A Product Retrieval System Robust to Subjective Queries

Kenji Sugiki[†] **and Shigeki Matsubara**[‡]

[†]Graduate School of Information Science,
Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, Japan
E-mail: sugiki@el.itc.nagoya-u.ac.jp
[‡] Information Technology Center,
Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, Japan
E-mail: matubara@nagoya-u.jp

Abstract: In this paper, we propose a method for retrieving products that match users' search queries written in natural language. We use a large amount of product reviews written by consumers, in order to match those queries with products. Since reviews include a lot of valuable information which commerce service providers don't supply, by using these reviews our system can respond to a wide variety of users' requests, especially to subjective requests. If a review includes a description that coincides with the condition described in a user's query, the product to which the review refers is considered the product that matches the query and is shown to the user by our system. Using our proposed method, we implemented an accommodation retrieval system named "Yado-tan". The result of search experiments using natural language queries confirmed the retrieval capability of using our method.

Keywords: product retrieval system; customer product reviews; natural language query; subjective query.

Reference to this paper should be made as follows: Sugiki, K. and Matsubara, S. (2008) 'A Product Retrieval System Robust to Subjective Queries', *Int. J. Product Lifecycle Management*, Vol. 3, Issue 2/3, pp.151–164.

Biographical Notes: Kenji Sugiki received his B.E. and M.E. degrees in information engineering and sciences from Nagoya University, Japan, in 2005 and in 2007, respectively. He is currently a Ph.D. student, at Nagoya University. His research interests include information retrieval, text mining, natural language processing.

Shigeki Matsubara received the Dr. of Engineering degree in information engineering from Nagoya University in 1998. He was a Research Fellow of the JSPS from 1996 to 1998, and a Research Associate from 1998 to 2002 at the Faculty of Language and Culture, Nagoya University. Since 2002, he has been an Associate Professor of the Information Technology Center, Nagoya University. His research interests include

natural language processing, spoken language processing, and digital library. He is a member of the IEEE, the ACM, and the ACL.

1 Introduction

In recent years, such electronic markets as Amazon.com and eBay have grown rapidly due to increases of Internet users. These sites deal in vast amounts of products and need to provide useful search environments for their users. Most sites, however, just search for prefixed data items such as product names, categories, and features. Since user needs may be diverse or highly subjective, they often cannot translate their needs into the queries which these search systems allow to be inputed.

On the other hand, there has been studies on a natural language interface for a product retrieval system. Dittenbach et al. (2003) proposed the system which allows users to describe their needs with natural language in an accommodation domain. In addition, recommendation systems using a dialogical approach have been proposed by Chai et al. (2002); Mcsherry (2005). Since the systems transforms natural language queries into database queries (e.g., *SQL*), it is difficult, however, to respond to the variety of natural language expressions. It is difficult to create transformation rules for exhaustive and sometimes subjective queries which can hardly be transformed by the systems

In this paper, we propose a product retrieval system robust to subjective queries written in natural language. We use consumers' reviews in order to match the natural language queries to appropriate products. Recently, a large number of reviews are generically offered by commerce service sites (e.g., Amazon.com, eBay), review sites (e.g., Epinions.com, Reviewcentre.com), and so on. Additionally, there are a large number of blogs whose authors post product reviews. Since consumers often use reviews when they decide whether to purchase a product or not, we assume that reviews are also useful for product retrieval tasks. By using reviews our system responds to a variety of user requests. The recommendation system using product reviews are proposed by Aciar et al. (2006). The respect of ranking products using the reviews is the same as this paper. However, this paper is the first attempt to build a product retrieval system based on users' queries.

We developed a product retrieval system for Japanese accommodation domain in order to present the retrieval capability of our method especially for diverse or highly subjective queries. In retrieval experimental results using 700,826 reviews for 20,588 accommodations from commerce service site, we confirmed that the system was able to respond to subjective queries such as "*beddo-ga hiroku-te negokochi-ga yoi yado* (*I am looking for a hotel with wide bed where I can sleep comfortably.*)" using large amounts of reviews.

This paper is organized as follows: Section 2 briefly gives on overview of our method and Section 3 describes a product retrieval system using the method. Section 4 describes the prototype system and reports our experiments and results. Finally, Section 5 shows conclusion and future works.

2 Query and Review

Our method retrieves the products whose reviews correspond to a natural language query. Figure 1 shows an example of a query and a review. In this example, we assume that the content of the review corresponds to that of the subjective query and plasma TV "TH-42PX500" is the product which matches the query.

To determine the degree of the correspondence between a query and a product to which reviews refer, we transform those natural language sentences into semantic representations as shown below. We determine the 3-tuples "(*object*, *item*, *value*)" as the representations. Here, "*object*" is a product name or a product category, "*item*" is a feature of the product, and "*value*" is the feature's value. The reason why we use 3-tuple representations is that when the user search for products, he/she often use three components "targeted product category, feature (*item*) , and value" as search conditions. As for an example of "mp3 player whose color is red and with simple design and clear sound," 3 tuples "(*mp3_player*, *color*, *red*), (*mp3_ player*, *design*, *simple*), (*mp3_ player*, *sound*, *clear*)" are extracted from this query.

We determine 3-tuple for a review as "(*review-object*, *item*, *value*)" and call it a *review-tuple*. Here, a *review-object* is a product name. For example, review tuple "(*TH-42PX500*, *gashitsu* (*image_ quality*), *kirei* (*high*))" is extracted from the review "*gashitsu-ga kirei* (*image quality is high*)" about *plasma TV* "*TH-42PX500*". Additionally, 3-tuples are also extracted from queries as well as reviews. We determine 3-tuple for a query as "(*query-object*, *item*, *value*)" and call it a *query-tuple*. Here, a *query-object* is a product category. For example, query-tuple "(*purazuma-terebi* (*plasma_TV*), *gashitsu* (*picture_quality*), *kirei* (*high*))" is extracted from the query "*gashitsu-ga kirei-na purasuma-terebi* (*Plasma TV with high picture quality*)". Compared the query-tuple with the review-tuple in above two examples, *item* and *value* of the query-tuple is identical with that of the review-tuple and thus we can judge that plasma TV "TH-42PX500" matches the query.

Figure 2 shows query-tuples and review-tuples extracted from the query and the review shown in Figure 1. The review-tuples correspond to the query-tuples and so we can consider plasma TV "TH-42PX500" as the product which matches the query.

The 3-tuple representations of our method are the same as that used in related works for opinion extraction tasks in Nasukawa et al. (2003); Tateishi et al. (2004); Hu and Liu (2004); Liu et al. (2005); Posescu and Etzioni (2005); Kobayashi (2006). Those studies aimed to extract opinions and worked to strictly extract opinions and to develop extraction method and term dictionary for the opinion representations. Our method, however, does not require to limit opinions to be extracted and even if inappropriate review-tuples are extracted, such review-tuples do not match query-tuples of users' queries. In addition, since the purpose of our research is different from that of those studies, we focus on the product retrieval system using reviews. Therefore, it can be say that we use comparatively easy solution.

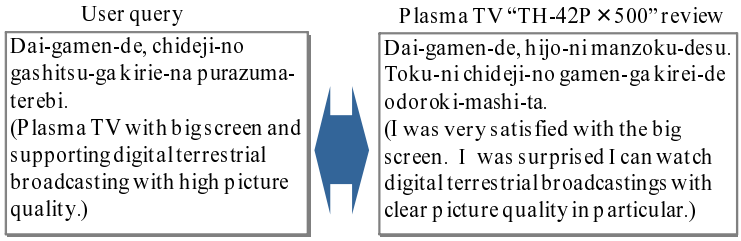

Figure 1 Example of a query and a review which matches the query

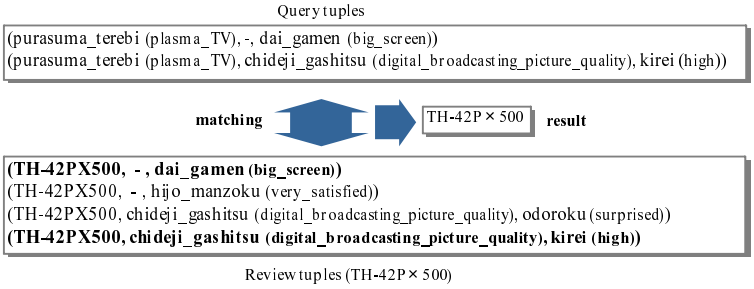

Figure 2 Example of matching between query-tuples and review-tuples

3 Product retrieval system using consumer reviews

3.1 Overview of product retrieval system

An overview of our system is given in Figure 3. Hereinafter, we will explain examples in the accommodation domain (to say, accommodation retrieval system). The system consists of the following two components: extraction step that the system extracts review-tuples from the review's body; and retrieval step that the system retrieves products which match a query.

In extraction step, the system extracts review-tuples from sentences of a review by applying transformation rules. The rules are based on the syntactic pattern of modification relations between *bunsetsus* [a]. In retrieval step, the system transforms a query into query-tuples, calculates the degree of correspondence of the each query-tuple to a review-tuple, and presents products which match the query.

3.2 Extraction method for review-tuples

In this study, we assume that the system uses reviews posted on commerce service sites and each review is written about a certain product. We determine the product to which the review refers as *review-object* and then the system extracts

[a] A *bunsetsu* is a Japanese phrasal unit that consists of one or more adjacent content words followed by any number of function words.

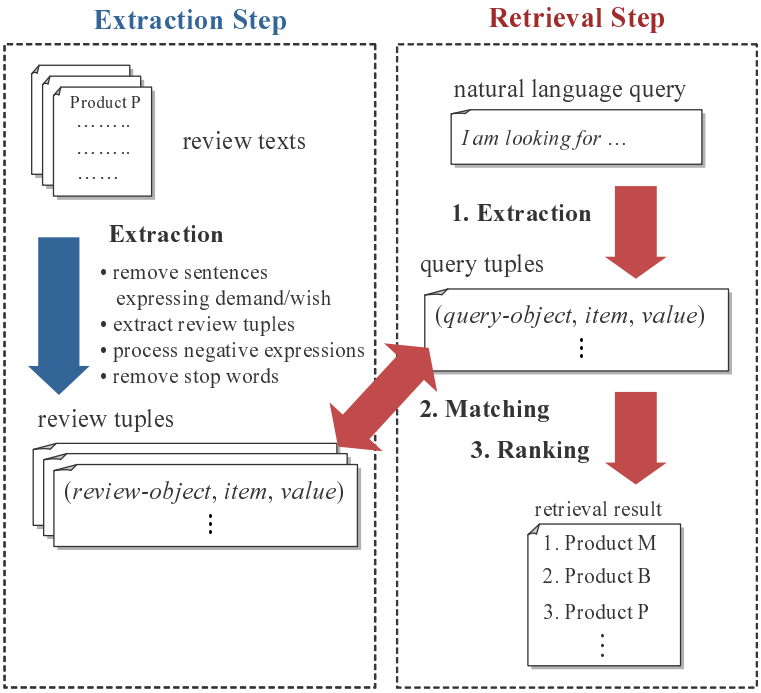

Figure 3 Overview of our product retrieval system

item-value pairs of review-tuples from the review's body. In what follows, we will describe the method for extracting the *item-value pairs*.

3.2.1 Transformation rules into review-tuples

Item-value pairs which appear in reviews mainly have following modification relations:

subject-predicate relation hotel A review: "*Heya-wa kirei-deshi-ta.* (*The room was clean.*)"
⇒ (*hotel A*, *heya* (*room*), *kirei* (*clean*))

modifier-head relation hotel B review: "*Totemo shinsetsu-na hoteru-jugyoin-deshi-ta.* (*She was a very kindly hotel staff.*)"
⇒ (*hotel C*, *hoteru-jugyoin* (*hotel staff*), *totemo shinsetsu* (*very kindly*))

where the above expression $A \Rightarrow B$ shows that review-tuples B is extracted from review A. In the former example, "*heya* (*room*)" is a item and a subject, "*kirei* (*clean*)" is a value and a predicate and then "*kirei*" depends on the "*heya*". In the latter example, "*totemo shinsetsu* (*very kind*)" modifies the head "*hotel jugyoin* (*hotel staff*)". From those relations we created 4 transformation rules from a review as shown in Figure 4. The rules are based on modification relations between *bunsetsus* and each *bunsetsu* included in the rules has the pattern constraints such as "noun + *wa/ga/mo* (case/dep. particle)" as shown in Figure 4. Here, $X \rightarrow Y$

(1)	$X \rightarrow Y \Rightarrow (O, X, Y)$ X: noun + *wa/ga/mo* (case/dependency particle) Y: verb, adjective, *sa-hen* noun + *suru* (auxiliary verb) e.g.) hotel A: *heya-ga* (*the room*) → *kirei-deshi-ta* (*was clean.*) ⇒ (*hotel A*, *heya* (*room*), *kirei* (*clean*))
(2)	$X \rightarrow Y_1 \rightarrow Y_2 \Rightarrow (O, X, Y_1), (O, X, Y_2)$ X: noun + *wa/ga/mo* (case/dep. particle) Y_1, Y_2: verb, adjective, *sa-hen* noun + *suru* (aux. verb) e.g.) hotel B: *heya-ga* (*the room*) → *kirei-de* (*was clean and*) → *kaiteki-deshi-ta* (*comfortable.*) ⇒ (*hotel B*, *heya* (*room*), *kirei* (*clean*)), (*hotel B*, *heya* (*room*), *kaiteki* (*comfortable*))
(3)	$Y \rightarrow X \Rightarrow (O, X, Y)$ X: noun + *wa/ga/mo/wo/ni/da/desu* (case/dep./target/contexual/direct-object particle or aux. verb) Y: adjective e.g.) hotel C: *kirei-na* (*clean*) → *heya-deshi-ta* (*room*) ⇒ (*hotel C*, *heya* (*room*), *kirei* (*clean*))
(4)	$Y \Rightarrow (O, , Y)$ Y: verb, adjective, *sa-hen* noun + *suru* (aux. verb) e.g.) hotel D: *shinsetsu-deshi-ta* (*kind.*) ⇒ (*hotel D*, , *shinsetsu* (*kind*))

O: *review-object*

Figure 4 Transformation rules into review-tuples

is that *bunsetsu* Y depends on *bunsetsu* X, $A \Rightarrow B$ is that if a sentence includes modification pattern A, then extract review-tuple(s) B.

In addition to the above rules, in order to extract more detailed information we adds some *bunsetsus* to *item* or *value*. In the case of rule 1 and 2, if the *bunsetsu* which includes the pattern "noun + *no* (dep. case)" depends on X or the *bunsetsu* which includes adverb or the pattern "noun + *ni/de/wo* (target/contextual/direct-object particle)" depends on Y, then those *bunsetsus* are added to X, Y, respectively.

For example, review-tuple "(*Nagoya Hotel*, *front-taio* (*front desk's service*), *shinsetsu* (*good*))" is extracted from modification relation "*front-no* (*front desk's*) → *taio* (*service*) → *shinsetsu* (*is good*)" in a *Nagoya Hotel* review. Review-tuple "(*Nagoya Hotel*, *konbini* (*convenience stores*), *chikaku-aru* (*close to the hotel*))" is also extracted from modification relations "*chikaku-ni* (*close to the hotel*) → *aru* (*is*)" and "*konbini-ga* (*convinience stores*) → *aru* (*is*)" in a *Nagoya Hotel* review.

3.3 Procedure of review-tuples extraction

Procedures of review-tuples extraction are shown below.

Preprocessing The system divides reviews' body into sentences and parses each sentence into a dependency tree using a dependency parser. Then the system removes the sentences which include such desire, demand or request expres-

sions as "$\cdots$ *shi-te hoshi* (*I want* $\cdots$ *to -*)", "$\cdots$ *nozomashi* (*I wish* $\cdots$)", and "$\cdots$ *ba ureshi* (*It would be nice if* $\cdots$)". Those sentences describe products' features which the products currently don't have and are inappropriate for the extraction target.

Extraction process The system extracts review-tuples from the parsed sentences based on the rules as shown in Figure 4. If a modification relation between a item candidate and a value candidate is *subject-predicate relation*, the system applies rule 1, 2 in sequence. In addition, if the relation is *modifier-head relation*, the system applies rule 3. Finally, if the system cannot found value candidates, applies rule 4. After that, the system adds the *bunsetsus* which include the particular patterns shown in preceding section to the modification relation to which the rules were able to apply.

Post-processing Finally, the system removes functional or stop words and transforms into review-tuples. The system removes the terms (e.g., prefix, particles, etc.) from *item* (*bunsetsus*) except for nouns and removes the terms (e.g., subsidiary verb, prefix, particles, etc.) from *value* (*bunsetsus*) except for nouns, verbs, adjectives, adverbs, prefixes, and suffixes. Additionally, the system eliminates words which have no significant meanings, such as "*koto, mono* (*thing*)", "*omou* (*imagine*)", and "*kangaeru* (*think*)". By eliminating those terms the system can treat some representation as the same and could improve retrieval recall.

Figure 5 shows an example of the extraction processes for review-tuples from "*Nagoya Hotel*" review "*heya-mo hiro-ku, yokuso-mo hiroku, yuttari-to kutsurogeru-node, taihen ki-ni itte-imasu.* (*I was able to relax in comfort in the large room, and the big bathtub was also very fancy.*)" In this example, the system applies transformation rule 2,4 to the dependency tree of the sentence and finally extracts 5 review-tuples.

3.4 Product retrieval

The system calculates each product score which shows the degree of satisfying the request of a user's query and presents those products in descending order of scores. We will describe the way to calculate the product scores based on matching between query-tuples and review-tuples in what follows.

3.4.1 Extraction of query-tuples

The system extracts query-tuples from a natural language query. In this study, we assume that users input natural language queries as the noun phrase which consists of search criteria and *query-object* (e.g., "*heya-ga hiroi yado.* (*the hotel with large room.*)". For example, a query is described as "*heya-ga hiro-ku kutsurogeru yado.* (*the hotel with comfortable, large room.*)". The reason why is that the users often ask about the products which they are looking for in such phrases as "˜ *wo sagashi-te imasu.* (*I am looking for* ˜)", "˜ *ga hoshi.* (*I want to buy* ˜)". Therefore, the system extracts the head of the noun phrase as *query-object* and extracts item-value pairs of query-tuples from the query using the same method

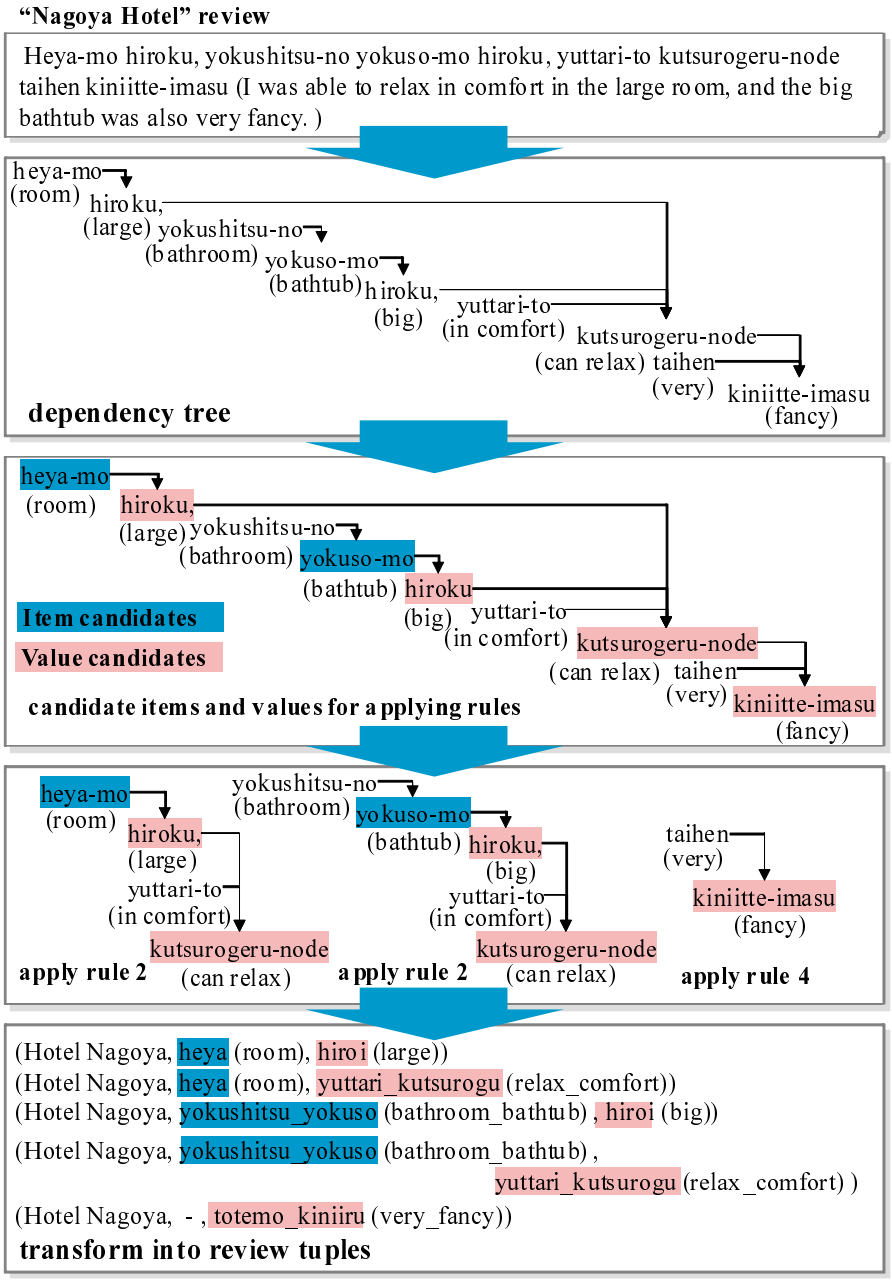

Figure 5 Example of review tuple extraction

as for the above extraction steps. For example, from the above query the system extracts 2 query-tuples "(*yado* (*hotel*), *heya* (*room*), *hiroi* (*large*)), (*yado* (*hotel*), *heya* (*room*), *kutsurogeru* (*comfortable*))".

3.4.2 Correspondence calculation between a review-tuple and a query-tuple

The system calculates the degree of the correspondence to the review-tuples for each query-tuple extracted from a query. The correspondence rate $c\_rate(r, o)$ between query-tuple r and review-tuple o are calculated as follows:

$$c\_rate(r,o) = \begin{cases} \frac{I_{ro}}{I_r} \times \frac{V_{ro}}{V_r} & (item \text{ in } r) \\ \alpha \times \frac{V_{ro}}{V_r} & (\text{no } item \text{ in } r) \end{cases} \tag{1}$$

where I_{ro} is the number of identical morpheme in *item* of query-tuple r to that of review-tuple o, I_r is the number of total morpheme in *item* of query-tuple r, V_{ro} is the number of identical morpheme in *value* of query-tuple r to that of review-tuple o, and V_r is the number of total morpheme in *value* of query-tuple r. Note that it is necessary that the last morpheme in *value* of r is identical to that of o.

The above formula is calculated as product of the fractions of identical morpheme in *item*, *value* respectively. For example, the correspondence rate of query-tuple q (*yado* (*hotel*), *heya* (*room*), *totemo-hiroi* (*very large*)) to review-tuple o (*Nagoya Hotel*, *heya* (*room*), *hiroi* (*large*)) is calculated as:

$$c\_rate(q,o) = 1 \times \frac{1}{2} = \frac{1}{2}$$

Note that if a query-tuple does not include *item*, the fraction of the *item* part is calculated as $\alpha (0 \leq \alpha \leq 1)$. In addition, only if a *review-object* (that is product) of a review-tuple is included in the category of a *query-object* (that is a product category), the above formula is performed.

3.4.3 Calculating product scores

Product score $Score(q, p)$ is calculated based on the degree of fulfilling query-tuples of the query q for review-tuples of product p as follows:

$$\begin{aligned} Score(q,p) &= \sum_{r_i \in R(q)} PF_i \cdot IOF_i \\ PF_i &= \sum_{o_j \in O(p)} pf_j \times c\_rate(r_i, o_j) \\ IOF_i &= \log \frac{ON}{of_i} \end{aligned} \tag{2}$$

where R is a set of query-tuples extracted from query q, $O(p)$ is a set of review-tuples extracted from reviews of product p, pf_j is the frequency of review-tuple o_j in $O(p)$, of_i is the number of *review-objects* which included the review-tuples to which the correspondence rates of query-tuple r_i in query Q is greater than 0, and ON is the total number of *review-objects*.

PF_i and IOF_i show following point of view respectively:

- The more customers describe the same review to a certain product, the more characterized the product is
- The fewer products consumers describe the same review to, the more characterized the product is.

Q#	search query	precision
Q1	*chekku-auto-go-mo nimotsu-wo azuka-tte-moraeru yado* (*the hotel that keeps baggages after checkout*)	10/10
Q2	*shokuji-ga washoku-to yoshoku-de eraberu yado* (*the hotel that features both Japanese and western styles*)	6/10
Q3	*chekku-auto-jikan-ga osoi yado* (*the hotel with a late checkout time*)	8/10
Q4	*furo-ga hiroku-te amenithi-ga jujitsu-shi-te-iru yado* (*the hotel with a large bathtub and many thoughtful amenities*)	9/10
Q5	*heya-no shomei-ga akarui yado* (*the hotel with bright room*)	10/10
Q6	*shuhen-ga kansei-na funniki-de, ochitsui-te-sugoseru yado* (*the hotel in a quiet area where I feel very comfortable*)	8/10
Q7	*konbini-ya resutoran-ga chikaku-te shokuji-ni komaranai yado* (*the hotel located near convenience stores or restaurants and I have no trouble with dining*)	10/10
Q8	*syuccho-ni kaiteki-de, akusesu-ga benri-na yado* (*the hotel comfortable for business trip, and accessible to public transport.*)	10/10
Q9	*ryori-ga oishiku-te, o-furo-ga kashikiri-dekiru yado* (*the hotel that I have delicious dinner and use a reserved a open-air bath*)	7/10
Q10	*beddo-ga hiroku-te, negokochi-ga yoi yado* (*the hotel with wide and comfortable bed*)	10/10
	precision at 10	0.86

Figure 6 Retrieval Results for 10 queries

That means the more review-tuples in a product match the query-tuples in a query and the more they satisfy requests which are difficult to be matched with the product, the higher the score of the product is.

4 Accommodation retrieval system

4.1 Overview of accommodation retrieval system

We developed an accommodation retrieval system named "Yado-tan" in order to show the feasibility of our method. The system uses a large number of reviews and allows users to describe queries as natural language. Figure 7 shows the screen of the retrieval result of the system. On the screen, the system presents accommodations with the summaries (e.g., picture, area, access, features) and reviews related to the query.

We used customer reviews from the review site "words-of-our-consumers" [a] of an accommodation reservation site named "rakuten travel". In our implementation, we registered 700,826 reviews for 20,588 accommodations on the system. We used the KNP (Kurohashi (1998)) as a Japanese dependency parser and set coefficient α in formula (1) to 0.02.

4.2 Retrieval experiment

4.2.1 Experimental methodology

We made 10 queries containing conditions which were difficult to be retrieved by only using the information which the accommodation owner provides, independently of reviews, and judged the validity of the retrieval results. The left side of Figure 6 shows the queries used in the experiment. Those queries are characteristic or high subjective. We evaluated the system by precision at k ($k = 10$). We browsed reviews for the top 10 accommodations of each query and judged suitability of accommodations for the query by whether reviews had descriptions which match the query.

4.3 Experimental results and discussion

The right side of Figure 6 shows the matched number of the top 10 accommodations for each query. Since the precision at 10 for the 10 queries is 86.0%, we confirmed the capability of our method for the queries which traditional systems could not deal with. Especially, the system was able to respond to characteristic or high subjective requests.

In the accommodations which match the queries, reviews which correspond to the request are confirmed as shown below:

query 1 *chekku-out-go-mo, gogo-10-ji-sugi-made azukatte-itadai-ta ue-ni, taihen shinsetsu-ni taio shite itadaki-mashi-ta.*
(*The front desk staff kept my baggage until 10 p.m. after check-out and was courteous.*)

query 4 *heya-ga hiro-ku, amenithi-mo jujitsu-shi-te-ori, kimochiyo-ku shukuhaku deki-mashi-ta.*
(*Since the room was large and I was satisfied amenity service, I was able to stay comfortably.*)

query 5 *heya-no shomei-ga akarui-no-ga tasukari-mashi-ta.*
(*I appreciated that the room was bright.*)

query 7 *konbini-to resutoran-ga chikai-node, sorehodo komaranai-to omoi-masu.*
(*Since some convenience stores and restaurants are close to the hotel, you will have no difficulty to eat.*)

[a] http://travel.rakuten.co.jp/auto/tabimado_bbs_top.html

query 8 *eki-ni-mo chika-ku, toshin-shisetsu-he-no akusesu-ga benri nanode manzoku-shitei-masu. shui-ha ochitsui-te imasu-shi, konbini-ga suuten-ari, resutoran-mo ohku, shuccho-suru-no-ga tanoshimi-na hoteru-no hitotsu-desu.*
(*I was satisfied since the hotel was located near the station and had easy access to the city center. The hotel is in a quiet area and there are some convenience stores and many restaurants near the hotel. So, this is one of hotels that I look forward to staying at on a business trip.*)

Compared with the queries, the reviews have the same expression as the queries. In other words, the system was able to retrieve experimental queries using those reviews.

Meanwhile, the accommodations which do not match the queries are retrieved by following causes:

(1) Since few reviews matched the query, the accommodations which lowly corresponded to the query were retrieved.

(2) The review-tuples which included inappropriate meanings against the query-tuples were accidentally retrieved as highly corresponded.

(3) In the case that more than one conditions were included in the query, only a part of the conditions was fulfilled.

The reason of (1), (2) is that we calculated the degree of the product correspondences to the query, considering no semantic relation. By using such semantic relations as the same or similar meanings between query-tuples and review-tuples, the system might obtain higher recall and can eliminate the review-tuples which included inappropriate meanings against the query-tuples.

The retrieval method based on semantic relations will be feasible for using dimensional compressions such as LSI or random projection, or similarity between *item-value pairs* based on thesaurus or corpus statistics. The latter can include not only synonym but antonym. By including synonymous and antonymous review-tuples and putting together, the system might present the more appropriate products. We will examine the method based on semantic relations.

The reason of (3) is caused by the score calculation in formula (2). Because each product score for the query is calculated from the sum of vectors of query-tuples included in query. It is conceivable that the system preferentially will present the product including more conditions or correspond to query operators. In the latter case, we need to estimate Boolean expressions between the review-tuples from a natural language query.

One of the feature of our soring method is that the accommodations included the large number of reviews tend to be higher on the ranking. The method uses inner product as similarity measure and the higher frequency of review-tuples which match query-tuples is, the higher the score of the product to which the review-tuples refer is.

We consider to use cosine coefficient instead of inner product. In cosine similarity, since each review-tuple is presented as the proportion of total of a product, the products which includes more characteristic features are highly scored. Consequently, inner product preferentially presents the products included more frequent

review-tuples which match given query-tuples. On the other hand, cosine similarity presents the products whose review-tuples which match given query-tuples are more characteristic. We need to discuss which coefficient is more effective

The one more problem is that it is difficult to strictly evaluate the usefulness of the method since only after the users actually use products they can judge whether their needs are satisfied. In addition, since viewpoints of users are diverse and differ from other users, we need to perform quantitative evaluations under a large number of queries and users.

Finally, we describe the availableness of our system in practice. In this method, it is difficult to retrieve areas or price ranges precisely, which are significant for accommodation retrieval. The issue can be solved by combining our system with a traditional system. Recent years, commerce service sites provide various (web service) APIs. For example of accommodation reservation services, those sites provide accommodation search APIs, accommodation information search APIs, vacant room search APIs. Those APIs allow users to retrieve by keyword, present accommodation detail information, and check vacant rooms on designated date. Therefore, combining our system with the APIs, we can develop the more practical system based on the mutual advantages. In addition, the users often check reviews respectively in commerce service or review sites and select the products which match users' requests. So, it is useful for product retrieval system to present the reviews which are relevant to the requests. Our system calculates review scores that are applied to our method and can present relevant products and the supporting reviews by priority to the queries and decrease users' time, costs and works. Consequently, our system is useful for practical applications.

5 Conclusion

We proposed the product retrieval method for corresponding to natural language queries, especially to the diversity, highly subjectivity of search requests. Using a large amount of consumers' reviews, our method can retrieve the products under the search conditions which are not specified by product providers. The system extracted tuples from a query and reviews by using transformation rules based on modification relations between *bunsetsus* and we judged the adequateness of retrieved products for natural language queries in vector model. We developed the accommodation retrieval system based on our method. In the experimental results using natural language query, the retrieval precision of the system reached 86.0% and we confirmed the capability of our system for the product retrieval.

Since we executed matching of review-tuples to query-tuples based on a degree of identity of strings between *item-value pairs*, retrieval recall was lower and mismatched review-tuples were added to product scores. Therefore, we will study the method for retrieving more appropriate products in consideration of semantic relations between review-tuples and query-tuples and similar or opposite review-tuples. In addition, quantitative evaluation by a large number of queries and users and evaluation approach will be our challenges in the future.

Figure 7 Example of retrieval result of accommodation retrieval system "Yado-tan"

Acknowledgments

The authors would like to thank Prof. Toyohide Watanabe of Graduate School of Information Science, Nagoya University for his valuable advice. They also acknowledge the many helpful suggestions of two anonymous reviewers and the participants of the 2007 ICDIM Conference on earlier versions of this paper.

References

Dittenbach, M., Merkl, D., and Berger, H. (2003) 'A Natural Language Query Interface for Tourism Information', *Proceedings of the 10th International Conference on Information Technologies in Tourism (ENTER 2003)*, pp. 152–162.

Chai, J., Horvath, V., Nicolov, N., Stys, M., and Kambhatla, N. (2002) 'Natural Language Assistant: A Dialog System for Online Product Recommendation', AI Magazine, Vol. 23, No. 2, pp. 63-75.

Mcsherry, D. (2005) 'Explanation in Recommender Systems', *Artificial Intelligence Review*, Vol. 24, No. 2, pp. 179–197.

Aciar S., Zhang D., Simoff S., and Debenham J. (2006) 'Recommender System Based on Consumer Product Reviews', *Proceedings of International Conference on Web Intelligence* (*WI* 2006), pp. 719–723.

Hu, M. and Liu, B. (2004) 'Mining Opinion Features in Consumer Reviews', *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004)*, pp. 755–760.

Liu., B., Hu, M., and Cheng, J. (2005) 'Opinion Observer: Analyzing and Comparing Opinions on the Web', *Proceedings of the 14th International World Wide Web Conference (WWW 2005)*, pp. 342–351.

Yi, J., Nasukawa, T., Bunescu, R., and Niblack, W. (2003) 'Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques', *Proceedings of the 3rd IEEE International Conference on Data Mining* (*ICDM 2003*), pp. 427–434.

Popescu, A.M. and Etzioni, O. (2005) 'Extracting product features and opinions from reviews', *The proceedings of the Human Language Technology Conference* (*HLT 2005*) and *The proceedings of the Conference on Empirical Methods in Natural Language Processing* (*EMNLP 2005*), pp. 339–346.

Kobayashi, N. (2006) 'Opinion Mining from Web documents: Extraction and Structurization', Doctoral Dissertation, *Nara Institute of Science and Technology.*

Tateishi, K., Ishiguro, Y., and Fukushima, T. (2004) 'A Reputation Search Engine That Collects People's Opinions Using Information Extraction Technology', *IPSJ Transaction on Databases*, Vol. 45 No. SIG 7 (TOD 22), pp. 115–123.

Kurohashi, S. (1998) 'KNP version 2.0', http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html.