Text Mining Agent for Net Auction

Yukitaka Kusumura, Yoshinori Hijikata and Shogo Nishida
Graduate School of Engineering Science Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, JAPAN
kusumura@nishilab.sys.es.osaka-u.ac.jp
hijikata@sys.es.osaka-u.ac.jp
nishida@sys.es.osaka-u.ac.jp

ABSTRACT

Net auctions have been widely utilized with the recent development of the Internet. However, it is a problem that there are too many items for bidders to select the most suitable one. We aim at supporting the bidders on net auctions by automatically generating a table which contains the features of several items for comparison. We construct a system called NTM-Agent(Net auction Text Mining Agent). The system collects Web pages of items and extracts the items' features from the pages. After that, it generates a table which contains the extracted features. This research focuses on two problems in the process. The first problem is that if the system collects items automatically, the results contain the items which is different from the items of the user's target. The second problem is that the descriptions in net auctions are not uniform (There are different formats such as sentences, items and tables. The subjects of some sentences are omitted.). Therefore, it is difficult to extract the information from the descriptions by conventional methods of information extraction. This research proposes methods to solve the problems. For the first problem, NTM-Agent filters the items by correlation rules about the keywords in the titles and the item descriptions. These rules are created semi-automatically by a support tool. For the second problem, NTM-Agent extracts the information by distinguishing the formats. It also learns the feature values from plain examples for the future extraction.

Keywords

text mining, information extraction, net auction

1. INTRODUCTION

Net auctions have become one of the most popular applications on the Internet. Usually, in net auctions, bidders (after here users) search items using keywords or categories and read the items' descriptions (after here *item descriptions*) written by the sellers. But when the number of the items becomes larger, this work becomes more troublesome for the users. For this problem, this paper introduces a system called *NTM-Agent*[1] (Net auction Text Mining Agent). NTM-Agent collects the Web pages of the items satisfied for the user's search demand, extracts the features of the items from the pages and makes a table containing these features. With this table, the user can compare the items at a glance.

We aim at the pratical system in the system design of NTM-Agent. In the first, NTM-Agent automatically searches the Web pages from real auction sites. For realizing information extraction with high precision, we use *domain knowledge* [1] about the items for every category. Concretely, the content of our domain knowledge is the keywords which express the characteristics of items in a category (after here *feature names* [2]). The system extracts the *feature value* corresponding to the feature name from the item descriptions. However, when the system automatically collects the Web pages of the items from net auction sites and automatically extracts information from the collected pages, there are the following problems:

Problem 1 The classification of items isn't uniform.

> In net auctions, a seller freely gives a title to his/her item and freely assigns a category to the item. Sometimes, the item is titled or categorized incorrectly. Therefore, when the system searches items using keywords or categories, the search results contain *noise items* [3] which are different from the items of the user's target (after here *target items*).

Problem 2 Item descriptions aren't uniform.

> A seller freely describes the item description. In some item descriptions, feature names aren't described (One reason of this is the characteristic of Japanese that a subject is often omitted.). It is impossible to search the feature value using the keyword of the missing feature name. And there are different formats in item descriptions in their layouts such as tables, items and sentences. It is difficult to extract information precisely by a constant method.

Our solutions for these problems are as follows:

[1]the knowledge about the field given to a system in advance
[2]such as "Processor", "Memory" and "HDD", in the case that the category is personal computers
[3]the peripherals such as a keyboard, memory card and hard disk drive in the case that a user wants a personal computer

SAC' 04, March 14-17, 2004, Nicosia, Cyprus

Solution 1 NTM-Agent filters the noise items by correlation rules about the keywords in the titles and the item descriptions. The creation of the correlation rules is supported by a tool using a market basket analysis.

Solution 2 To extract the feature values from the item descriptions in which the feature names aren't described, NTM-Agent learns the pair of the feature name and the feature value, when it extracts the features from the item descriptions which have a simple correspondence between the feature name and the feature value. After learning the pairs of the feature name and the feature value, if the feature names aren't described in the item description, NTM-Agent searches the feature values learned before. To extract the feature value from the item descriptions with mixed format in their layouts, NTM-Agent distinguishes the format type from tables, items and sentences, and extracts the feature values in the most suitable way.

This paper is organized as follows. Section 2 introduces some related works and compares this research with them. Section 3 outlines the process of NTM-Agent, shows the domain knowledge used in the system. Section 4 and Section 5 explain the filtering method of noise items and the extracting method of items' features. Section 6 evaluates NTM-Agent. Finally, Section 7 offers some conclusions.

2. RELATED WORK

Our research is related to the field of agent for electronic commerce, information extraction from the Web[2] and natural language processing. In this section, we describe the difference between the researches of each field and ours.

In the field of agent for electronic commerce, some agent systems were developed to help users to find their target item from a large number of items. Biddingbot[3] and Shopbot[4] are research prototypes which summarize the information of items on electronic commerce sites. Biddingbot sends the request from a user to some auction sites and summarizes the items about only the items' prices. Shopbot searches items in online shops and summarizes the characteristics of the items. Differently from Biddingbot, NTM-Agent extracts not only the price of the item but also the information about the feature (performance, condition and so on) of item. Shopbot automatically extracts the features of the items using the domain knowledge which contains the examples of feature values. Shopbot does keyword matching between the examples of feature values and the text of the page and learns where the feature values are generally described in the page. However this method can deal with only the page that has a constant description and a constant format like the pages in online shops. While, NTM-Agent can extract information from any pages with un-uniform descriptions and formats. Furthermore while other systems do not filter noise items, NTM-Agent filters noise items. This function is crucial for the electronic commerce sites where miscellaneous sellers participate in.

In the field of information extraction from the Web, many researches were also conducted to acquire computer-readable information from semistructured documents. They learn templates to extract information from a Web page by analyzing the structure of HTML tags. The templates are called a wrapper. The main research subject is the generation of templates by learning given sites(Knoblock[5], Freitag[6] and Chen[7]). However, the templates can be used only for the site learned in advance. Furthermore while the systems can deal with highly structured HTML only. NTM-Agent deals with not only structured text but also unstructured text like sentences.

In the field of natural language processing, many researches provide methods for extracting information from free text. MUC(Message Understanding Conference)[8] is the conference of information extraction. MUC applies a common task, which is information extraction from newspapers. The methods proposed in MUC learn the patterns of phrase and structure in a sentence or a document from a large size of training data tagged by human. Therefore the methods need the training data. Furthermore, because newspapers are written by professional writers, the text is organized grammatically and lexically. NTM-Agent extracts information from the texts which are described by a large number of amateur authors in net auctions. Text of the item description in net auctions misses some of feature names and has mixed formats of descriptions. It is the originality of our study that NTM-Agent deals with those un-uniform texts written by many amateur authors.

3. NTM-AGENT

Section 3.1 presents the process flow in NTM-Agent. Section 3.2 describes the domain knowledge used in NTM-Agent.

3.1 Process flow in NTM-Agent

We implemented NTM-Agent as a JAVA servlet. The users can use NTM-Agent on their own browser. Figure 1 shows the system structure. The following describes the process in the system:

1: The user inputs search keywords about the item he/she wants (or a URL of the search result page in an auction site) and selects the item's category (This means that the user selects the domain knowledge for extraction) on their browser.

2: The *search module* accesses the auction site with the *domain knowledge for search*, and gets the search result pages.

3: The titles and the URLs of the items are extracted from the search result pages. The titles are checked with the *filtering rules*, and the noise items are removed. The page of each item is collected by the item's URL. The filtering rules is described in Section 4.

4: Each item description is taken out from the item's page by using templates (They are in the domain knowledge for search.). The item descriptions of the items are checked with the filtering rules, and the noise items are removed. The item descriptions of the target items are sent to the *extraction module*. The extraction module is described in Section 5.

5: The feature values of each item are extracted by using the domain knowledge for extraction. The feature values are registered to the feature database and stored as XML files.

6: Clicking the button "NEXT", the user downloads the table created from the XML files. An example of the

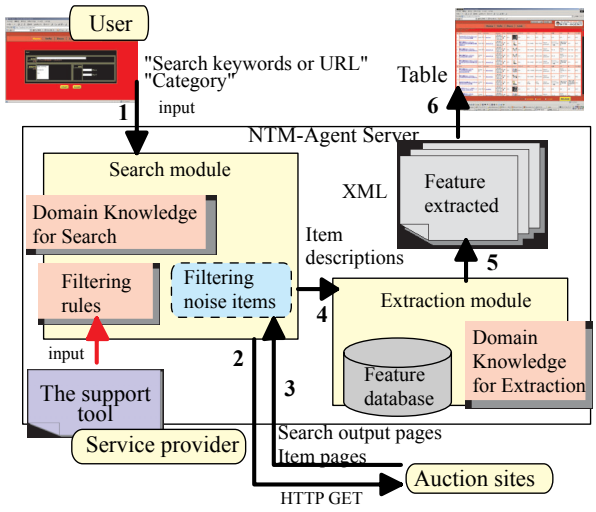

Figure 1: The system structure of NTM-Agent

table is shown in Figure 2. The feature names are in the first line of the table. The feature values are in the second line and after of the table. The first row of the table shows the titles of items. "N/A" in some cells means that the feature value isn't written in the item description or the feature value can't be found by NTM-Agent.

Because all of the communication is conducted by HTTP, the communication message goes through firewalls. Therefore the user can use any auction sites, as long as the domain knowledge for search is prepared for those sites.

3.2 Domain knowledge

NTM-Agent uses the following domain knowledge:

Domain knowledge for search The link structures of auction sites, the templates for the search result pages and the templates for the item's pages

Domain knowledge for extraction Keywords (feature names) representing the characteristics of the items in each category

The domain knowledge for search is used for searching the items' pages in the auction sites and extracting some parts of the document (title, price, item description and bidding deadline) from the page. This is created for each auction site. Figure 3 shows an example of the template for the item's pages in Yahoo Auction. The left keyword of ':' is the name of the part to be extracted. The text after ':' between `<START>` and `</START>` or between `<STOP>` and `</STOP>` is a clue to identify the part to be extracted. This is a raw HTML text which exists before or after the part to be extracted. In Figure 3, `ITEM_DESCRIPTION` is declared. The system will search the text between `<P><P><P>` and `</TBODY></TABLE>` in the page. The discovered text will be extracted as the item description.

```
ITEM_DESCRIPTION:
<START><P><P><P>
</START>
<STOP></TBODY></TABLE>
</STOP>
```

Figure 3: Template for item's pages

The domain knowledge for extraction is feature names. This is a clue to search where the feature value is in the item description. This is created for each category of items. The feature to be extracted is described with the synonyms of the feature name. The location of the feature name will be identified by searching the keywords of the synonyms.

4. FILTERING OF NOISE ITEMS

To remove noise items, we use the keywords which have the correlation with target items (Keyword set A) and the keywords which have the correlation with noise items (Keyword set B). Namely, NTM-Agent uses these keywords as the following two types of rules:

A rule for target items: IF a keyword in Keyword set A is contained in text, THEN the item is a target item.

A rule for noise items: IF a keyword in Keyword set B is contained in text, THEN the item is a noise item.

NTM-Agent checks titles and item descriptions with the rules. NTM-Agent uses only either rules for target items or rules for noise items. The filtering feature with each type of rules is as follows:

The filtering feature with rules for target items: By using the rules for target items, the target items are distinguished. NTM-Agent removes the items except the target items boldly. While some target items may be removed incorrectly, many noise items can be removed.

The filtering feature with rules for noise items: By using the rules for noise items, the noise items are distinguished. NTM-Agent removes only the noise items carefully. While only a few noise items may be removed, target items are hardly removed.

The selection of the type of rules depends on the number of items in the search result. Namely, if the number of items in the search result is above a threshold α, NTM-Agent uses the rules for target items. If the number of items is below α, NTM-Agent uses the rules for noise items. This switch of the two types of rules (after here, the *switching mechanism*) is used by the following reasons. In the case that the number of items is small, the number of the noise items is also small. The noise items are not a problem for a user so much. NTM-Agent should show all target items even if some noise items remain. On the other hand, in the case that the number of items is big, the noise items will be a problem. NTM-Agent should remove the noise items even if some target items are removed. In our implemented system in Section 3, the threshold α is set to 100.

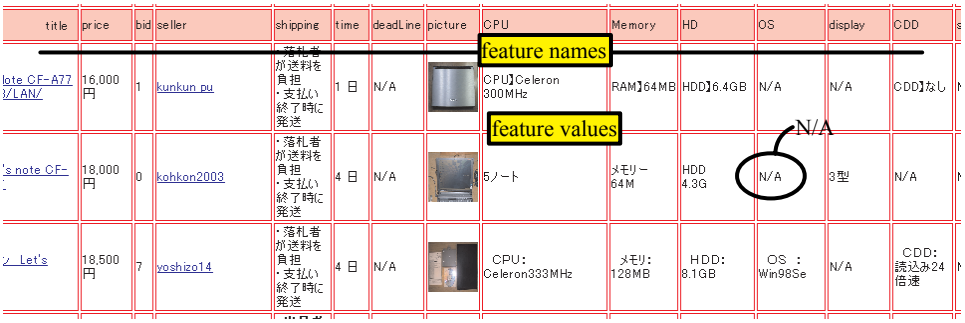

Figure 2: The output of NTM-Agent

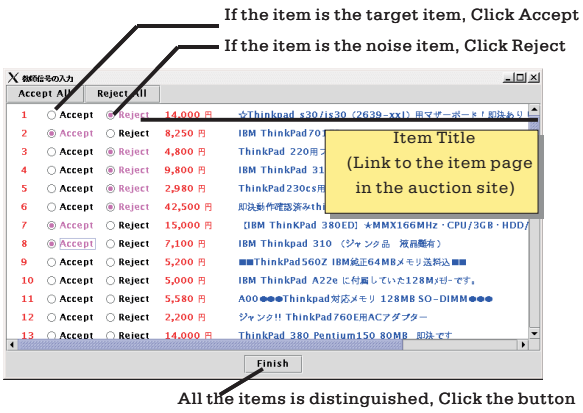

Figure 4: Interface to input teacher signal

Our filtering method needs the rules on keywords for each category of items. These rules must be prepared by human. However it is difficult for human to find the keywords that have correlation with the item's type (a target item or a noise item). For this problem, we developed a tool which supports creating the rules. The tool is for service providers who serve the tables of items' features for the users on the Web, or for heavy users who want to customize the default rule. These people is called *users* in this Section 4. We make this tool as a general one which does not depend on a specific item category. Therefore we use a market basket analysis[9] which is the most fundamental algorithm for extracting correlation rules.

In our tool, when the user searches the items of the category which the user wants to create correlation rules, the system acquires the titles and the item descriptions in the search result and displays an interface to input teacher signals (See Figure 4). The user inputs whether the user accepts or rejects the item (judges whether the item is a target item or a noise item) on the interface. The system conducts a market basket analysis with the teacher signals inputted by the user and learns the keywords of titles (and item descriptions) which collocate with the teacher signals. The system displays those keywords and whether they collocate with the target item or the noise item on the window as a rule. After the user's checking or editing the rules on the interface, the rules are stored in the NTM-Server.

Table type
in <TABLE>tag
<TR> ~a feature name~<TD>~○ ○ ○ ~

Items type

<LI> ~a feature name~ ○ ○ ○ ~
or
~a feature name~ ○ ○ ○ ~ "
 or LF"

Sentences type
the others

Figure 5: Definitions of tables, items and sentences

5. EXTRACTION OF FEATURE VALUES

Section 5.1 describes the extracting methods of items' features. Section 5.2 describes the learning methods of the feature name and the feature value.

5.1 Extraction from each format of item descriptions

We defined three types of the formats in item descriptions as Figure 5. NTM-Agent distinguishes the format type from tables, items and sentences, and extracts the feature values in the following way. Text is divided into text parts in every signal to pause(`<TR>` tags for table, `<LI>` tags and `<BR>` tags for items and some punctuation marks for sentences). The text part which contains the feature name is taken out. After that a morphological analysis[10] is done on the text part, the numerical description, the proper noun [4] and the words near the feature name are extracted preferentially. And for sentences, when a noun doesn't exist, the predicate is extracted. This is because the predicate shows the characteristics of the item well next to the noun.

5.2 Learning method of the feature name and the feature value

When the feature name is missing in the item description, it is impossible to search the text part that contains the feature value to the missing feature name. To cope with this problem, our system learns the pair of the feature name and the feature value from the item descriptions that is easy to identify the pair. And when the feature name is missing, the system searches the feature value learned before and extracts it if the system finds.

Figure 6 shows the process of learning method. When the

[4]It is inferred with the dictionary in the system of morphological analysis

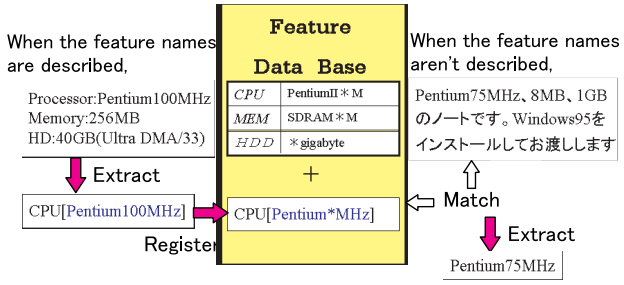

Figure 6: Learning the feature value

feature names are described in the item description, the feature values are extracted in the method described in Section 5.1. After extracting the feature values, they are stored in a database with their feature names. The database is called the *feature database*. Note that, the system translates numerical characters to the sign '*' to deal with any numerical data. And when the feature names are missing in the item description, the system refers to the feature database and acquires the keyword of the feature value. Then it searches the keyboard in the item description and extracts it when it finds the keyword.

6. EVALUATION

We evaluated NTM-Agent by the following ways:

- The simulation experiment of the filtering method
- The simulation experiment of the extracting method
- The user experiment of NTM-Agent

The first one and the second one are to see the performance of the filtering method and the extracting method. The third one is to evaluate the entire system by seeing the users' performance of a task. Section 6.1, Section 6.2 and Section 6.3 describes above three evaluations respectively.

6.1 Simulation experiment of filtering method

We verify the validity of filtering method and the validity of switching mechanism by the filtering precision and the filtering recall. The equations to calculate the two parameters are as follows:

1. Filtering precision $= \mid B \mid / \mid A \mid$
2. Filtering recall $= \mid B \mid / \mid C \mid$

A , B and C in the above equations have the following meanings:

A **:** Set of the items in the output of filtering

B **:** Set of the target items which are included in Set A

C **:** Set of the target items in the input of filtering

We calculated the above parameters of our filtering method. For comparison, we also calculated the above parameters of random filtering. The filtering precision of random filtering is the ratio of the target items in relation to all the items in the search result. The filtering recall of random filtering is calculated as that random filtering acquires the same number of items as the number of the items which NTM-Agent acquired. Our filtering method switches the correlation rules by the number of items in a search result. Therefore, we conduct two kinds of experiment. The first one assumes the case that the number of items is small (about 30 items). The second one assumes the case that the number of items is large (about 100 items). Table 1 shows the condition of the experiment.

Figure 7-(a) shows the filtering precision and the filtering recall when the number of the search result is small. Figure 7-(b) shows the filtering precision and the filtering recall when the number of the search result is large. The precision and the recall of baby clothing are not much different between NTM-Agent and random filtering. This reason is that the number of the noise items in baby clothing is small(We calculated the ratios of noise items in three categories. They are 65% in computer, 53% in car and 8% in baby clothing). Thus, the effective rules couldn't be learned because there were a few noise items in the teacher signals in baby clothing. Our filtering method isn't valid for the category which contains a few noise items. However, the precision of random filtering of baby clothing is much higher than that of the other categories. This means that there are many target items and a few noise items in the search result of baby clothing. We consider that this fault of our filtering method is not such a serious problem in the categories which don't contain so many noise items like baby clothing.

Next, we discuss the result of computer and car. The filtering precision and recall of NTM-Agent are higher than those of random filtering. Therefore we can say that our filtering method is effective for the category which includes noise items to some extent. We compare Figure 7-(a) and Figure 7-(b) to see whether the switching mechanism is valid or not. When we see the result in Figure 7-(a), the filtering precision of NTM-Agent is not much higher than that of random filtering while the filtering recall of NTM-Agent is 100%. When the number of items in the search result is small, the user does not mind that a few noise items are included if all target items are contained in the table. On the other hand, when we see the result in Figure 7-(b), the filtering precision of NTM-Agent is much higher than that of random filtering while the filtering recall of NTM-Agent is not 100%. When the number of items in the search result is large, the user wants to check target items on the table with high precision even if some target items are removed. From this, we can say that our switching mechanism is valid to ease the user's workload.

6.2 Simulation experiment of extracting method

In this section, we describe three evaluations. The first one is to see the performance of our extracting method. The second one is to see whether or not the feature database has effectiveness in extracting feature values. These evaluations were conducted by seeing the average of evaluation parameters of all features. However, the difficulty of extraction varies by each feature. Therefore, in the third evaluation, we investigate the efficiency of extraction for every feature. This result will be helpful for considering the improvement of our extracting method. Section 6.2.1 describes the performance of our extracting method. Section 6.2.2 describes the evaluation of the feature database. And Section 6.2.3 compares the efficiency of extraction among features.

Table 1: The condition in the experiment for filtering

Category of items	Computer	Car	Baby clothing
Search keyword	Gateway	Corolla	one-pieces
Number of items	30 and 100 items	30 and 100 items	30 and 100 items
Filtering rules	20 rules learned from 100 items	7 rules learned from 100 items	8 rules learned from 100 items

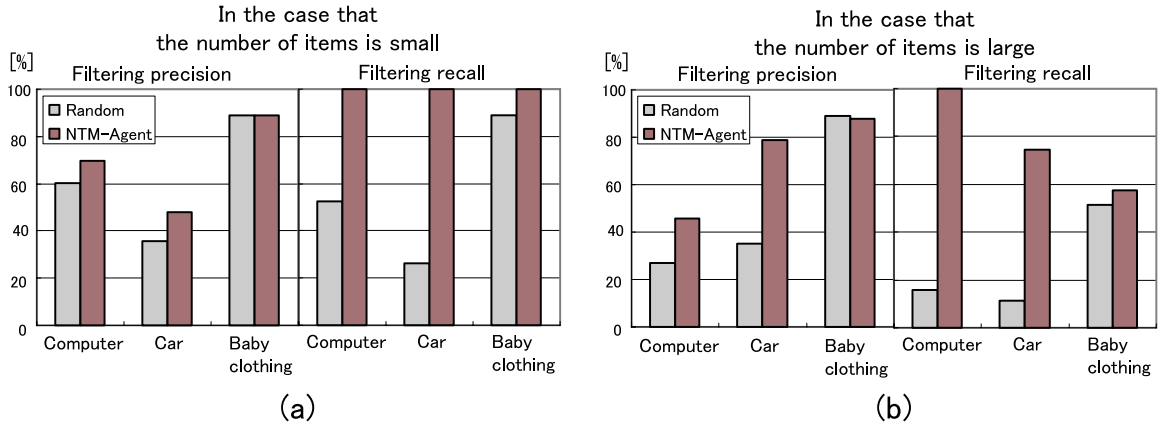

Figure 7: The precision and the recall of filtering

Table 3: The precision of extraction

	Extracting precision [%]	Extracting recall[%]
Computer	60.3	61.9
Car	71.3	52.2
Baby clothing	56.6	54.2
Average	62.7	56.1

6.2.1 *Efficiency of extraction*

To evaluate our extracting method, we calculated the extracting precision and the extracting recall. The equations of the two parameters are as follows:

1. Extracting precision = $| B | / | A |$
2. Extracting recall = $| B | / | C |$

A , B and C in the above equations have the following meanings:

A **:** Set of the all feature values extracted by NTM-Agent.

B **:** Set of the correct feature values extracted by NTM-Agent.

C **:** Set of the all feature values in all item descriptions used in the experiment.

Table 2 shows the condition of the experiment. Table 3 shows the extracting precision and the extracting recall. The extracting precision and the extracting recall are about 60%. We are not sure whether or not this information extraction is effective at this precision and recall. Therefore we conducted user experiment in Section 6.3.

6.2.2 *Evaluation of the feature database*

To evaluate the effectiveness of the feature database, we compare the extracting precision and the extracting recall between the extraction with the feature database and the extraction without the feature database. The extraction without the feature database was carried out under the same condition as the condition of the experiment in Section 6.2.1. After that, we calculated the improvement rate of the extracting precision and the improvement rate of the extracting recall.

The equation of the improvement rate is as follows:

1. Improvement rate = $| B | / | A |$

A and B in the above equations have the following meanings:

A Extracting precision (or extracting recall) of the extraction without the feature database.

B Extracting precision (or extracting recall) of the extraction with the feature database.

Table 4 shows the improvement rate of the feature database. The average of the improvement rate for extracting recall is about 1.1. This means that NTM-Agent can extract extra 10% feature values by the feature database. Therefore, we considered the feature database is effective.

6.2.3 *Comparison of the extraction among features*

Table 5 shows the extracting precision and the extracting recall of each feature in three categories. We can see that the extracting precision and the extracting recall of "CPU", "Mileage" and "Size" are high. The reason is that NTM-Agent extracts numerical values and proper nouns preferentially. These features are always described in numerical values and proper nouns. The extracting precision and the extracting recall of some features (CD drive, Scratch, Dot

Table 2: The condition of the experiment of extraction

Category	Computer	Car	Baby clothing
Search keyword	Vaio	Stepwagon	baby clothing
Feature values in the item descriptions	118 values	138 values	142 values
Features in the domain knowledge	8 features	7 features	7 features
Feature names in the domain knowledge	40 keywords	34 keywords	26 keywords
Learning of the feature database	from 1000 items	from 1000 items	from 1000 items

Table 4: The improvement rate of the feature database

	Extracting precision	Extracting recall
Computer	1.07	1.26
Car	1.02	1.03
Baby clothing	1.04	1.05
Average	1.04	1.11

disability and Color) are low. We think there are two reasons.

First reason is that when the feature names are hardly described, the feature values are also hardly learned and extracted. "Color" is an example of such features. The descriptions about "color" are like "This is black." or "This is a white cover of diaper." These descriptions don't contain the word "color". In this case, the feature values are hardly learned and extracted. One solution to this problem is to add examples of feature values to the feature database. This may be effective for the feature whose feature values are the limited words.

Second reason is that NTM-Agent extracts not a feature name itself but the word which is near a feature name though the feature values are the feature names in some features. "CD drive" is an example of such features. The item descriptions about CD drive are like "Extra hardware: CD drive FD drive USB". In this case, NTM-Agent searches the feature name "CD drive" and extracts not "CD drive" but "FD drive". One solution to this problem is to define two types of features (Type A and Type B) to change the extracting method. NTM-Agent extracts the words which are near the feature name for Type A. NTM-Agent extracts the feature names themselves for Type B.

6.3 User experiment of NTM-Agent

We conducted a user experiment by giving tasks to subjects. The task is to select an item in the category of laptop computer from the search result of a real net auction (Yahoo!Auction[11]). The subjects were divided into two groups, Group A which uses NTM-Agent and Group B which doesn't use NTM-Agent. We compared Group A and Group B by seeing the two parameters, *selection time*(the time until a subject selects an item) and *link click*(the number of times that a subject clicks a link).The reason why we used the two parameters is that the parameters must have relationship with the amount of the user's work and are measurable explicitly. The experiment procedure is as follows:

1:Preparation After explaining the task to a subject, ask the subject to write out the condition of item which he/she wants.

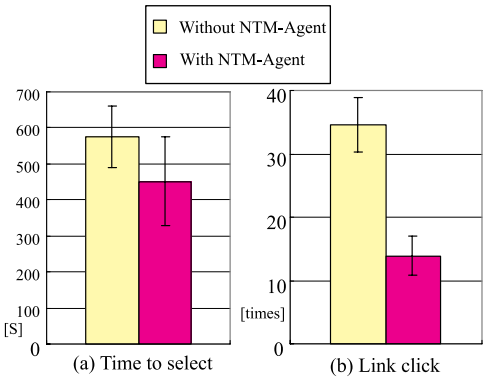

Figure 8: Result of subject experiment

2:Task Give two tasks to the subject and measure the above two parameters for each task. Task 1 is for training of the task, Task 2 is for the evaluation [5].

3:Analysis Calculate the means of the parameters and compare them. To remove extreme cases for accurate evaluation, remove the maximum and the minimum value in each group. Conduct t-test to see the significant difference between the means of the parameters of Group A and those of Group B.

20 users in their twenties or thirties participated in this experiment. 10 users out of them had bought some items in net auctions. In dividing the subjects into two groups, the 10 users were divided into two groups equally. We conducted t-test (two tailed, error probability $\alpha = 0.05$) for Group A and Group B. And we confirmed the significant difference between Group A which uses NTM-Agent and Group B which does not use NTM-Agent in both selection time and link click. The mean of selection time for each group is shown in Figure 8-(a). The mean of link click for each group is shown in Figure 8-(b). NTM-Agent reduces 21% of selection time and 60% of link click. Therefore, we can say that NTM-Agent is effective enough to reduce the burdens of users.

[5]Task 1 uses the search result of "Vaio", Task 2 uses the search result of "Thinkpad".

Table 5: The extracting precision and the extracting recall for each feature

Computer	CPU	Memory	HDD	OS	Display	CD drive	Scratch	Dot disability
Extracting precision [%]	93.4	55.6	68.4	86.7	45.5	33.3	30.8	33.3
Extracting recall [%]	83.3	60	72.2	81.3	76.9	28.6	28.6	14.3

Car	Mileage	Type	Inspection	Color	Engine displacement	AT/ MT	Scratch
Extracting precision [%]	91.3	85	76	15.4	66.7	83.3	20
Extracting recall [%]	75	65.4	65.5	9.1	16.7	52.6	12.5

Baby clothing	Size	Material	Number of clothing	Color	New/Used	Sex	Stain
Extracting precision [%]	84.2	66.7	50	26.1	56	40	25
Extracting recall [%]	74.4	58.3	42.9	54.5	45.2	50.0	20.0

7. CONCLUSIONS

We developed an agent which searches item descriptions, extracts the important information from the item descriptions and displays them in a table so that users can compare the items easily. We focused on two problems to develop such an agent. First problem is that the search results contain noise items. Second problem is that the item descriptions aren't uniform. For the first problem, we proposed a filtering method by the correlation rules of the keywords of the items' titles and item descriptions. For the second problem, we proposed a learning method of the feature values to the missing feature name and a method which distinguishes the description format of item description and extracts information in the most suitable way for the format type. We implemented the agent and conducted two simulation experiments and a user experiment. The conclusions of the experiments are as follows:

- Our filtering method is effective for the category which has noise items to some extent.
- Learning method of feature values in our extracting method contributed to the improvement of the extracting precision and the extracting recall.
- The combination of filtering noise items and extracting item's features in NTM-Agent can reduce a user's work load to select an item to bid.

The problem of NTM-Agent is the preparation of the domain knowledge for search and the domain knowledge for extraction. The service providers must prepare the knowledge. These tasks are burdens for them. Solving these problems allows service providers to offer their services more easily. As future works, we plan to work on the support of the service providers when making the domain knowledge.

These years, the informal texts on the Web like BBS are increasing. The demand of the technologies which deal with these texts is becoming higher. We proposed some technologies in NTM-Agent to deal with such texts in one application(net auctions) and proved the effectiveness. This is worthy for the development of new applications which utilizes informal texts on the Web.

8. REFERENCES

[1] Y.Kusumura, Y.Hijikata, and S.Nishida. Ntm-agent:text mining for net auction. *Proceedings of SAINT2003*, pages 356–359, 2003.

[2] L. Eikvil. Information extraction from the world wide web: a survey. Technical Report 945, Norwegian Computing Center, 1999.

[3] T. Ito, N. Fukuta, T. Shintani, and K. Sycara. Multiagent cooperative bidding support for electronic auctions. *Proceedings of ICMAS2000*, pages 435–436, 2000.

[4] R.B.Doorenbos, O.Etzioni, and D.S.Weld. A scalable comparison-shopping agent for the world wide web. *Proceedings ofthe First International Conference on Autonomous Agents*, pages 39–48, 1997.

[5] J.Ambite, N.Ashish, G.Barish, C.Knoblock, S.Minton, P.Modi, I.Muslea, A.Philpot, and S.Tejada. Ariadne: A system for constructing mediators for internet sources. *Proceedings of ACM SIGMOD Conference on Management of Data*, pages 561–563, 1998.

[6] D.Freitag. Information extraction from html. *Proceedings of AAAI-98*, pages 517–523, 1998.

[7] C.Chang and S.Lui. Iepad:information extraction based on pattern discovery. *Proceedings of WWW2001*, pages 681–688, 2001.

[8] DARPA. *Proceedings of the Seventh Message Understanding Conference(MUC-7)*, 1998.

[9] R.Agrawal, T.Imielinski, and A.Swami. Mining associations between sets of items in massive databases. *Proceedings of ACM SIGMOD Conference on Management of Data*, pages 207–216, 1993.

[10] JUMAN. Morphological analyzer for japanese. http://www-nagao.kuee.kyoto-u.ac.jp/nl-resource/index-e.html.

[11] Yahoo!Auctions. http://auctions.yahoo.co.jp/.