Multi-aspects Review Summarization Based on Identification of Important Opinions and their Similarity *

Ryosuke Tadano, Kazutaka Shimada, and Tsutomu Endo

Department of Artificial Intelligence, Kyushu Institute of Technology,
680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan
{r_tadano, shimada, endo}@pluto.ai.kyutech.ac.jp

Abstract. The development of the Web services lets many users easily provide their opinions recently. Automatic summarization of enormous sentiments has been expected. Intuitively, we can summarize a review with traditional document summarization methods. However, such methods have not well-discussed "aspects". Basically, a review consists of sentiments with various aspects. We summarize reviews for each aspect so that the summary presents information without biasing to a specific topic. In this paper, we propose a method for multi-aspects review summarization based on evaluative sentence extraction. We handle three features; ratings of aspects, the tf-idf value, and the number of mentions with a similar topic. For estimating the number of mentions, we apply a clustering algorithm. By integrating these features, we generate a more appropriate summary. The experiment results show the effectiveness of our method.

Keywords: sentiment analysis, multi-aspects review summarization, ratings of aspects, important sentence extraction, opinion integration.

1 Introduction

As Web services like CGMs have become widely used, people can easily post reviews for products or services. Although handling these information (evaluative information) has become necessary, there exists too much information on the Web. Therefore, extracting information that users want and summarizing them have been expected recently. Intuitively, we can summarize a review with traditional document summarization methods. For instance, Brandow $et\ al.$ (1995) have summarized a document by extracting sentences with some features such as the presence of signature words and the location in the document. For sentiment summarization, Pang and Lee (2004) have extracted all subjective sentences. They suggested that these extracted sentences could be used as summaries. However, a review basically consists of sentiments with various aspects (i.e., "image quality" and "usability" of a camera). Therefore, we need to extract information for each aspect in the case of review summarization. Aspect summarization can present information without biasing to a specific topic. We focus on multi-aspects review summarization in this research.

We generate a summary by extracting important sentences and arranging them. Since we extract summary sentences, we need to discuss which sentences are important and how to extract important sentences. In the case of treating multi-reviews, we need to handle the redundant information. Pang and Lee (2008) have reported that while in traditional summarization redundant information is discarded, in sentiment summarization redundancy indicates the importance of opinions. Therefore, we treat redundancy as a feature for decision of important sentences. We assume that reviews we treat in this research have multiple aspects and a reviewer gives ratings of the aspects (i.e., 0 to 5 stars). Reviewers also write free comments about the target. We leverage

* This work was supported by Kayamori Foundation of Informational Science Advancement

Copyright 2010 by Ryosuke Tadano, Kazutaka Shimada, and Tsutomu Endo

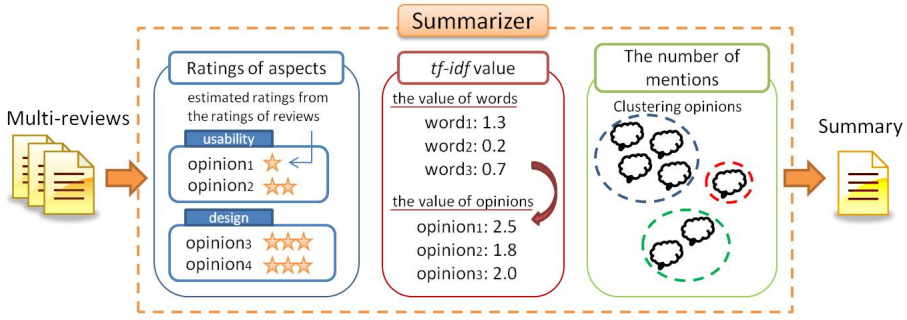

Figure 1: The outline of our method.

three features: ratings of aspects of reviews, the tf-idf value, and the number of mentions with a similar topic. We apply a clustering algorithm to sentences to measure the number of mentions with a similar topic. Then, we aim to integrate those features and generate a more appropriate summary. Figure 1 shows the outline of our method.

2 Features for sentence extraction

For generating a summary, we define the following three features: (1) ratings of aspects, (2) the tf-idf value, and (3) the number of mentions with a similar topic. The following sections describe how we treat these features.

2.1 Ratings of aspects

If we generate a summary, the summary needs to contain the proper balance of whole opinions in the reviews. For instance, if we summarize only positive opinions, the summary cannot tell readers negative opinions of a target.

We focus on ratings of aspects given to reviews as a feature to deal with this problem. We assume that a reviewer writes a comment corresponding to the rating. For instance, if a reviewer gives 1 star for one aspect, the reviewer writes negative comments for the aspect. We assign ratings of aspects of a review to evaluative sentences included in the review. Each evaluative sentence has the rating corresponding to the aspect of the sentence. We use the pair of the evaluative sentence and the rating to consider the distribution of the ratings for summarization.

2.2 The tf-idf value

The tf-idf algorithm is used as a major algorithm for many tasks such as important sentence extraction. This algorithm features words which only appear in a target document as more important. We similarly apply the tf-idf algorithm to compute the importance of an evaluative sentence. First, we divide sentences in reviews to morphemes by using a morphological analyzer[1]. Then, we define the tf-idf value of a word i in target reviews j as below. Note that we treat only content words (except words such as suffixes and pronouns).

$$tf_j^i \times idf_i = \frac{\log_2(freq(i,j)+1)}{\log_2(words(j))} \times \{\log_2(\frac{Rev_{all}}{Rev_{include(i)}}) + 1\} \quad (1)$$

where $freq(i,j)$ is the frequency of i for j, $words(j)$ is the number of words belonging to j, Rev_{all} is the number of all reviews, and $Rev_{include(i)}$ is the number of reviews including i.

Next, we compute the importance of an evaluative sentence by using the tf-idf value of a word. We denote an evaluative sentence by $S = \{w_1, w_2, ...\}$, where each w is a word which has tf-idf value in the sentence. The importance of an evaluative sentence tf-idf_S is as below.

$$tfidf_S = \frac{\sum_{w \in S} tfidf^w}{|S|} \quad (2)$$

[1] We used Mecab. http://mecab.sourceforge.net/

where $|S|$ is the total number of w in the sentence. This is the importance of an sentence based on feature words.

2.3 The number of mentions with a similar topic

We treat multi-reviews for the target of summarization. In this case, some reviewers might write similar opinions. These similar opinions have possibilities to be redundantly extracted as a summary. On the other hand, the opinion mentioned by many reviewers is important. We need to handle redundant information regarding as a feature to determine an important sentence. Therefore, we aim to integrate similar opinions by clustering them.

In this paper, we apply the k-$means$ algorithm which is widely used as the clustering algorithm because of its simplicity. Since the k-$means$ algorithm is a non-hierarchical method, firstly we need to specify how many clusters we divide. However, It is difficult to know the optimum number of divided clusters beforehand. Seki $et\ al.$ (2007) has estimated the valid number of clusters by statistically evaluating the clustering result. We apply their algorithm to our task.

We divide evaluative sentences to morphemes and construct a vector space using these morphemes as features. Note that we treat content words, adjectives, and verbs. A score of each feature is the tf-idf value computed in section 2.2 if the feature has the tf-idf value. If a feature does not have the tf-idf value, we simply assign the frequency within the evaluative sentence as the score. Besides, we introduce the concept of centrality of the word which has been reported by Ishii $et\ al.$ (2001) to characterize features. This method assumes that words such as a subject case and an objective case in a sentence indicate a topic of the sentence. We also apply the concept to our task and weight central words in the evaluative sentence. By clustering evaluative sentences based on these algorithms, we can generate some clusters including similar sentences.

However, we found that clusters generated by our algorithm tended to be divided too much in a preliminary experiment. It denotes that similar opinions which should belong to the same cluster belong to other clusters.

We revise the clusters by using another approach. In the algorithm mentioned above, each cluster is formed with evaluative sentences which are gathered by co-occurrence of some words. Therefore, each cluster contains words which appear in common in the cluster. We identify representative words of each cluster by using this feature. If a cluster contains one sentence, we regard all nouns and adjectives in the sentence as representative words. In other cases, we regard words which appear more than half of the number of sentences in the cluster as representative words for the cluster. Using the representative words, we integrate clusters which contain the same representative words. Since a cluster often contains some representative words, we need to decide which cluster is the best for integration. In this case, we compare the tf-idf value of representative words and integrate clusters by using the representative word with the top tf-idf value.

3 Summarizer

In this section, we describe how to generate a summary based on three features mentioned in section 2. We compute the importance of each cluster by integrating the tf-idf value of each sentence and the number of mentions. The importance of a cluster $Imp(C)$ is as below.

$$Imp(C) = \frac{\sum_{S \in C} tfidf_S}{|C|} \times \log(|C| + 1) \tag{3}$$

where $C = \{S_1, S_2, ...\}$ is a cluster, each S is an evaluative sentence, $|C|$ is the total number of sentences in the cluster. The $Imp(C)$ is the importance which means both the importance of feature words and the number of mentions.

Besides, we treat ratings of aspects to reflect the proper balance of whole opinions of reviews. The process for the sentence extraction is as follows:

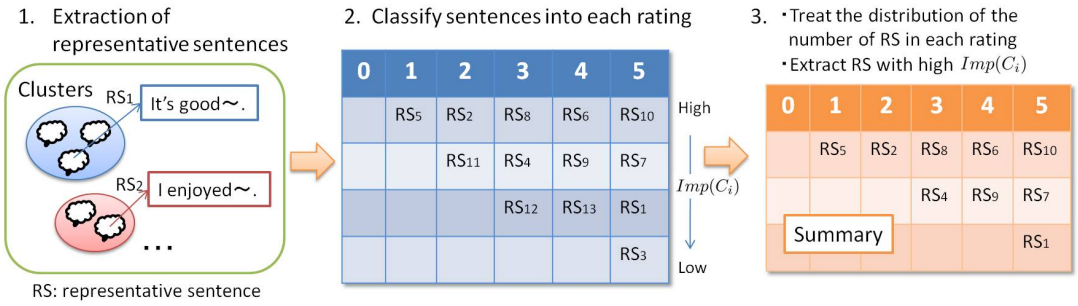

Figure 2: The process of our summary generation.

1. identify a representative sentence which has the top tf-idf from each cluster,
2. classify representative sentences into the rating to which the sentence belongs,
3. extract sentences with high $Imp(C)$ on the basis of the distribution of the total number of representative sentences belonging to each rating.

Figure 2 shows the process of our summary generation. We generate a summary of target reviews by arranging extracted sentences.

4 Experiment

In this section, we evaluated our summarization. First, we describe the data set for the experiment. Then, we evaluate the effectiveness of our clustering method. After that, we generate a summary with our method. We subjectively evaluate the effectiveness of using the three features. Finally, we compare our summary with a manual summary for quantitative evaluation.

4.1 Data set

We used game review documents as target data. The review documents were extracted manually from the Web site[2]. Seven evaluative criteria are given to each review, i.e., <Originality (o)>, <Graphics (g)>, <Music (m)>, <Addiction (a)>, <Satisfaction (s)>, <Comfort (c)>, and <Difficulty (d)>. We chosen one of the Nintendo DS software "New Super Mario Brothers" as the target data. There were 170 reviews for this game.

The target data was annotated beforehand. In this experiment, three annotators (A_1, A_2, A_3) annotated the target reviews. They annotated 25 reviews (approximately 450 sentences) in 170 reviews. Annotators detected evaluative sentences from the reviews and annotated aspects to which the sentences belong. We target the 25 reviews for summarization[3].

For quantitative evaluation, annotators manually generated summaries. Annotators extracted 50 sentences as a summary from the 450 sentences in the 25 reviews. There is no limit how many sentences they extract for each aspect.

4.2 Evaluation of clustering

At first, we evaluated the effectiveness of our clustering method. We classified evaluative sentences detected from reviews into the seven aspects. The clustering was independently-applied in each aspect. We assigned two sentences with the highest and the lowest tf-idf value as initial clusters.

Table 1 shows the change of the number of clusters in each aspect. "original" is the number of evaluative sentences (one sentence as one cluster), "k-$means$" is the number of clusters which were applied the original k-$means$ algorithm, and "revised" is the number of clusters which were integrated by our method. The number of clusters in the revised step decreased to approximately

[2] http://ndsmk2.net/
[3] Note that we computed the tf-idf value with 170 reviews.

Table 1: The number of clusters in each step.

Annotator		a	c	d	g	m	o	s
A_1	original	50	83	49	26	18	124	90
	k-$means$	17	54	28	13	7	40	24
	revised	14	27	16	10	4	20	20
A_2	original	43	73	67	33	19	160	115
	k-$means$	29	42	42	19	11	39	29
	revised	22	23	16	11	7	26	23
A_3	original	46	83	51	27	15	54	198
	k-$means$	19	33	21	17	6	25	89
	revised	16	20	15	11	5	15	50

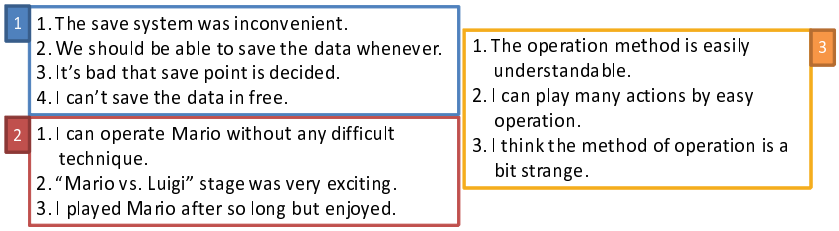

Figure 3: An example of integrated clusters after revision.

30 % of the original sentences. We verified the decrease of clusters between the k-$means$ step and the revised step. In our subjective evaluation, the revised step could form better clusters than the k-$means$ step. Therefore, we conclude that the revision of the clustering was effective.

Figure 3 shows an example of the clusters in the aspect <Comfort (c)>[4]. Each labeled box ($1-3$) was a cluster. The cluster 1 was a successful example. In this aspect, there were 16 sentences which mentioned a key "save" and 15 sentences were integrated into the same cluster. In this case, all sentences in this cluster indicated a similar opinion. The "save" was suitable for a topic of the summarization. On the other hand, in the cluster 2, many sentences were integrated into the same cluster by co-occurrence of the word "Mario". We think the reason that "Mario" tended to frequently appear in all the reviews because of the nature of the product, and it had the high tf-idf value. However, "Mario" was not a topic for the summarization. We need to identify words which widely appear and decrease the importance of such words. Besides, the current clustering method did not handle a polarity of a sentence. As seen in the cluster 3, a cluster might consist of different polar opinions. We need to deal with the problem; for instance, by using the rating information as a feature of the clustering.

4.3 Evaluation of a summary

Using the result of clustering, we generated a summary. For evaluating the effectiveness of each feature, we summarized 25 reviews with the following three methods.

Meth1. only the tf-idf value

Meth2. the tf-idf value and $Imp(C)$

Meth3. **Meth2** and rating information

Table 2 shows an example of **Meth3**'s summary for the aspect <Addiction (a)> with A_1's annotation. The "rating" is the actual rating of the review containing the representative sentence, "Ave" is the average of the ratings in the cluster containing the representative sentence, and "STD" is the standard deviation of the ratings in the cluster.

[4] Actually all the sentences in the target data are written in Japanese.

Table 2: A summary by **Meth3** for <Addiction (a)>.

Representative sentence	$Imp(C)$	Rating	Ave	STD
I can play more if I completed to collect star coins.	5.001	5	4.286	1.030
The feature of Mario series is that we are not tired.	3.833	5	4.000	0.816
It feels good to break a stage with big Mario.	5.994	4	3.600	1.020
I often feel it monotonous play.	1.773	3	3.000	0.000
I was addicted in collection of coins.	8.328	2	3.273	1.286
It's fatal that this game does not have other exciting features.	3.445	0	2.000	1.633

Table 3: A summary by **Meth1**

Extracted sentences
I was addicted in collection of coins.
Big mushroom item is very funny idea.
Challenge of Star coins collection is good.
It feels good to break a stage with big Mario.
Collection of Star coins is good idea.
I enjoyed collecting Star coins.

Table 4: A summary by **Meth2**

Extracted sentences
I was addicted in collection of coins.
It feels good to break a stage with big Mario.
I can play more if I completed to collect star coins.
I was bored after I cleared this game once.
The feature of Mario series is that we are not tired.
We easily play this game with no stress.

We discuss the effectiveness of **Meth3**. One of the problems in the result of Table 2 was that there existed the sentence to which its rating did not correspond. For example, although "I was addicted to collect coins." was usually considered as a positive opinion, the rating of the review containing the sentence was 2. The reason was that most of the sentences in the review containing the representative sentence had negative opinions for this aspect. This is due to the fact that opinions of reviewers sometimes might be inconsistent with the ratings. We need to handle the non-consistence of ratings if we treat the rating information. We verified the difference between the rating of the representative sentence and the average rating in the cluster. STD in Table 2 also indicated that ratings in some clusters varied widely. We think the reason that we do not treat the polarity for the clustering or opinions in the cluster were not similar because clustering accuracy was not enough. We need to discuss a more high-accuracy clustering approach and an algorithm for representative sentence extraction.

We compared the summaries of each method. Summaries by **Meth1** and **Meth2** are shown in Table 3 and Table 4. As seen in Table 3, we verified that the summarization by **Meth1** contained some sentences about "star coin". It also consisted of only positive opinions. As compared with **Meth1**, **Meth3** could extract the proper balance of opinions. It shows that opinion integration and using ratings were effective. On the other hand, there were no great difference between **Meth2** and **Meth3**. Both methods use the same representative sentences and select them with the high $Imp(C)$. Therefore, both summarizations became similar. We need to use ratings more effectively. For instance, by using the rating information for the determination of representative sentences, **Meth3**'s result might become better.

4.4 Quantitative evaluation

We compared our summaries with manual summaries. We extracted 50 sentences by using **Meth1**, **Meth2**, and **Meth3**. We estimated how many sentences we extract for each aspect by using the distribution of the number of evaluative sentences in each aspect. The estimation approach was simple idea but we verified the result was relatively close to the manual one.

We used ROUGE-N (Lin, 2004) for evaluation of summaries. It indicates an n-gram recall between reference summaries and a candidate summary. ROUGE-N is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in S_H} \sum_{g_N \in S} C_m(g_N)}{\sum_{S \in S_H} \sum_{g_N \in S} C(g_N)} \tag{4}$$

Table 5: ROUGE-N between our summaries and manual summaries.

		a	c	d	m	g	o	s	Ave_{wgt}
N=1	Meth1	0.287	0.496	0.347	**0.421**	0.082	0.363	0.383	0.345
	Meth2	**0.435**	**0.539**	0.333	0.186	**0.137**	**0.372**	0.414	0.362
	Meth3	0.413	0.487	**0.350**	0.342	0.127	0.363	**0.422**	**0.367**
N=2	Meth1	0.117	**0.207**	0.118	**0.370**	0.013	**0.204**	0.103	0.150
	Meth2	**0.297**	0.203	0.127	0.039	**0.040**	0.176	0.112	0.144
	Meth3	0.223	0.146	**0.130**	0.287	0.030	0.178	**0.115**	**0.151**

Table 6: ROUGE-N between annotators.

	a	c	d	m	g	o	s	Ave_{wgt}
N=1	0.440	0.443	0.402	0.543	0.900	0.459	0.467	0.480
N=2	0.232	0.234	0.226	0.337	0.808	0.224	0.184	0.263

where S_H is a set of reference summaries, g_N is the N length n-gram, $C(g_N)$ is the frequency of the g_N in the reference summary, and $C_m(g_N)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries.

Table 5 shows ROUGE-1 and ROUGE-2 between our summaries and manual summaries. Those scores are the average scores in three annotators. Since the distribution of each aspect was different, we computed the weighted average Ave_{wgt}. We weighted the number of sentences in reference summary for each aspect.

Meth3 could achieve the best Ave_{wgt} score but there was no significant difference between three methods. We need to discuss two matters. One is the effectiveness of using the rating information. **Meth3** summarized reviews with the rating information but the result showed that it was not effective. We think the reason that we used rating information only for summary sentence extraction. We can treat the rating information for our clustering method and representative sentence extraction. We need to improve the use of the rating information.

The other problem is that $Imp(C)$ did not work effectively. We used $Imp(C)$ for **Meth2** and **Meth3** but the results were not different from **Meth1**. We think the reason that the representative sentences were not appropriate. The current method decides a representative sentence of a cluster by using only tf-idf value of the sentence. The representative sentence might not reflect whole opinions in the cluster. We need to introduce other features, for instance, a centroid of a cluster, a length of a sentence, and phrase structures.

We computed ROUGE between annotators as reference (Table 6). We regarded A_1 as the reference summary and the scores are the average of A_2 and A_3. While most of the scores were higher than the result of Table 5, the scores themselves were not so high. This result shows that it is difficult to generate a same sentiment summary even if between humans[5].

5 Related work

Meng and Wang (2009) have extracted aspects from the specification of the target product and summarized reviews with hierarchic structures. As a result, they could extract appropriate aspects for the products. However, the generated summary did not include detailed opinions about the product. In contrast, our method can treat detailed information by extracting important sentences with feature words. Blair-Goldensohn $et\ al.$ (2008) have computed a polarity value of sentences based on the maximum entropy method with WordNet and rating information. They extracted evaluative sentences with a high polarity value preferentially and generated a summary. As the advantage of their method, it could estimate the polarity of sentences with high accuracy. However,

[5] Note that this result contains the following difficulty; (1) detect same evaluative sentences from reviews, (2) assign same aspects to evaluative sentences, and (3) select same summary sentences for each aspect.

it is not always true that sentences with high polarity values are appropriate for a summary. They also did not treat redundancy of the summary. Lu and Zhai (2008) have introduced the concept of aspects to a PLSA model. They integrated expert reviews and ordinary opinions scattering in the Web. Opinions which should be integrated are identified by measuring the number of mentions in the Web. Although their method are very effective, their purpose is to add more information, such as similar and supplementary opinions, to the base review.

In this paper, we did not discuss the identification of the aspect of sentences. However, it is important to identify the aspect information for the sentiment summarization task. Hadano $et\ al.$ (2010) have identified an aspect of an evaluative sentence with machine-learning approaches. We need to introduce such a method to our task in the future.

6 Conclusion

In this paper, we focused on the multi-aspects review summarization. We handled three features; ratings of aspects, the tf-idf value, and the number of mentions with a similar topic. We used the clustering method to integrate similar opinions. The experiment result showed that we could integrate similar opinions and it led to the redundancy elimination of a summary. We evaluated the effectiveness of using all three features for summarization. In our subjective evaluation, we verified that the summarization by using the three features was effective. However, in quantitative evaluation, there was no great difference between the summary using all three features and summaries using only one or two features. Future work includes (1) improvement of the clustering algorithm, (2) more appropriate decision of representative sentences, and (3) treating the rating information more effectively.

References

Blair-Goldensohn, Sasha, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A. Reis, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. In *Proceedings of WWW 2008: NLPIX Workshop*.

Brandow, Ronald, Karl Mitze, and Lisa F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31(5), 675–685.

Hadano, Masashi, Kazutaka Shimada, and Tsutomu Endo. 2010. Aspect identification of sentiment sentences using a clustering algorithm (in japanese). In *Proceedings of FIT 2010*.

Ishii, Hiroshi, Rihua Lin, and Teiji Furugori. 2001. An automatic text summarization system based on the centrality of word roles in sentences (in japanese). *Information Processing Society of Japan SIG Notes*, 2001(20), 83–90.

Lin, Chin-Yew. 2004. Looking for a few good metrics: Automatic summarization evaluation – how many samples are enough? In *Proceedings of NTCIR Workshop 4*.

Lu, Yue and Chengxiang Zhai. 2008. Opinion integration through semi-supervised topic. In *Proceedings of WWW 2008*, pp. 121–130.

Meng, Xinfan and Houfeng Wang. 2009. Mining user reviews: from specification to summarization. In *Proceedings of ACL-IJCNLP 2009*, pp. 177–180.

Pang, Bo and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of Association for Computational Linguistics (ACL)*, pp. 271–278.

Pang, Bo and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2 (1-2), 1–135.

Seki, Tsunehito, Kazutaka Shimada, and Tsutomu Endo. 2007. Acquisition of synonyms using relations in product. *Systems and Computers in Japan*, 38(12), 25–36.