The Determination of Affirmative and Negative Intentions for Indirect Speech Acts by a Recommendation Tree

Takuki Ogawa, Kazuhiro Morita, Masao Fuketa, Jun-ichi Aoe
Dept. of Information Science and Intelligent Systems
University of Tokushima
Tokushima City, Japan

***Abstract*—For context-based recommendation systems, it is necessary to detect affirmative and negative intentions from answers. However, traditional studies cannot determine these intentions from indirect speech acts.**

In order to determine these intentions from indirect speech acts, this paper defines a recommendation tree and proposes an algorithm of deriving intentions of indirect speech acts by the tree. In the proposed method, a recommendation condition (RC) is introduced and it is classified into a required RC, a selectable RC, and a not-selectable RC. The recommendation tree is constructed by nodes and edges corresponding to these three conditions. The deriving algorithm determines affirmative and negative intentions of indirect speech acts by tracing the trees.

From experimental results, it is verified that the accuracy of the proposed method is about 40 points higher than the traditional method.

Keywords*—recommendation system; indirect speech acts; affirmative intention; negative intention

I. INTRODUCTION

As recommendation systems [1] assist users to select commodities, services, and information, it is necessary to elicit users' requirements in conversational contexts. There are many studies of context-based recommendation systems [2-9]. In these systems with proactive recommendations, affirmative and negative intentions of answers are important in order to decide items [5,6,8,9]. For example, items are commodities in e-commerce sites.

In answers with affirmative and negative intentions, there are direct speech acts and indirect speech acts. The direct speech acts represent these intentions by the following two approaches for a recommendation "How about having a cake today?": the first is fixed phrases such as "O.K." and "No, thank you." without "cakes". The second is sentences representing acceptance and rejection intentions such as "I like cakes." and "I don't want to have cakes." with "cakes", respectively.

In the indirect speech acts, there are two patterns for the recommendation: the first is the affirmative answers that select other cakes excluding chocolate cakes such as "I don't want to have chocolate cakes.". The second is the negative answers that select other foods excluding cakes such as "I want to have Japanese noodles".

In order to determine intentions from sentences, there are two methods: the first uses machine learning such as SVM or HMM, the second uses meaning of words and grammars.

For the machine learning methods which classify sentences into intentions or tag-sets with affirmative and negative ones, Fernandez and Picard [10] divided sentences of Spanish CallHome database (telephone conversations) into eight kinds of dialog act tags [11] by SVM. Surendran and Levow [12] classified sentences of HCRC MapTask corpus [13] into twelve kinds of dialog act tags [14] by SVM and HMM. Stolcke et al. [15] proposed methods of a domain-independent framework for tagging the Discourse Annotation and Markup System of Labeling (DAMSL) tag-set [16] to sentences of conversational speeches. Ravi and Kim [17] classified sentences on discussion boards into six speech act categories by N-gram features and linear SVM. Mera, Ichimura, and Yamashita [18] recognized affirmative and negative intentions from answers of questions by the fuzzy theory.

These methods have the advantage that classification models are constructed automatically, but they expend considerable efforts to collect a large learning data.

For the second classifications by meanings of words and grammars, Kitamura, Watanabe, Sekiguchi, and Suzuki [19] estimated negative intentions by combinations of the following five grammatical features: words, auxiliaries, verbs, adjectives, and superordinate concepts of words in previous sentences. Mera [20] and Yoshie et al. [21] calculated affirmative values of sentences by combining these values of words and formulas of grammars (including modality). These values indicate the strength of affirmative intentions and are defined by questionnaires within [0.0-1.0] scales. For example, affirmative values, "Yes" and "No", are defined as 0.94 and 0.06, respectively. Affirmative formulas reflect effects of modalities in affirmative values. The examples of modalities are adverbs such as "very" and "a little", negative modalities, and double negative modalities.

These methods have the merit that their rules have broad utilities for sentences of many domains, but they can not classify answers of indirect speech acts.

In order to determine affirmative and negative intentions from indirect speech acts, this paper defines a new recommendation tree and proposes a new algorithm of deriving intentions of indirect speech acts by the tree. The tree has a root node which indicates a recommendation. The root node has child nodes corresponding to the following three kinds of recommendation conditions (RC):

1) R_RC is the required RC. In a recommendation, "How about having a cake today?", there are R_RCs, "you", "have", "cake", and "today". In RC "cake", there are R_RCs "sweet" and "sweets".

2) S_RC is the selectable RC. In RC "cake", there are S_RCs of kinds of cakes such as "chocolate cake", "short cake", and "Mont Blanc".

3) NS_RC is the non-selectable RC such as "tomorrow", "Japanese noodle" for the recommendation.

The deriving algorithm determines the intention to the root node from the intention to RCs by tracing the trees. From the indirect speech act "I don't like chocolate cakes." for the above recommendation, the algorithm derives the affirmative intention of the root node from the rejection intention of the S_RC "chocolate cake".

Sections 2 and 3 propose the recommendation tree and the algorithm of deriving intentions, respectively. Section 4 evaluates the proposed method by three kinds of open tests. Section 5 concludes the proposed method.

II. A RECOMMENDATION TREE

In this paper, recommendations include four necessary concepts of RC: "WHO", "WHEN", "WHAT", and "VERB". These concepts have RCs related to persons, schedules, objects, and actions, respectively. Table 1 shows examples of RCs of these concepts.

TABLE I. EXAMPLES OF RCS OF FOUR CONCEPTS

Concepts	RCs
WHO	you he
WHEN	today tomorrow
WHAT	cake curry
VERB	go have

Fig. 1 shows a part of the recommendation tree by using RCs in Table 1. In figures of this paper, node labeled by *x* corresponding to string *x*.

In Fig. 1, root node "REC" indicates the recommendation. The root node has four child nodes corresponding to concepts (concept nodes): "WHO", "WHEN", "WHAT", and "VERB". These nodes have child nodes of RCs (RC nodes). For example, concept node "WHO" has RC nodes "you" and "he". There are three kinds of edges (R_edges, S_edges, and NS_edges) for R_RC, S_RC, and NS_RC, respectively. Root node and concept nodes are connected by R_edges as shown by double lines. Concept nodes and RC nodes are connected by S_edges as shown by single lines. These kinds of edges are changed by each recommendation. For the recommendation "How about having a cake today?", the recommendation tree in Fig. 1 is modified (Fig. 2). In Fig. 2, edges of nodes "you", "today", "cake", and "have" are modified to R_edges. Edges of nodes "he", "tomorrow", "curry", and "go" are set to NS_edges as shown by dotted lines.

The recommendation tree can be extended by expanding terminal nodes. Considering an example in Fig. 3, RC node "cake" in Fig. 1 constructs the subtree as the root node. RC node "cake" has RC nodes "taste" and "kind" with R_edge as child nodes. RC node "taste" has RC node "sweet" with R_edge as a child node. RC node "kind" has RC nodes "Mont Blanc", "short cake", and "chocolate cake" with S_edge as child nodes.

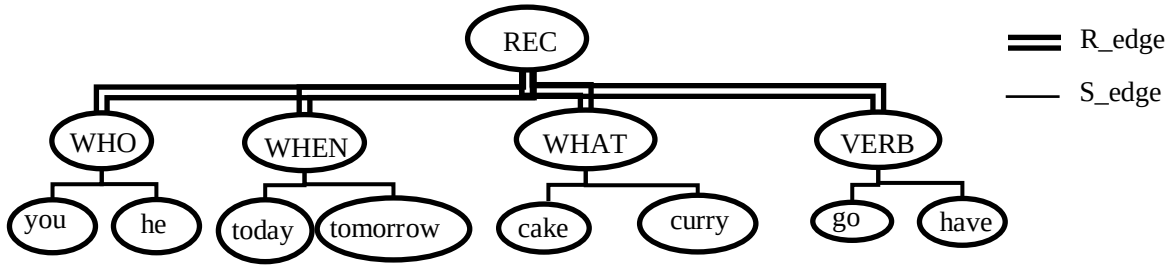

Fig. 1. A part of the recommendation tree

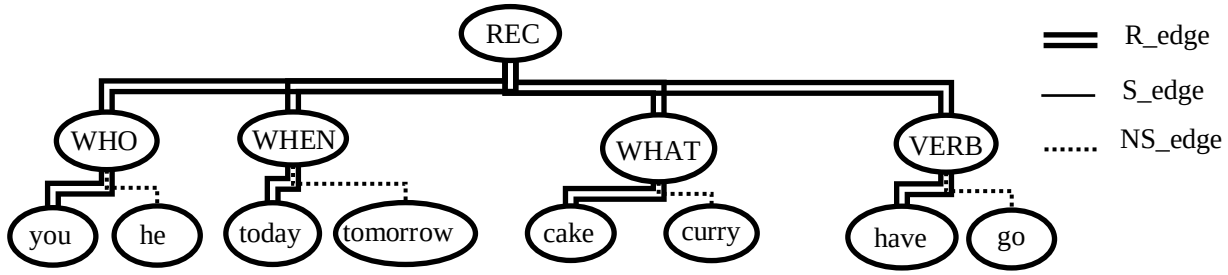

Fig. 2. A part of the recommendation tree of "How about having a cake today?"

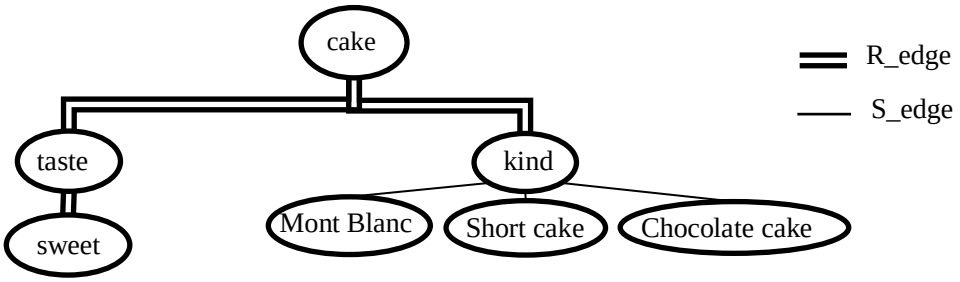

Fig. 3. The subtree of node "cake"

III. AN ALGORITHM OF DERIVING INTENTIONS

An algorithm to be proposed here derives the intention of node "REC" from intentions of RC nodes by tracing a recommendation tree. Suppose that there is the acceptance intention of RC node "curry" for a recommendation "How about having a cake today?" in Fig. 2. Then, the intention of node "REC" is derived to the negative intention. Before proposing the algorithm, the following definitions are prepared.

- Definition

Suppose that NODE[x] is a node for string x. Let EDGE[NODE[x],NODE[y]] be the kind of edges (*NC-edge*, *S_edge*, and *NS_edge*) between NODE[x] and NODE[y]. Let INTENTION[NODE[x]] be the intention of NODE[x] which has one of three kinds intentions: *acceptance*, *rejection*, and *no_information* in this algorithm. *No_information* means that a node doesn't have any intentions. All intentions of nodes are initialized to *no_information*. PARENT(NODE[x]) represents the parent node of NODE[x]. SIBLING(NODE[x]) returns the set of sibling nodes of NODE[x].

- Rejection intentions of PARENT(NODE[x])

INTENTION[PARENT[NODE[x]]] is *rejection* if REJECTION(NODE[x]) is true. It is computed by (1)-(4), where "$\vee$" and "$\wedge$" means logical disjunction and logical conjunction, respectively.

$$\text{REJECTION}(\text{NODE}[x]) = \text{REJECTION1}(\text{NODE}[x]) \vee \text{REJECTION2}(\text{NODE}[x]) \vee \text{REJECTION3}(\text{NODE}[x]) \quad (1)$$

$$\text{REJECTION1}(\text{NODE}[x]) = \exists x(\text{INTENTION}[\text{NODE}[x]] = \textit{rejection} \wedge \text{EDGE}[\text{NODE}[x], \text{PARENT}(\text{NODE}[x])] = \textit{R\_edge}) \quad (2)$$

$$\text{REJECTION2}(\text{NODE}[x]) = \exists x(\forall \text{SIBLING}(\text{NODE}[x])(\text{INTENTION}[\text{SIBLING}(\text{NODE}[x])] = \textit{rejection} \wedge \text{EDGE}[\text{PARENT}(\text{NODE}[x]), \text{SIBLING}(\text{NODE}[x])] = \textit{S\_edge}) \wedge (\text{INTENTION}[\text{NODE}[x]] = \textit{rejection} \wedge \text{EDGE}[\text{PARENT}(\text{NODE}[x]), \text{NODE}[x]] = \textit{S\_edge})) \quad (3)$$

$$\text{REJECTION3}(\text{NODE}[x]) = \exists x(\text{INTENTION}[\text{NODE}[x]] = \textit{acceptance} \wedge \text{EDGE}[\text{NODE}[x], \text{PARENT}(\text{NODE}[x])] = \textit{NS\_edge}) \quad (4)$$

Suppose that users refuse NODE["cake"], and then INTENTION[NODE["cake"]] is *rejection*. In Fig. 2, PARENT(NODE["cake"]) is NODE["WHAT"], and EDGE[NODE["cake"],NODE["WHAT"]] is equal to *R_edge*. For REJECTION 1, INTENTION[NODE["WHAT"]] is *rejection*.

Next, suppose that users refuse NODE[x] for all x such that x is "chocolate cake", "short cake", and "Mont Blanc". Then, INTENTION[NODE[x]] is *rejection*. In Figs. 2 and 3, PARENT(NODE[x]) is NODE["kind"], and EDGE[NODE["kind"],NODE[x]] is *S_edge*. For REJECTION 2, INTENTION[NODE["kind"]] is *rejection*.

Finally, suppose that users accept NODE["curry"], and then INTENTION[NODE["curry"]] is *acceptance*. In Fig. 2, PARENT[NODE["curry"]] is NODE["WHAT"], and

EDGE[NODE[“curry”],NODE[“WHAT”]] is equal to *NS_edge*. For REJECTION 3, INTENTION[NODE[“WHAT”]] is *rejection*.

- Acceptance intentions of PARENT(NODE[x])

INTENTION[PARENT(NODE[x])] is *acceptance* if ACCEPTANCE(NODE[x]) is true. It is computed by (5)-(7).

$$\text{ACCEPTANCE}(\text{NODE}[x]) = \text{ACCEPTANCE1}(\text{NODE}[x]) \vee \text{ACCEPTANCE2}(\text{NODE}[x]) \quad (5)$$

$$\text{ACCEPTANCE1}(\text{NODE}[x]) = \exists x(\text{INTENTION}[\text{NODE}[x]] = \textit{acceptance} \wedge \text{EDGE}[\text{NODE}[x], \text{PARENT}(\text{NODE}[x])] = (\textit{R\_edge} \vee \textit{S\_edge})) \quad (6)$$

$$\text{ACCEPTANCE2}(\text{NODE}[x]) = \exists x(\text{INTENTION}[\text{NODE}[x] = \textit{rejection} \wedge \text{EDGE}[\text{PARENT}(\text{NODE}[x]), \text{NODE}[x]] = \textit{S\_edge} \wedge \exists \text{SIBLING}(\text{NODE}[x]) (\text{INTENTION}[\text{SIBLING}(\text{NODE}[x])] \neq \textit{rejection} \wedge \text{EDGE}[\text{PARENT}(\text{NODE}[x]), \text{SIBLING}(\text{NODE}[x])] = \textit{S\_edge})) \quad (7)$$

Suppose that users accept NODE[“cake”], and then INTENTION[NODE[“cake”]] is *acceptance*. In Fig. 2, PARENT(NODE[“cake”]) is NODE[“WHAT”], and EDGE[NODE[“cake”],NODE[“WHAT”]] is equal to *R_edge*. For ACCEPTANCE 1, INTENTION[NODE[“WHAT”]] is *acceptance*.

Next, suppose that users accept NODE[“chocolate cake”], and then INTENTION[NODE[“chocolate cake”]] is *acceptance*. In Fig. 2, PARENT(NODE[“chocolate cake”]) is NODE[“WHAT”], and EDGE[NODE[“chocolate cake”],NODE[“WHAT”]] is equal to *S_edge*. For ACCEPTANCE 1, INTENTION[NODE[“WHAT”]] is *acceptance*.

Finally, suppose that users reject NODE[“chocolate cake”] and don’t reject NODE[“short cake”]. Then, INTENTION[NODE[“chocolate cake”]] is *rejection*, and INTENTION[NODE[“short cake”]] is *no_information* or *acceptance*. In Fig. 3, PARENT(NODE[x]) for x=”chocolate cake” and x=“short cake” is NODE[“kind”], and EDGE[NODE[“kind”],NODE[x]] is *S_edge*. For ACCEPTANCE 2, INTENTION[NODE[“kind”]] is *acceptance*.

By using above definitions, the proposed algorithm is defined as below.

- An algorithm of deriving intentions

Input: ANSWER_NODE[] and ANSWER_INTENTION[]

ANSWER_NODE[] is a list of strings for nodes accepted or rejected by answers. ANSWER_INTENTION[] is a list of intentions for elements in ANSWER_NODE[]. Indexes of ANSWER_INTENTION[] are elements in ANSWER_NODE[]. For the answer “I like curries”, ANSWER_NODE[] is {“curry”} and ANSWER_INTENTION[“curry”] is {“*acceptance*”}, respectively.

Output: INTENTION[NODE[“REC”]]

Method:

```
for i=1 to n do/*n is the number of elements in ANSWER_NODE[]*/
  INTENTION[NODE[ANSWER_NODE[i]]=ANSWER_INTENTION[ANSWER_NODE[i]]
  target_node = NODE[ANSWER_NODE[i]]
 while target_node ≠ NODE[“REC”] do
    if REJECTION(target_node) is true then
      INTENTION[ PARENT(target_node)] = rejection
    else if ACCEPTANCE(target_node) is true then
      INTENTION[ PARENT(target_node)] = acceptance
    endif
    target_node = PARENT(target_node)]
 endwhile
 if INTENTION[ NODE[“REC”]] is rejection then
    INTENTION[ NODE[“REC”]] = negative
    break
 else if INTENTION[ NODE[“REC”]] is acceptance then
    INTENTION[ NODE[“REC”]] = affirmative
 endif
endfor
```

End of Algorithm

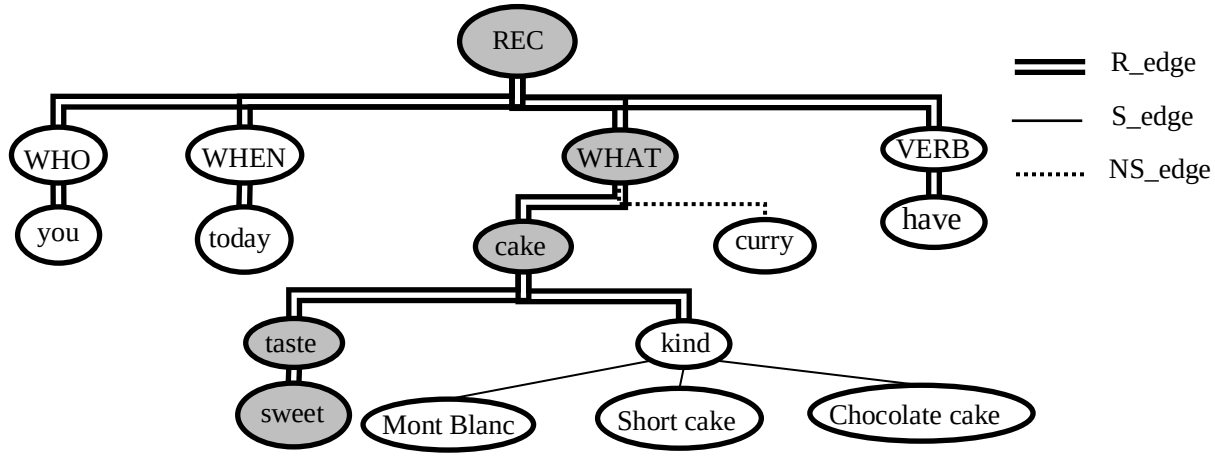

Fig. 4. The derivation process of the answer "I like something sweet."

In case of a recommendation "How about having a cake today?", examples of derivations from answers "I like something sweet." (accepting the node with *R_edge*), "I hate something sweet." (rejecting the node with *R_edge*), "I like curries." (accepting the node with *NS_edge*), and "I dislike short cakes" (rejecting the node with *S_edge*) are as follows:

Example 3.1

For an answer "I like something sweet.", ANSWER_NODE[] is {"sweet"} and ANSWER_INTENTION["sweet"] is {"*acceptance*"}.

By tracing NODE[*x*] for all *x* such that *x* is "sweet", "taste", "cake", "WHAT", and "REC", INTENTION[NODE["REC"]] is become *affirmative*. Fig. 4 shows intentions of nodes. Gray-shaded circles represent nodes with the intention of *acceptance*.

Example 3.2

For an answer "I hate something sweet.", ANSWER_NODE[] is {"sweet"} and ANSWER_INTENTION["sweet"] is {"*rejection*"}.

By tracing NODE[*x*] for all *x* such that *x* is "sweet", "taste", "cake", "WHAT", and "REC", INTENTION[NODE["REC"]] is become *negative*. Fig. 5 shows intentions of nodes. Black-shaded circles represent nodes with the intention of *rejection*.

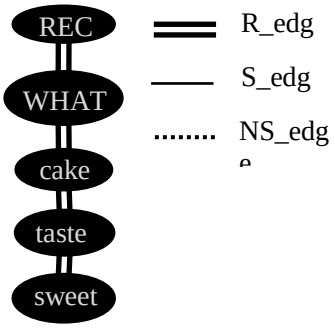

Fig. 5. The derivation process of the answer "I hate something sweet."

Example 3.3

For an answer "I like curries.", ANSWER_NODE[] is {"curry"} and ANSWER_INTENTION["curry"] is {"*acceptance*"}.

By tracing NODE[*x*] for all *x* such that *x* is "curry", "WHAT", and "REC", INTENTION[NODE["REC"]] is become *negative*. Fig. 6 shows intentions of nodes. Gray-shaded circles and black-shaded circles represent nodes with intentions of *acceptance* and *rejection*, respectively.

Example 3.4

For an answer "I dislike short cakes", ANSWER_NODE[] is {"short cake"} and ANSWER_INTENTION["short cake"] is {"*rejection*"}.

By tracing NODE[*x*] for all *x* such that *x* is "short cake", "kind", "cake", "WHAT", and "REC", INTENTION[NODE["REC"]] is become *affirmative*. Fig. 7 shows intentions of nodes. Gray-shaded circles and black-shaded circles represent nodes with intentions of *acceptance* and *rejection*, respectively.

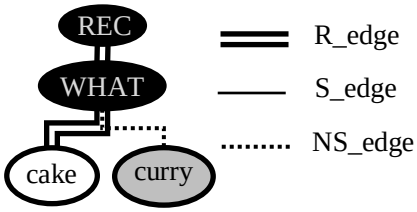

Fig. 6. The derivation process of the answer "I like curries."

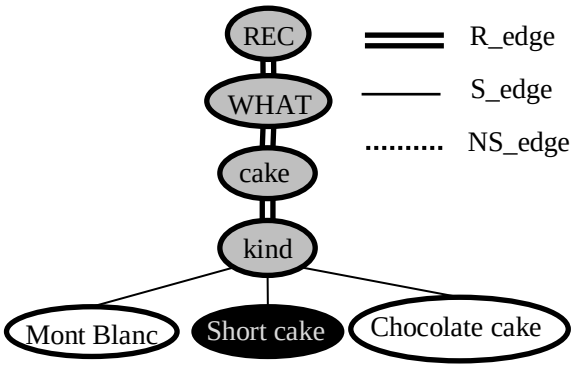

Fig. 7. The derivation process of the answer "I dislike short cakes."

IV. EXPERIMENTS

A. Knowledge for experiments

In daily lives, it is common to recommend foods including cakes and Japanese noodles, and movies. In this experiment, the following three recommendations are assumed, where "Resident Evil" is a title of a movie:

- Recommendation 1: "How about having a cake today?"
- Recommendation 2: "How about having a Japanese noodle today?"
- Recommendation 3: "How about going to the movie, Resident Evil?"

In order to determine intentions from answers to them, a recommendation tree is needed. This experiment constructs the tree from closed corpora which have 500 answers for each recommendation. Answers are collected by four undergraduate students. From corpora, RC nodes and the recommendation tree are defined by discussions with these students. Examples of RC nodes with concept nodes are presented in Table 2. For RC nodes "cake", "Japanese noodle", and "Resident Evil", more detailed descendant nodes are constructed. Total numbers of descendant nodes are 236 nodes. Examples of descendant nodes for each RC node are presented in Table 3. In Table 3, *R_edge* and *S_edge* between a node and a parent node show R and S, respectively.

TABLE II. EXAMPLES OF RC NODES WITH CONCEPT NODES

Concept nodes	RC nodes	Numbers
WHO	You, He, She	7
WHEN	Now, Today, Tomorrow	7
WHAT	Cake, Japanese noodle, Resident Evil	51
VERB	Have, See	10

TABLE III. EXAMPLES OF DESCENDANT NODES OF NODES"CAKE", "JAPANESE NOODLE", AND "RESIDENT EVIL"

Parent	Child	Grandchild
Cake	Genre[R]	Sweets[R], Dessert[R], Confectionery[R]
	Taste[R]	Sweet[R]
	Kind[R]	Short cake[S], Mont Blanc[S], Mille-feuille[S]
	Ingredient[R]	Flour[R], Sugar[R], Egg[R] Butter[S], Apple[S], Strawberry[S], Banana[S]
Japanese noodle	Genre[R]	Noodles[R], Food[R]
	Taste[R]	Spicy[S], Light[S], Salty[S]
	Kind of soup[R]	Miso[S], Soy[S], Salt[S]
	Kind of noodle[R]	Crimp[S], Straight[S], Thin[S], Thick[S]
	Ingredient[R]	Flour[R] Garlic[S], Bean sprouts[S], Onion[S], Sesame seeds[S]
	Size[R]	Large[S], Medium[S], Small[S]
Resident Evil	Screen type[R]	Caption[S], Dub[S], 3D[S]
	Genre of films[R]	Horror[R], Action[R]

(R and S means required and selectable)

B. Knowledge for experiments

In order to evaluate the accuracy of the proposed method, two experiments for closed and open tests are carried out. The closed test uses corpora for constructing the recommendation tree with 500 answers for each recommendation. Open tests uses corpora with 100 answers for each recommendation such as the appendix of this paper. These corpora are collected by ten undergraduate students who don't accumulate closed corpora, and they make ten answers to each recommendation without restriction of responses.

The traditional method proposed by Yoshie et al. [21] is used as a comparative method. Table 4 shows results on the closed test of the proposed method. Tables 5 and 6 show results on open tests for the proposed method and the

TABLE IV. RESULTS ON THE CLOSED TEST OF THE PROPOSED METHOD

	Correct sentences	Total sentences	Correct rates (%)
Recommendation 1	472	509	92.7
Recommendation 2	512	556	92.1
Recommendation 3	476	506	94.1

TABLE V. RESULTS ON THE OPEN TEST OF THE PROPOSED METHOD

	Correct sentences	Total sentences	Correct rates (%)
Recommendation 1	81	100	81.0
Recommendation 2	83	100	83.0
Recommendation 3	84	100	84.0

TABLE VI. RESULTS ON THE OPEN TEST OF THE COMPARATIVE METHOD

	Correct sentences	Total sentences	Correct rates (%)
Recommendation 1	38	100	38.0
Recommendation 2	40	100	40.0
Recommendation 3	34	100	34.0

Comparative methods, respectively. In tables 4, 5, and 6, correct rates mean percentages of correct sentences in total sentences.

From Table 4, it is verified that correct rates of the proposed method becomes high for the closed tests. From Tables 5 and 6, all accuracies of the proposed method are about 40 points higher than the comparative method in open tests of recommendations 1, 2, and 3.

In the open test, problems of the proposed method are misclassifications of complex sentences and the lack of knowledge.

- Misclassifications of complex sentences

The example of the complex sentence is “I’ve had enough, therefore I choose a small dish.” which has two sentences, “I’ve had enough” and “I choose a small dish”. The intention of the first sentence is negative because the sentence rejects R_RC,”have”. The intention of the second sentence is affirmative because the sentence accepts S_RC, “small”. The intention of the sentence is affirmative. However, the proposed method produces a negative intention because the negative intention is prior to the affirmative one.

The way to solve the problem is to consider conjunctions and give priority to the intention estimated from a backward-sentence.

- The lack of knowledge

A part of misclassifications of the proposed method are occurred in sentences which include non-defined nodes. For example, the proposed method misclassifies the sentence, “I have to go to a piano lesson.”, because there is no node “piano lesson” in the child nodes of node “WHAT”. This problem can be solved by introducing knowledge of daily lives.

V. CONCLUSIONS

This paper has proposed a method of determining affirmative and negative intentions from indirect speech acts. In the proposed method, a recommendation tree has been defined and an algorithm of deriving intentions of indirect speech acts by the tree is proposed.

The tree consists of nodes and edges corresponding to the three kinds of RC: R_RC, S_RC, and NS_RC. The root node indicates the recommendation and has four concept nodes as child nodes. Concept nodes have RC nodes as child nodes. The deriving algorithm determines affirmative and negative intentions of indirect speech acts by tracing the trees.

From experimental results for three kinds of open tests, all accuracies of the proposed method are about 40 points higher than the traditional method.

VI. FUTURE WORK

Future works are to improve misclassifications of complex sentences of an acceptance sentence and a rejection sentence, and to construct invitational knowledge of daily lives.

REFERENCES

[1] A. Levi, O. Monkryn, C. Diot, and N. Traft, "Finding a needle in a haystack of reviews: cold start context-based hotel recommender system demo." Proceedings of the sixth ACM conference on Recommender systems. ACM, 2012.

[2] P. Lops, G. Marco, and S. Giovanni, "Content-based recommender systems: State of the art and trends." Recommender Systems Handbook. Springer US, 73-105, 2011.

[3] B. J. Han, S. Rho, S. Jun, and E. Hwang, "Music emotion classification and context-based music recommendation." Multimedia Tools and Applications 47.3, 433-460, 2010.

[4] P. Johansson, “Natural language interaction in personalized EPGs”, InProc. of Workshop notes from the 3rd International Workshop on Personalization of Future TV, Johnstown, Pennsylvania, USA , 27-31, 2003.

[5] P. Johansson, “Madfilm-a multimodal approach to handle search and organization in a movie recommendation system”, In Proceedings of the 1st Nordic Symposium on Multimodal Communication , 53-65, 2003.

[6] T. Misu, and T. Kawahara, “Speech-based interactive information guidance system using question-answering technique”, In Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. , 4, 145-148.

[7] H. Shimazu, “ExpertClerk: navigating shoppers' buying process with the combination of asking and proposing”, In Proceedings of the 17th international joint conference on Artificial intelligence, 2, 1443-1448, 2001.

[8] H. Shimazu, “ExpertClerk: A Conversational Case-Based Reasoning Tool forDeveloping Salesclerk Agents in E-Commerce Webshops”, Artificial Intelligence Review, 18(3-4), 223-244, 2002.

[9] C. A. Thompson, M. H. Goeker, and P. Langley, “A personalized system for conversational recommendations”, J. Artif. Intell. Res. (JAIR), 21, 393-428, 2004.

[10] R. Fernandez, and R. W. Picard, “Dialog act classification from prosodic features using support vector machines”, In Speech Prosody 2002, International Conference, 291-294.

[11] L. Levin, A. Thymé-Gobbel, A. Lavie, K. Ries and K. Zechner, “A discourse coding scheme for conversational Spanish”, In International Conference on Speech and Language Processing, 1998.

[12] D. Surendran, and G. A. Levow, “Dialog act tagging with support vector machines and hidden Markov models”, In Proceedings of Interspeech , 2006, 1950-1953.

[13] A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard. J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. S. Thompson, and R. Weinert, "The HCRC map task corpus", Language and Speech,34(4), 351-366, 1991.

[14] J. Carletta, S. Isard, G. Doherty-Sneddon, A. Isard, J. C. Kowtko, and A. H. Anderson, "The reliability of a dialogue structure coding scheme", Computational Linguistics, 23(1), 13-31, 1997.

[15] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech", Computational linguistics, 26(3), 339-373, 2000.

[16] M. Core, and J. Allen, "Coding dialogs with the DAMSL annotation scheme", In AAAI fall symposium on communicative action in humans and machines, 28-35, 1997.

[17] S. Ravi, and J. Kim, "Profiling student interactions in threaded discussions with speech act classifiers", Frontiers in Artificial Intelligence and Applications, 357-364, 2007.

[18] K. Mera, T. Ichimura, and T. Yamashita, "Analysis of User Communicative Intention from Affirmative/Negative Elements by Fuzzy Reasoning and Its Application to WWW-based Health Service System for Elderly", In Proc. of the 6th Intl. Conf. on Soft Computing (IIZUKA2000) , 971-976.

[19] J. Kitamura, Y. Watanabe, Y. Sekiguchi, and Y. Suzuki, "An Extraction and Processing Method of User's Denial Utterance for a Speech Dialog Device", Transactions of Information Processing Society of Japan, 46(7), 1789-1796, 2005. (in Japanese)

[20] K. Mera, "Analyzing affirmative/negative intention from plural sentences", Proc. KES'01, 1222-1227.

[21] M. Yoshie, K. Mera, T. Ichimura, T. Yamashita, T. Aizawa, and K. Yoshida, "Analysis of affirmative/negative intentions of the answers to yes-no questions and its application to a web-based interface", Journal of Japan Society for Fuzzy Theory and Systems, 14(4), 393-403, 2002.

Appendix

Examples of answers in Open corpora are shown as follows:

[Answers to recommendation 1, "How about having a cake today?"]

Answers	Intention
Let's go now.	Affirmative
I can't get enough of sweet food.	Affirmative
I have a Mont Blanc.	Affirmative
I dislike short cakes.	Affirmative
Oh goody!	Affirmative
My tummy is full.	Negative
Maybe another time.	Negative
I don't like cakes.	Negative
I can't have it because I have a piano lesson now.	Negative
I don't have it because I have egg allergies.	Negative

[Answers to recommendation 2, "How about having a Japanese noodle today?"]

Answers	Intention
Yes, I want to have it.	Affirmative
Let's go to a low price Japanese noodle shop.	Affirmative
I want to have 3 bowls.	Affirmative
I want to have a miso-flavored noodle.	Affirmative
O.K.	Affirmative
I got tired of it.	Negative
Shall we have other foods?	Negative
I can't have it because I feel bad.	Negative
I don't go to have it.	Negative
I can't have noodles.	Negative

[Answers to recommendation 2, "How about going to the movie, Resident Evil?"]

Answers	Intention
I want to watch it with 3D scenography.	Affirmative
I like action movies.	Affirmative
It seems pleasant.	Affirmative
I want to go if the ticket price is discounted.	Affirmative
I like movies very much.	Affirmative
I want to watch animation movies.	Negative
I don't want to watch it.	Negative
I watch it by a rental video.	Negative
I dislike something terrible.	Negative
I'm not interested it.	Negative