.

able for extraction. These methods are used for general-purpose summarization, while this paper uses domain knowledge to mine domain-specific patterns for summarization.

2.2 Information Extraction

Information extraction (IE) was introduced in the 1970's for extracting specific factual information from natural-language text, e.g., from newspaper articles. IE is used to apply a sequence of steps of formal linguistic analysis to obtain the syntactic and semantic structure of the text, to extract only the specified kinds of information from the structured text, and finally to store the information into a database for later querying. A document that does not contain the specified kind of information is considered irrelevant and is discarded.

The resulting output consists of required items as slot values of a structured template (see Table 1). Based on the linguistic analysis produced by Natural Language Processing (NLP) parsers from unstructured text, extraction patterns are used to match facts. These facts are then used to fill the slots of the resulting template. An extraction pattern contains a place-holder for specific tokens and their surrounding context. The surrounding context may be fixed, and the token may be the variable. For instance, *X* **was/were infected by** *Y* **on** *Z* is a sample pattern. It could be used for matching outbreak event from a sentence like "18 people were infected by H1N1 on Friday." The fixed surrounding context in this example is **... were infected by ... on ...**;the variable tokens are *X*, *Y* and *Z*. According to the definition of required slots by the IE system, *X* here could be any noun group belonging to the semantic class *human*, (e.g., *18 people*, *a 38-year-old Brazilian woman*, etc.), *Y* can be a name of an infectious disease, and *Z* may be any representation of date group (e.g. *Friday*, *24th of May*), etc.

An IE system usually has a large number of such extraction patterns to match the required facts. Different systems, depending on their purpose and domains, will have different patterns. Finding extraction patterns is therefore considered to be a core task in building IE systems, since the quality of the resulting template largely depends on the quality of extraction patterns. In general, there are two ways of obtaining suitable patterns: the knowledge-engineering approach and the machine-learning approach. In the knowledge-engineering approach, the extraction patterns are defined by computational linguists with the help of knowledge experts in the required domain, [11, 12, 13, 14]. Machine-learning approaches try to automatically identify essential regularities for information extraction from a training document collection. Examples of IE systems using this approach include *AutoSlog-TS*, [15], *CRYSTAL*, [16], *PALKA*, [17] and *PULS*, [18, 19]. These approaches can be combined in a hybrid fashion, depending on the nature of the task and the amount of noise in the unstructured data. Our method is related to the machine-learning approach.

Disease:	Cholera
Country:	China
Time:	01.02.2013
Total:	5
Victims:	people
Status:	dead

Table 1. Template produced by IE process

3 PATTERN ACQUISITION

In this paper, we focus on experiments for two domains: medical epidemics and business intelligence. We describe the application of our method to extract frequent patterns in these target domains. The acquisition of patterns is viewed as a domain-specific task; our experiments for the medical and business domains are done separately. After quick manual post-selection, we use these patterns to directly generate summaries, or induct them into the IE system. Our method can be viewed as consisting of three steps: data collection and NLP pre-processing; pattern mining; scenario-based selection.

3.1 Data Collection and NLP Pre-processing

We do not use annotated text for training for a new domain, rather we use potentially relevant texts for acquisition of domain-specific patterns.

In the medical domain, we use IR to collect news articles from Web-based news sources, which contain at least one item from a list of relevant keywords. The keywords include disease and symptom names (e.g., "H1N1", "Cholera"), words related to disease outbreaks (e.g., "patient", "hospital", "outbreak") and words describing patient status (e.g., "dead", "sick"). Some of the retrieved articles may contain disease outbreak events, for example, "17 people were affected by H1N1 outbreak that began this week in the US." Most articles do not contain a disease outbreak. We have collected over a million potentially relevant articles between 2013.01.01 to 2013.05.31. Some example sentences from these articles are shown below:

- M1: China reports another bird flu death, total now 8.
- M2: The H7N9 strain has infected 24 people, all of them in eastern China, of whom eight have died.
- M3: This could be an important step in identifying additional causes for obesity in humans, especially considering dramatic increases in childhood obesity in the United States.

In the business domain, we use our IR system to get business-related news from about 1,000 providers of business news, such as BBC News Business, New York Times Business Day, Yahoo!News Business, etc.. Between 2013.01.01 to 2013.05.31, we have collected 216,565 news articles. The following are some example sentences from these articles:

- B1: The project involves a total investment of CNY 650mn (EUR 78.81mn).
- B2: Spanish bank, Banco Etcheverria, has approved a capital increase of EUR 499,659.12 (US$ 660,886.19).
- B3: MPS sells stake in Biverbanca to CRAsti for €208.96mn.

For each domain, we randomly select 10,000 of these possibly relevant articles for pattern acquisition.

In general, natural-language text data is difficult to handle since writers can express a piece of information in numerous different ways. In order to make patterns more clear and reduce sparsity, we perform NLP pre-processing. For each domain, sentences of the selected 10,000 possibly relevant articles are split into words; punctuation is removed. In standard data-mining terms, each word in the sentence is treated as an item and each sentence is treated as one sequential transaction of items. [10]

First, we need to define the key item in the domain. In our experiments, infectious disease is the key item in the medical domain and company is the key item in the business domain. Sentences which do not contain the domain-specific key item are considered to be irrelevant and are not used. We use dictionaries of infectious disease names and company names (including their synonyms and acronyms) to determine whether the sentence contains a key item. Disease names and company names are extracted from text using the named-entity recognition module in our IE system.[20, 21, 22, 23]

Second, we convert domain-specific key items and other general categorical items into their types and use their types as items. In this study, such items include:

- Infectious disease names are converted to "c-disease-name".
- Country names are converted to "c-country-name".
- Company names are converted to "c-company-name".
- Items describing a human, such as "people", "patient", "man", etc., are converted into "c-human".
- Years are converted to "c-year".

- Numbers (e.g., "104", "1 280", "2,367" "five") are converted to "c-number".

- Currencies (e.g., "RMB", "€") are converted to "c-currency".

Third, we remove stop words, such as "an", "and", "he", "that", etc., from the sentences and keep only content words as items.

After this pre-processing, each transaction $T =< W_1, W_2, W_3, ..., W_n >$ describes a sentence mentioning at least one disease or company. Each item W in T represents either a content word in the sentence, or a type of categorical item.

Medical transactions from example sentences are shown below. Example M3 is not used since "obesity" is not an infectious disease. In total, 10,000 articles contain 14,816 such transactions.

- TM1: **c-country-name** reports **c-disease-name** death total now **c-number**

- TM2: **c-disease-name** has infected **c-number c-human** in **c-country-name c-number** have died

Accordingly, business transactions contain at least one company name; example B1 is removed. The 10,000 articles generate 35,024 transactions.

- TB2: **c-country-name** bank **c-company-name** has approved capital increase of **c-currency c-number c-currency c-number**

- TB3: **c-company-name** sells stake in **c-company-name** to **c-company-name** for **c-currency c-number**

3.2 Pattern Mining

Items (including types of categorical items) in the transactions are considered in sequential order. We try to find frequent sequential patterns, P, to describe an outbreak event from these transactions, such as,

- P1: (c-disease-name, infect, c-number, c-human)

- P2: (c-country-name, report, c-disease-name)

These patterns only find adjacent sequential items or types. This means if a transaction is formed by three sequential items W_1, W_2 and W_3, then the only allowed sequential patterns are (W_1, W_2) or (W_2, W_3) or (W_1, W_2, W_3); pattern (W_1, W_3) is not allowed. Patterns containing a single item (e.g., (W_1)) are not used.

The support S_p of each pattern P is calculated as follows. The count $|P|$ is increased by one when P is found in a transaction T. If P is found twice in the same transaction T, the count increases only by one.

$$S_p = \frac{|P|}{|T|} \tag{1}$$

where $|T|$ is the total number of transactions. If we set the minimum support S_{min} to be 0.01, the pattern needs to appear at least in 1% of the transactions to be picked up as a frequent sequential pattern.

We use an Apriori-like algorithm [24] to mine frequent sequential patterns. The pipeline of this algorithm is described below.

- Initialize: read in the input and generate the initial counts for all two-item patterns, (W_1, W_2).

- Iteration: starts with two-item patterns; stops when there is no possible next round patterns. Inside the loop, we get the counts for all possible frequent patterns generated by two frequent patterns in the previous round; and we use these counts to generate next round's possible frequent patterns. For example,

  - we have frequent patterns (W_1, W_2), (W_2, W_3) and (W_4, W_5) initially;
  - we generate next round's possible candidates (W_1, W_2, W_3) from (W_1, W_2) and (W_2, W_3) since the k-1 suffix of (W_1, W_2) and k-1 prefix of (W_2, W_3) are the same (where k refers to the number of items in the patterns);

– if the suffix of one pattern is not the same as the prefix of another pattern, we do not generate any candidate from these two patterns, such as (W_2, W_3) and (W_4, W_5).

- Output all patterns where $S_p \geq S_{min}$.

The number of acquired patterns increases as the S_{min} decreases. Table 2 shows some statistics for different values of S_{min}. Since we start with two-item patterns at round 1, when number of rounds is 3, the mined patterns contain at most 4 items. In the table, the *key* patterns are those that contain at least one key item (c-disease-name or c-company-name depending on the domain); they are generally more relevant.

S_{min}	# of rounds		# of patterns		# of key patterns	
	Med	Bus	Med	Bus	Med	Bus
0.01	3	3	209	91	51	68
0.005	6	5	473	213	102	139
0.001	8	6	3159	2232	506	1280

Table 2. Acquired patterns for different S_{min}

3.3 Scenario-based Selection

After acquiring frequent patterns, we manually select patterns which could be used to describe our target scenarios. In the medical domain, the scenario is infectious disease outbreak. Through quick manual selection, we selected 236 patterns, which describe an outbreak event, from among the 506 key patterns. A sample of the selected patterns are shown in Table 3.

In the business domain, the scenarios correspond to major business activities. We do not know in advance what types of activities are important, nor what keywords reporters use to write the news articles. From the 1280 key patterns which contain at least one *c-company-name*, we have manually selected 259 patterns, which describe a business activity of a company. When selecting patterns, we also group them by assigning a scenario label to each pattern. Table 4 shows some examples of selected patterns, with their support and scenario label. In total, we have identified 12 frequent types of activities as scenarios in the business domain.

Pattern	$S_p(\%)$
(infected with c-disease-name)	1.86
(strain c-disease-name has killed)	0.58
(new c-disease-name cases)	0.65
(c-disease-name epidemic in)	0.52
(c-number c-human died c-disease-name)	0.27
(c-number c-disease-name cases were reported)	0.27
(cases c-disease-name in c-country-name)	0.26
(c-number c-disease-name cases in)	0.22
(c-number cases c-disease-name reported)	0.12
(c-disease-name outbreak in c-country-name)	0.11

Table 3. Examples of manually selected patterns for the infectious disease outbreak scenario

Pattern	$S_p(\%)$	Scenario
(c-company-name recall)	0.21	Product Recall
(c-company-name advertising)	0.20	Marketing
(c-company-name investments)	0.19	Investment
(c-company-name purchase c-company-name)	0.13	Acquisition
(c-country-name c-company-name plans)	0.23	Planing
(c-country-name c-company-name unveils)	0.20	New Product
(c-country-name c-company-name opens)	0.19	Open
(c-company-name contract is)	0.18	Contract
(c-country-name c-company-name launch)	0.14	New Product
(c-company-name has launched c-company-name)	0.13	New Product
(c-country-name c-company-name acquires)	0.12	Acquisition
(c-country-name c-company-name appoints)	0.12	Management Succession
(c-company-name deal is)	0.12	Contract
(c-country-name c-company-name approves)	0.11	Announcement
(c-country-name c-company-name buys)	0.11	Acquisition
(c-country-name c-company-name supply)	0.11	Contract
(c-country-name c-company-name gets)	0.10	Contract
(c-company-name is owned by c-company-name)	0.10	Ownership
(c-company-name has been awarded c-currency)	0.11	Investment

Table 4. Examples of manually selected patterns for scenarios in the business domain

4 SCENARIO-BASED DOCUMENT SUMMARIZATION

We use the manually selected patterns to generate domain-specific scenario-based single-document summaries, in two steps. First, we use the same NLP pre-processing module as described in Section 3.1 to convert sentences of a document into transactions. Then, we select sentences which match any of the domain-specific patterns as summary sentences. A document containing no such sentence is regarded as irrelevant for the defined scenario in the domain.

For the scenario of disease outbreaks in the medical domain, sentences containing any pattern of disease outbreaks are returned as the summary for a document. For example, Figure 1 demonstrates one example of a summary of a document.

Similarly, we use the 259 manually selected patterns to generate summaries in the business domain. In addition to the summary, we also automatically label a document with the scenario labels of the patterns; a document may be assigned more than one scenario label.

To evaluate the performance of the summarization, we randomly select 10,000 documents from our corpus, described in Section 3.1 for each domain. The evaluation corpus does not overlap with the documents used for pattern acquisition, as also described in Section 3.1. We use our method to generate summaries for each domain. Results of the evaluation are shown in Table 5.

Domain	$\|Doc\|$	$\|Sum\|$	$Avg\|Doc_s\|$	$Avg\|Sum_s\|$
Medical	10,000	523	17.3212	4.21
Business	10,000	3,120	23.1336	4.78
$\|Doc\|$: number of documents $\|Sum\|$: number of documents which generate a summary $Avg\|Doc_s\|$: average number of document sentences $Avg\|Sum_s\|$: average number of summary sentences				

Table 5. Statistic results of summary evaluation

To perform a manual evaluation, for each domain we randomly select 20 documents which generate a summary and 20 documents which generate no summary. An expert from each domain is invited to manually pick sentences from these 40 documents (T) to generate manual summaries. Domain experts do not know what patterns we use, but they understand the scenarios. The expert from medical domain is expected to select only outbreak-related sentences from a document, while the expert from business domain knows 12 scenarios we are using and only selects sentences containing these 12 scenarios from a document to generate the summary. The evaluation results using accuracy, precision, recall and F1 score are shown in Table 6. These measures are calculated using the following formulae:

- D_o: documents which generate a summary by our method
- D_e: documents which generate a summary by expert
- $S_{o(d)}$: summary sentences by our method in document d, where $d \in T$
- $S_{e(d)}$: summary sentences by expert in document d, where $d \in T$
- $S_{(d)}$: all sentences in document d, where $d \in T$
- $|x|$: number of x
- $\neg x$: not x; e.g. $\neg D_o$ means documents which do not generate any summary using our method

$$A_D = \frac{|D_o \cap D_e| + |\neg D_o \cap \neg D_e|}{|T|} \quad (2)$$

$$P_D = \frac{|D_o \cap D_e|}{|D_o|} \quad (3)$$

$$R_D = \frac{|D_o \cap D_e|}{|D_e|} \quad (4)$$

$$F1_D = \frac{2P_D R_D}{P_D + R_D} \quad (5)$$

$$A_S = \frac{\sum_{d \in T} |S_{o(d)} \cap S_{e(d)}| + \sum_{d \in T} |\neg S_{o(d)} \cap \neg S_{e(d)}|}{\sum_{d \in T} |S_{(d)}|} \quad (6)$$

$$P_S = \frac{\sum_{d \in T} |S_{o(d)} \cap S_{e(d)}|}{\sum_{d \in T} |S_{o(d)}|} \quad (7)$$

$$R_S = \frac{\sum_{d \in T} |S_{o(d)} \cap S_{e(d)}|}{\sum_{d \in T} |S_{e(d)}|} \quad (8)$$

$$F1_S = \frac{2P_S R_S}{P_S + R_S} \quad (9)$$

The precision of both document-level and sentence-level summarization are very high in the

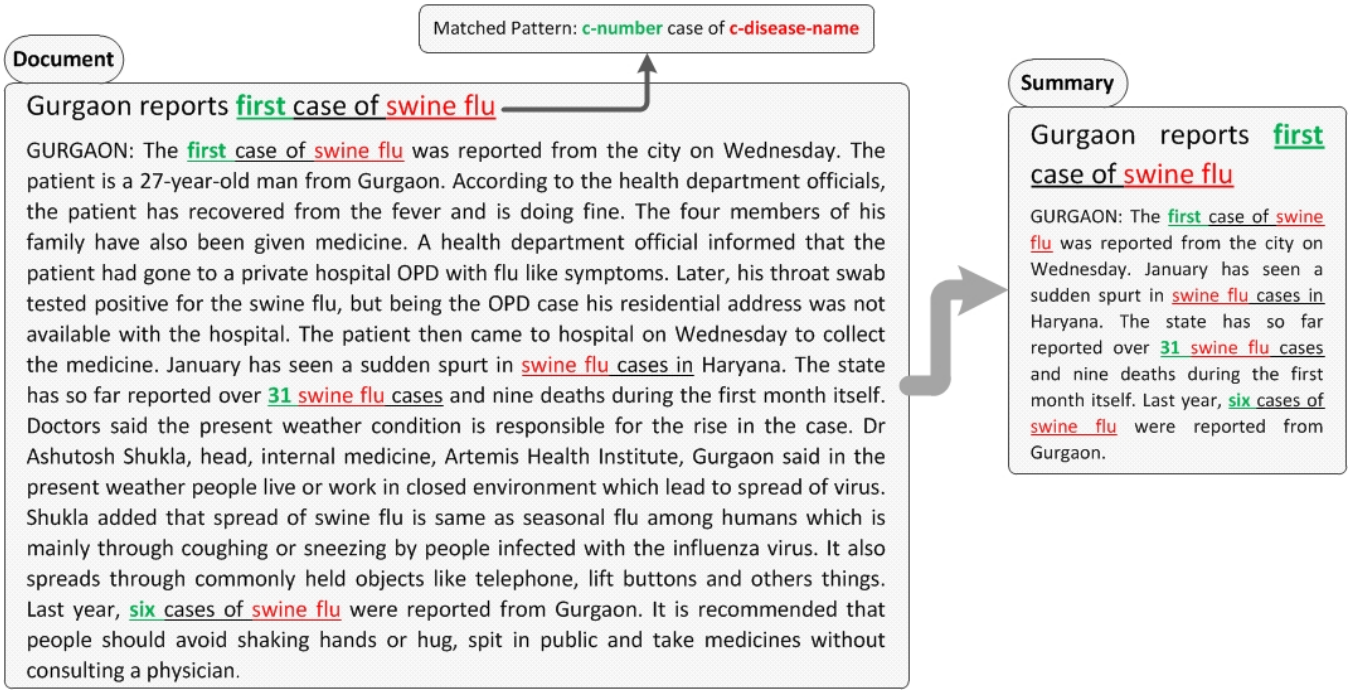

Figure 1. Example of summary for disease outbreak scenario in medical domain

Domain	Document level				Sentence level			
	A_D	P_D	R_D	$F1_D$	A_S	P_S	R_S	$F1_S$
Medical	97.50	100.0	95.24	97.56	78.90	82.77	53.21	64.78
Business	72.50	100.0	72.00	83.72	45.85	83.06	38.33	52.45

Table 6. Manual evaluation of summary

two domains. This demonstrates that our patterns are very reliable for scenario-based summarization in a specific domain. When comparing the differences between summaries generated using our method and ones generated by an expert, we have found that sometimes the document describes exactly the same information in two sentences in slightly different ways, such as the title and the first sentence of the document. Our method selects both sentences because they both match the patterns, while the expert chooses one of them to generate the summary. This decreases the precision of the method.

Document-level recall is much better than the sentence-level recall. This means that a relevant document will most likely describe the scenario-based information using some frequent patterns at least in one sentence. Summarization in the medical domain achieves better recall. This might be due to two reasons. First, we have only pre-defined one scenario in the medical domain, i.e., infectious disease outbreak, while there are 12 scenarios in business domain. The number of mined patterns per scenario is much higher in the medical domain. Second, the dictionary for infectious diseases is more stable and complete. In the business domain, many company names and their acronyms are not in our dictionary.

5 CONCLUSIONS AND FUTURE WORK

This paper demonstrates that a combination of NLP techniques and frequent sequential pattern mining algorithm can be used for mining frequent patterns in a specific domain from unstructured natural-language text, i.e., news articles. With a minimum manual selection effort, we use these patterns to generate domain-specific scenario-based document summaries. We have applied the method in two domains. The evaluation results show that scenario-based summarization can serve to filter out irrelevant documents and also extract important sentences from relevant documents as summaries for pre-defined scenarios in a specific domain. For document level information retrieval, this method achieves

very high precision while keeping quite high recall in both domains in our study. This demonstrates that this method may solve the problems for scenario-based information retrieval in a specific domain. We are continuously generating summaries from documents manually. These documents and summaries will be used in our evaluation in future work to make the evaluation more reliable.

In the future, we plan to improve our NLP pre-processing. For this study, we use seven item parsers to handle categorical items, including disease name, company name, country name, human, year, number and currency. From the results, we have found that the dictionaries for diseases and companies are not complete, especially the one for companies. Integrating a named entity (NE) parser in pre-processing stage for handling unknown names should improve the quality of the mined patterns. In addition, we plan to handle more categorical items, e.g., location, date, organization, etc. A general NLP parser could also be used to increase the chance of finding relevant patterns. The parser will tag words with their part-of-speech, analyze the unstructured text into phrases (e.g., noun phrase, verb phrase, etc.), and lemmatize words into their base forms (e.g., normalizing tense, number, etc.) By applying a general parser, we should be able to generate cleaner and more informative transactions for mining patterns.

We are also working on integrating the mined patterns into our IE system for extracting attributes of pre-defined scenario events in the domain, such as disease name, country, etc., as shown in Table 1. Some of these patterns already match at least three categorical items. These categorical items can be directly converted into attributes in an IE output. For example, pattern "c-number c-disease-name cases in c-country-name" can generate an IE event of disease outbreak with three attributes.

REFERENCES

[1] R. Gaizauskas and Y. Wilks, "Information Extraction: Beyond Document Retrieval," *Computational Linguistics and Chinese Language Processing*, vol. 3, no. 2, pp. 17–60, 1998.

[2] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," *Comput. Linguist.*, vol. 28, no. 4, pp. 399–408, Dec. 2002. [Online]. Available: http://dx.doi.org/10.1162/089120102762671927

[3] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Dev.*, vol. 2, no. 2, pp. 159–165, Apr. 1958. [Online]. Available: http://dx.doi.org/10.1147/rd.22.0159

[4] P. B. Baxendale, "Machine-made index for technical literature: An experiment," *IBM J. Res. Dev.*, vol. 2, no. 4, pp. 354–361, Oct. 1958. [Online]. Available: http://dx.doi.org/10.1147/rd.24.0354

[5] H. P. Edmundson, "New methods in automatic extracting," *J. ACM*, vol. 16, no. 2, pp. 264–285, Apr. 1969. [Online]. Available: http://doi.acm.org/10.1145/321510.321519

[6] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '95. New York, NY, USA: ACM, 1995, pp. 68–73. [Online]. Available: http://doi.acm.org/10.1145/215206.215333

[7] C.-Y. Lin, "Training a selection function for extraction," in *Proceedings of the Eighth International Conference on Information and Knowledge Management*, ser. CIKM '99. New York, NY, USA: ACM, 1999, pp. 55–62. [Online]. Available: http://doi.acm.org/10.1145/319950.319957

[8] J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '01. New York, NY, USA: ACM, 2001, pp. 406–407. [Online]. Available: http://doi.acm.org/10.1145/383952.384042

[9] K. M. Svore, L. Vanderwende, and C. J. Burges, "Enhancing single-document summarization by combining ranknet and third-party sources." in *EMNLP-CoNLL*. Citeseer, 2007, pp. 448–457.

[10] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," *Advances in automatic text summarization*, pp. 111–121, 1999.

[11] D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks, "Named entity recognition from diverse text types," in *Recent Advances in Natural Language Processing 2001 Conference*, Bulgaria, 2001.

[12] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "Gate: A framework and graphical development environment for robust NLP tools and applications," in *In Proceedings of the 40th Anniversary*

Meeting of the Association for Computational Linguistics, 2002.

[13] T. Jayram, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu, "Avatar information extraction system," *IEEE Data Engineering Bulletin*, vol. 29, pp. 40–48, 2006.

[14] W. Shen, A. Doan, J. Naughton, F., and R. Ramakrishnan, "Declarative information extraction using datalog with embedded extraction predicates," *VLDB*, pp. 1033–1044, 2007.

[15] S. Patwardhan and E. Riloff, "Learning Domain-Specific Information Extraction Patterns from the Web," in *ACL 2006 Workshop on Information Extraction Beyond the Document*, 2006.

[16] S. Soderland, D. Fisher, J. Aseltine, and W. G. Lehnert, "CRYSTAL: Inducing a Conceptual Dictionary," *CoRR*, 1995.

[17] J. Kim and D. Moldovan, "Acquisition of Linguistic Patterns for Knowledge-Based Information Extraction," *IEEE Transactiops on Knowledge and Data Engineering*, vol. 7, no. 5, pp. 713–724, 1995.

[18] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen, "Automatic acquisition of domain knowledge for information extraction," in *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, ser. COLING '00. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 940–946. [Online]. Available: http://dx.doi.org/10.3115/992730.992782

[19] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen, "Unsupervised discovery of scenario-level patterns for information extraction," in *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ser. ANLC '00. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 282–289. [Online]. Available: http://dx.doi.org/10.3115/974147.974186

[20] R. Grishman, S. Huttunen, and R. Yangarber, "Real-Time Event Extraction for Infectious Disease Outbreaks," in *In Proceedings of the 3rd Annual Human Language Technology Conference HLT-2002*, San Diego, CA, 2002.

[21] S. Huttunen, A. Vihavainen, M. Du, and R. Yangarber, "Predicting the relevance of event extraction for the end user," *Multi-source, Multilingual Information Extraction and Summarization*, pp. 163–176, 2013.

[22] M. Du, J. Kangasharju, O. Karkulahti, L. Pivovarova, and R. Yangarber, "Combined analysis of news and twitter messages," *Proceedings of the Joint Workshop on NLP&LOD and SWAIE SemanticWeb, Linked Open Data and Information Extraction*, 2013.

[23] M. Du, M. Pierce, L. Pivovarova, and R. Yangarber, "Supervised classification using balanced training," in *Statistical Language and Speech Processing*. Springer, 2014, pp. 147–158.

[24] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*. IEEE, 1995, pp. 3–14.