LETTER

Extraction from the Web of Articles Describing Problems, Their Solutions, and Their Causes

Masaki MURATA[†a)], ***Member***, **Hiroki TANJI**[††], **Kazuhide YAMAMOTO**[††], **Stijn DE SAEGER**[†††], **Yasunori KAKIZAWA**[†††], ***and*** **Kentaro TORISAWA**[†††], ***Nonmembers***

SUMMARY In this study, we extracted articles describing problems, articles describing their solutions, and articles describing their causes from a Japanese Q&A style Web forum using a supervised machine learning with 0.70, 0.86, and 0.56 F values, respectively. We confirmed that these values are significantly better than their baselines. This extraction will be useful to construct an application that can search for problems provided by users and display causes and potential solutions.

key words: *problems, solutions, causes, Web, learning*

1. Introduction

Blogs and Web bulletin boards that deal with problems have advanced in recent years. When we encounter problems, we also investigate them using the Web through such blogs and Web bulletin boards. Further, time and effort are required to gather necessary information because a vast amount of information exists on the Web. If the investigation target is narrowed by automatically extracting articles describing problems, then the investigation cost can be reduced.

In a previous study, De Saeger, et al. identified nouns and phrases concerning problems and object-problem pairs using lexico-syntactic patterns to find hyponyms of the term "problem" and the dependency structures between nouns and verbs [1]. In this way, expressions describing problems have been acquired. Torisawa, et al. developed a system called TORISHIKI-KAI [2] that can retrieve information on problems. TORISHIKI-KAI displays a word and the problems concerned with it and enables Web retrieval by using those words as keywords. However, their study limits concrete use to executing retrieval by keywords with such expressions, and the articles obtained by the retrieval do not necessarily describe problems. In our study, we employed a machine learning technique using the features of the acquired expressions and only extract articles describing problems from among Web document sets.

The information required by people who investigate problems is not necessarily limited to the instances of problems. They want to know the answers to such questions as "How should I solve this problem?" and "What caused it?" Thus, as the next step of the extraction of articles describing problems, we extract the articles that describe solutions and causes from articles dealing with problems.*

Baldwin, et al. studied troubleshooting using articles in the Linux forum [4] and focused on Linux problems. In contrast, our study is not limited to one specific field. Moreover, there are other differences between the two studies. Baldwin, et al. judged whether a thread (a set of documents that discuss a certain topic) includes a solution for a problem using such machine learning as SVM for the classification and such domain specific numeric features as the number of Linux distribution mentions and the proportion of words relative to the full thread. Their experimental results showed that the overall accuracies of their method were low and almost the same as a simple method that assigns the majority class in the training data set to all test instances.

Kim, et al. studied the extraction of problem/solution key phrases in patent documents and the use of extracted key phrases for patent retrieval and the automatic discovery of technology trends [5], [6]. They extracted problem/solution key phrases using language model probabilities and linguistic clues. Their study resembles ours because both handle the extraction of problems and solutions. The difference is that their study explicitly labeled problem and solution phrases, but our study used various features to classify documents.

2. Methods

2.1 Dictionary of Problems

We used the expressions for describing problems as the features of machine learning. We considered nouns, verbs, and adjectives for describing problems and manually selected the nouns describing problems acquired in a previous study by De Saeger, et al. [1] and the verbs that are dependent on those nouns. Adjectives with negative polarity in publicly available evaluation-expression dictionaries [7], [8] were extracted. Phrases describing problems, such as noun/adjective pairs, were found in evaluation-expression dictionaries [8], [9]. We also used patterns typically referring to problems, such as "*dekinai*" (cannot) and "*shinikui*" (difficult to do), and onomatopoeic words describing problems, such as "*bisho-bisho*" (a sound indicating that some-

Manuscript received September 17, 2010.

[†]The author is with Tottori University, Tottori-shi, 680–8552 Japan.

[††]The authors are with the Nagaoka University of Technology, Nagaoka-shi, 940–2188 Japan.

[†††]The authors are with the National Institute of Information and Communications Technology, Kyoto-fu, 619–0289 Japan.

a) E-mail: murata@ike.tottori-u.ac.jp

DOI: 10.1587/transinf.E94.D.734

*This paper is based on our previous conference paper [3].

thing got wet in the rain). This dictionary has 20,429 nouns, 2,790 verbs, 954 adjectives, 5,909 phrases, 14 patterns, and 110 onomatopoeic words.

2.2 Extracting Articles Describing Problems

Articles describing problems were extracted from Web documents using the following machine learning methods: maximum entropy (ME) [10] and support vector machine (SVM) [11]. In experiments using the SVM method, "C" = 1 and "d" = 1 or 2 were used, where "C" is the soft margin parameter and "d" is the dimension number of the polynomial kernel [12]. The features used in our experiments include the following:

- Article length
- Number of words
- Word unigrams, bigrams, and trigrams
- Character strings at the end of sentences
- Length of first and last sentences
- Number of words matching each problem dictionary
- Words matching each problem dictionary
- Number of words matching all problem dictionaries

We used the above features for the following reasons: We thought that an article describing a problem was likely to be longer because it will include a detailed explanation of the problem. So we used the first two features because they are characteristic expressions indicating a problem and such expressions are useful for extracting articles describing problems. We thus used the third feature. In Japanese, the main verbs and modality expressions are found at the end of sentences. Since such expressions are important, we used the fourth feature. We also believe that the first and last sentences in an article are important. Therefore, we used the fifth feature. Because we thought that an article describing problems includes many words in problem dictionaries, we used the last three features.[†]

We used a method that employs simple matches with a problem dictionary (called "Dic Match"). The method determines that an article addresses a problem if it contains at least a threshold number of expressions described in the dictionary. On the basis of preliminary experiments, we set the threshold to 2.

The example article below was determined to be about a problem because it included the four underlined expressions matching the problem dictionary:

> Example: After the security software was installed, I was confused because the power supply to the personal computer could not be cut because of obstruction by the other security software.

2.3 Extracting Articles Describing Solutions/Causes

Articles describing solutions and causes are extracted using machine learning methods resembling those described in Sect. 2.2 from a document set of previously extracted articles describing problems. We used the SVM and ME methods and the identical features used in Sect. 2.2. In the features, we used the dictionaries of problems in Sect. 2.1 for solutions/causes. We did not use the dictionaries of words indicating solutions/causes.

Table 1 Results for articles describing problems.

	Precision	Recall	F-value
Baseline	0.263	1.000	0.416
Dic Match	0.473	0.806	0.596
ME	0.592	0.840	0.695
SVM1	0.639	0.768	0.698
SVM2	0.633	0.715	0.671

3. Experiments

3.1 Extracting Articles Describing Problems

We experimentally extracted articles describing problems from the Japanese Web document set using the ME method, the SVM method, and Dic Match. The Web document set was Yahoo! Chiebukuro [13], which is a Q&A style Web forum (the Japanese version of Yahoo! Answers). Yahoo! Chiebukuro has three types of articles: "questions," "normal answers," and "best answers." Since the cases of problems were frequent in question-type articles, we only used them.

We used 2,000 question articles from Yahoo! Chiebukuro that were manually tagged to indicate whether they concerned problems. We evenly divided them into two data sets: A and B. Data sets A and B included 281 and 263 articles describing problems, respectively.

We first determined the thresholds for which the F value becomes the best in the ME and SVM methods in 10-fold cross validations using Data set A. The thresholds are a probability value in the ME method and indicate distance with a separation plane in the SVM method.

We made an open test using Data set A as training data and Data set B as test data. Table 1 shows the precision, recall, and F values of the acquired articles for the test data by ME, d = 1, and d = 2 of the SVM method (SVMs 1 and 2) and Dic Match. ME, SVM 1, and SVM 2 methods used the threshold at which the F values become the best for the ME and SVM methods in the cross-validation results. We used a baseline method where the system always classifies an instance as a "problem." Statistical significance was determined using the bootstrapping method [14].[††] The F

[†] Analysis of the actual useful features is described in Table 3 of Sect. 4.

[††] In the bootstrapping method, we assume that we want to compare Methods 1 and 2. We randomly and redundantly extract N data items from an evaluated data set. N is the number of all data items in an evaluated data set. We repeat this 10,000 times and obtain 10,000 data sets. We obtain the F values of Methods 1 and 2 for 10,000 experiments using 10,000 data sets. In 10,000 experiments, we calculate the ratio in which Method 1 obtains higher F values than Method 2. When the value exceeds 0.95, we can roughly estimate that Method 1 outperforms Method 2 at a significance level of 0.05.

values of the machine learning methods were significantly better than those of the baseline method and Dic Match at a significance level of 0.05.

3.2 Articles Describing Solutions and Causes

Figure 1 shows the experimental recall-precision curve using SVM 1 with changing thresholds that extracts articles describing problems. When the recall ratio was 0.2, the precision ratio was about 0.9. We believe that we can acquire large amounts of data by maintaining a high precision ratio because Yahoo! Chiebukuro has a large collection of documents. We used SVM 1 to acquire Yahoo! Chiebukuro articles containing descriptions of problems and their solutions. By adjusting the SVM decision boundary to maintain a constant 90% precision, we obtained a set of 9,313 problem articles out of 100,000 candidate articles on Yahoo! Chiebukuro. The solutions and causes of problems are typically found in best-answer articles. We thus used article sets consisting of extracted question articles and their best answers.

From the 9,313 articles, we used 2,000 articles of Yahoo! Chiebukuro that were manually given tags, indicating whether they described solutions/causes. We evenly divided them into two data sets: C and D. Data sets C and D include 608 and 571 articles describing the solutions of problems and 325 and 273 articles describing the causes of problems.

A 10-fold cross validation was performed using Data set C. We determined the threshold at which F values became the best in the ME and SVM methods using 10-fold cross validation, as described in Sect. 3.1.

We made an open test using Data set C as training data and Data set D as test data. Table 2 shows the precision, recall, and F values of the extractions of articles describing the solutions/causes of problems by the ME, SVM 1, and SVM 2 methods. We used a baseline method where the system always classifies an instance as a "solution" for "solution" experiments and "cause" for "cause" experiments. With the bootstrapping method, we confirmed that the F values of the machine learning methods significantly outperformed those of the baseline at a significance level of 0.05.

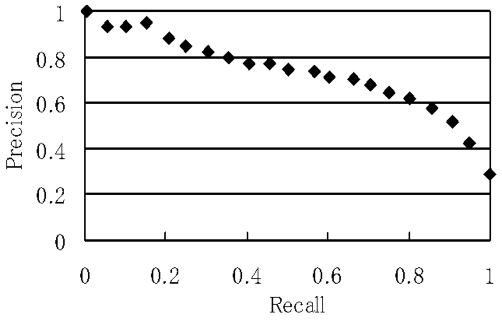

Fig. 1 Recall-precision curve of problems.

Table 2 Results for articles describing solutions/causes.

	Solutions			Causes		
	Precision	Recall	F-value	Precision	Recall	F-value
Baseline	0.571	1.000	0.727	0.273	1.000	0.429
ME	0.795	0.928	0.856	0.427	0.751	0.544
SVM1	0.743	0.953	0.835	0.431	0.810	0.562
SVM2	0.724	0.970	0.829	0.397	0.861	0.543

4. Discussion

The maximum entropy method can obtain the values of $\alpha_{a,j}$ for category a and feature j [15]. We normalized the α values for each feature so that the sum of the αs for all categories equals 1. Feature j with a higher normalized $\alpha_{a,j}$ value is found to be more important for the system to judge that the category of a data item with feature j is a. Table 3 shows the normalized α values of features calculated in experiments that extract articles describing problems/solutions/causes.

Features describing the total number of dictionary matches significantly influence the judgment of "problems." We also found that articles including adversative conjunctions are likely to describe problems such as "Recently, I bought a PC but the DVD drive was not attached!" Moreover, we found that articles including "*wakarimasen*" (don't understand) are easily judged to be "problems" and articles including question marks are easily judged to be "not a problem." Articles including "*wakarimasen*" are judged to be "problems" because Yahoo! Chiebukuro contained many sentences, such as "I don't know whether the bag is real or fake because it was a gift." It was easy to judge that articles including "?" are "not a problem" because many questions

Table 3 Features with high αs for problem/solution/cause.

Features with High αs for Problem	
Examples of features	α (Problem)
Matches with all dictionaries	0.662
"*nodesuga*" (but/however)	0.617
"*wakarimasen*" (don't understand)	0.615

Features with high αs for not a problem	
Examples of features	α (Not a problem)
"*tte*" (is)	0.634
"?"	0.613

Features with high αs for Solution	
Examples of features	α (Solution)
"*houhou*" (method/way)	0.606
"*nara*" (if)	0.589
"*mashou*" (let's)	0.573

Features with High αs for Not a Solution	
Examples of features	α (Not a Solution)
"*aru*" (exist)	0.575
"*nodesu*" (is)	0.553

Features with high αs for cause	
Examples of features	α (Cause)
Matches with noun dictionary	0.587
"*kara*" (because)	0.575
"*iru*" (is)	0.567

Features with high αs for not a cause	
Examples of features	α (Not a cause)
"*watashi*" (I)	0.565
"*mashita*" (was)	0.562

are general knowledge questions, such as "What are the recommended ingredients in Japanese hotchpotch?" This tendency is believed to be particular to question-answer type bulletin boards.

In terms of solutions, articles including "*nara*" (if) are easily judged as "solutions," for example, "If the network cable is pulled out and the symptom is resolved, you must back it up and initialize or restore it." Solutions often follow expressions indicating the action: "*mashou*" (let's).

In terms of causes, the feature describing the number of matches with the noun dictionary largely influences the judgments of the causes of problems. Many answers that explain the causes of problems were brief and used such words to describe the problems as "spyware." For example, "It is spyware!" Answer articles including first-person pronouns generally do not describe causes because the experiences of respondents do not necessarily describe the causes of problems.

We used problem dictionaries for "solution/cause." In our experiments, dictionaries were useful for "solution/cause." For example, for "cause," the feature of a noun dictionary for problems has a high α value, as in Table 3, and was useful for cause identification. For "solution," the feature of a verb dictionary for problems has a relatively high α value and was useful for solution identification.

5. Conclusion

In this study, we extracted articles that describe problems from question articles of Yahoo! Chiebukuro using machine learning methods. Then we extracted articles that describe solutions and the causes of problems from best-answer articles corresponding to the question articles that describe problems. As a result, we extracted articles describing problems at an F value of 0.7 and identified articles describing their solutions at an F value of 0.86 and those describing their causes at an F value of 0.56. We also examined the crucial features for extracting documents.

References

[1] S. De Saeger, K. Torisawa, and J. Kazama, "Looking for trouble," Coling 2008, pp.185–192, 2008.

[2] K. Torisawa, S. De Saeger, Y. Kakizawa, J. Kazama, M. Murata, D. Noguchi, and A. Sumida, "Torishiki-kai, an autogenerated web search directory," ISUC 2008, pp.179–186, 2008.

[3] H. Tanji, M. Murata, Y. Kakizawa, S. De Saeger, K. Torisawa, and K. Yamamoto, "Extraction of sentences expressing troubles from the Web," pp.140–143, 2009.

[4] T. Baldwin, D. Martinez, and R.B. Penman, "Automatic thread classification for linux user forum information access," The Twelfth Australasian Document Computing Symposium (ADCS 2007), pp.72–79, 2007.

[5] Y. Kim, J. Ryu, and S.H. Myaeng, "A patent retrieval method using semantic annotations," Proc. Conference on Knowledge Discovery and Information Retrieval (KDIR 2009), pp.211–218, 2009.

[6] Y. Kim, Y. Tian, Y. Jeong, R. Jihee, and S.H. Myaeng, "Automatic discovery of technology trends from patent text," Proc. ACM SAC'09, pp.1480–1487, 2009.

[7] H. Takamura, "Semantic orientations of words," http://www.lr.pi.titech.ac.jp/~takamura/pndic_en.html, 2005.

[8] N. Kobayashi, "Evaluation-expression dictionary," http://www.syncha.org/evaluative_expressions.html, 2006.

[9] N. Kaji, "Polar phrase dictionary," http://www.tkl.iis.u-tokyo.ac.jp/~kaji/polardic/, 2006.

[10] M. Utiyama, "Maximum entropy modeling package," http://www.nict.go.jp/x/x161/members/mutiyama/softwa-re.html#maxent, 2006.

[11] T. Kudoh, "TinySVM: Support vector machines," http://www.chasen.org/~taku/software/TinySVM/, 2000.

[12] T. Kudo and Y. Matsumoto, "Japanese dependency analysis based on support vector machines," EMNLP/VLC 2000, 2000.

[13] Yahoo! Japan Corporation, "Yahoo! Chiebukuro data for research laboratories offer data specifications NII offer version, ver. 1.0," 2007.

[14] B. Efron, "Bootstrap methods: Another look at the jackknife," The Annals of Statistics, vol.7, no.1, pp.1–26, 1979.

[15] M. Murata, R. Nishimura, K. Doi, T. Kanamaru, and K. Torisawa, "Analysis of the degree of importance of information using newspapers and questionnaires," Proc. IEEE NLP-KE 2008, pp.137–144, 2008.