Cleaning of Auction Data for Bidding Decision

Shu-Gang Han and Chia-Hui Chang

Dept. of Computer Science and Information Engineering,

National Central University, Taiwan

Email: kenthan@db.csie.ncu.edu.tw chia@csie.ncu.edu.tw

ABSTRACT

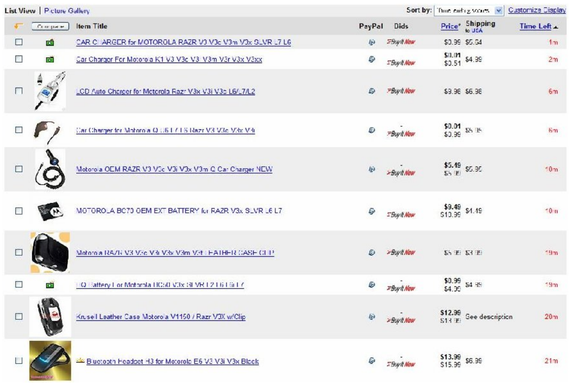
Bidding for products on the Internet has become a common activity in our daily life. However, it's a tedious problem that there are too many items for the bidder to select the cheapest one. In the results providing by eBay, only a small number of results are target items. This is a common situation while the user is searching for a main product in 3C. We aim at helping the bidder compare items easily on auction websites. In this thesis we propose CADBid, which is a web-based system built between auction websites and the bidder. CADBid is able to automatically filter out non-target items and clean the descriptions about these items. Afterward, a list is generated which helps the bidder compare these items. The list only shows the target items along with their important properties. Our work focuses on two tasks. The first task is item filtering. The second is cleaning of descriptions. After cleaning of descriptions, the clean descriptions are used to assist the first task. We view the two tasks as classification problems and propose two feature sets. We build two classification models based on Support Vector Model. Our experiment shows that cleaning of description is helpful because clean descriptions indeed improve the accuracy of item filtering. With CADBid, the bidder will be convenient while making a good decision on which item to bid.

1: INTRODUCTIONS

With the growing of Internet technology, bidding on auction websites has become one of the most popular activities on the Internet. eBay [1] is the most representative one of auction websites. Usually, these website provide a web form for the bidder to search for the targets, where the bidder inputs product name or model name as a keyword and submits it to the search engine of auction website. For example, the bidder looks for a hot cell phone - Motorola V3x, no doubt the bidder inputs "Motorola V3x" as a keyword. After submitting the keyword, the auction website returns all of the matched items in a list page. List pages show the rough information of matched items. They may contain the title, picture, price and so on. Many auction websites provide a sorting function in list pages (Figure 1). The bidder can sort the items according to prices, time left, etc.

After skimming the information on the list pages, the bidder can click the link of the item which he is interested in. The link then redirects to a detailed page of the item (Figure 2). There is much information shown on the detailed pages, like information about the seller, history of every offering by bidders, duration of the item, category which the item belongs to, and a description area for the item.

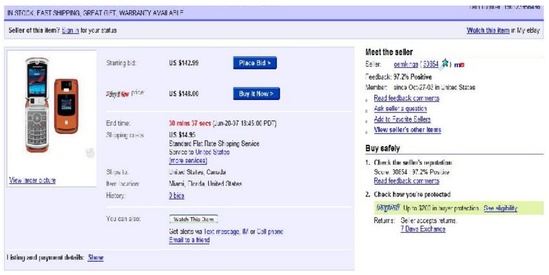
Although every auction website provides a friendly user interface for searching, and users can get all the information from list pages and detailed pages, they still encounter many problems on looking for what they want to buy and bid.



List View | Picture Gallery   Sort by: Time ending soonest   Customize Display

	Compare	Item Title	PayPal	Bids	Price*	Shipping to USA	Time Left
		CAR CHARGER for MOTOROLA RAZR V3 V3c V3m V3x SLVR L7 L6		Buy It Now	$0.99	$5.64	1m
		Car Charger For Motorola K1 V3 V3c V3i V3m V3r V3x V3xx		Buy It Now	$0.01 $0.51	$4.99	2m
		LCD Auto Charger for Motorola Razr V3x V3i V3c L6/L7/L2		Buy It Now	$9.98	$6.98	6m
		Car Charger for Motorola Q L6 L7 L5 Razr V3 V3c V3x V3i		- Buy It Now	$0.01 $0.99	$5.95	6m
		Motorola OEM RAZR V3 V3c V3i V3x V3m Q Car Charger NEW		- Buy It Now	$5.49 $5.99	$5.95	10m
		MOTOROLA BC70 OEM EXT BATTERY for RAZR V3x SLVR L6 L7		Buy It Now	$9.49 $10.99	$1.49	10m
		Motorola RAZR V3 V3c V3i V3x V3m V3t LEATHER CASE CLIP		Buy It Now	$5.99	$3.99	19m
		HQ Battery For Motorola BC50 V3x SLVR L2 L6 L6i L7		- Buy It Now	$0.99 $4.99	$4.99	19m
		Krusell Leather Case Motorola V1150 / Razr V3X w/Clip		- Buy It Now	$12.99 $13.99	See description	20m
		Bluetooth Headset H3 for Motorola E6 V3 V3i V3x Black		- Buy It Now	$13.99 $15.99	$6.99	21m

Figure 1: List Page in eBay



IN STOCK, FAST SHIPPING, GREAT GIFT, WARRANTY AVAILABLE   Item number: [illegible]

Seller of this item? Sign in for your status   Watch this item in My eBay

View larger picture

Starting bid: US $142.99 Place Bid >

Buy It Now price: US $148.00 Buy It Now >

End time: 30 mins 37 secs (Jun-20-07 18:45:00 PDT)

Shipping costs: US $14.95
Standard Flat Rate Shipping Service
Service to United States
(more services)

Ships to: United States, Canada

Item location: Miami, Florida, United States

History: 0 bids

You can also: Watch This Item
Get alerts via Text message, IM or Cell phone
Email to a friend

Listing and payment details: Show

Meet the seller
Seller: oemkings (30854) me
Feedback: 97.2% Positive
Member: since Oct-27-03 in United States
- Read feedback comments
- Ask seller a question
- Add to Favorite Sellers
- View seller's other items

Buy safely
1. Check the seller's reputation
Score: 30854 | 97.2% Positive
Read feedback comments
2. Check how you're protected
PayPal Up to $200 in buyer protection. See eligibility
Returns: Seller accepts returns.
7 Days Exchange

Figure 2: Detailed Page in eBay

The first problem is the huge number of returned items [2]. The total number of items on auction websites is very large. It is increased in an astonishing speed every second. If the user inputs a hot product name as the keyword, thousands or even more items are returned. In consideration of time, it is impossible to read all the items in list pages, even the detailed pages of these items. What's even worse, part of returned items are not what the user expects. The bidder needs extra time to filter out irrelevant items by his eyes.

The problem comes from two reasons. The first reason is that some tokens in the title are not related to the item. In order to sell a product successfully, the seller usually adds some popular keywords to the end of title. For example, the seller adds “Michael Jordan” to a title of a basketball shoes. This trick makes the item gain a high opportunity to be showed in list pages; however, it is not the bidder’s target.

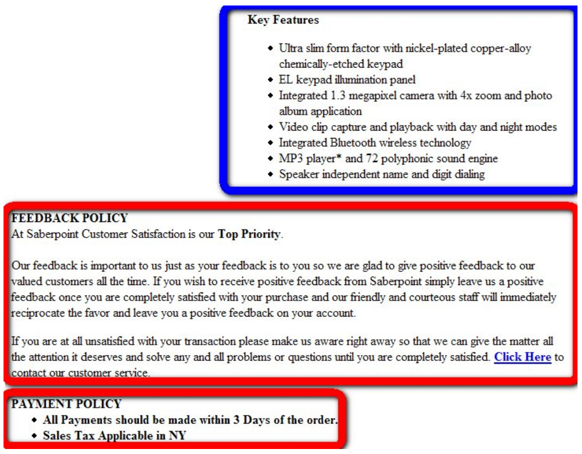
The second reason is that the keyword doesn’t truly represent what the bidder wants. In some categories, especially in 3C products (communications, computers, and consumer products), new products enter the market accompanied with many accessories. For example, battery, charger and headset in the market of cell phone products. Compared to these accessory products, we call the others “main products”. Inputting a product name in such categories gets a complex result set, which is a combination of main products and heterogeneous accessories. In many categories, the proportion of main products to accessories is about 1:9. The bidder wastes a lot of time looking at what he doesn’t want to buy.

The second problem is about the noisy description in the detailed page. Description area is used for content related to the bid, e.g., product status, warranty period, promotions for other bids, and instructions of transaction. In terms of comparing items, some information isn’t important. Figure 3 is an example of description area. There are three blocks in Figure 3. The block circled with blue line is genuine description. The two blocks circled with red line are payment and feedback policies. The latter part is useless while comparing items.

The descriptions in shopping website and description area in auction website are all in HTML format, but there is no constant template in description area for auction items. Every seller wants to make this area different to others, because a unique and attractive description area is important in the competitive market. However, for a buyer who needs a comparison of the many candidates found, such autonomy becomes a hazard for extracting product properties from various description areas.

To solve these problems, Yukitaka et al proposed NTM-agent [2] to conquer these problems. NTM-Agent is a standalone application which filters irrelevant items using correlation rules discovered from users' feedback. NTM-agent uses these labeled data as training data to build a model for filtering irrelevant items. Stand-alone applications have such limitation that they are not easy to distribute. In the information explosive age where most information is viewed from browsers, the ability to integrate with browser is extremely important for such applications. For the second problem, NTM-agent extracts product properties from auction web pages and then generating a property list. The model for property extraction needs prepared domain knowledge.

In this paper, we reconsider such problems and incorporate a framework for integration NTM-agent with browsers. For irrelevant item filtering, we can use either pre-trained model or dynamic model like NTM-agent. For property extraction, we apply supervised text mining technique. On the other hand, we use webpage cleaning [3][4][5] to improve irrelevant item filtering. Webpage cleaning includes segmentation which divides the item descriptions into blocks and a binary classification mode which filters noisy sections.


Key Features

- Ultra slim form factor with nickel-plated copper-alloy chemically-etched keypad
- EL keypad illumination panel
- Integrated 1.3 megapixel camera with 4x zoom and photo album application
- Video clip capture and playback with day and night modes
- Integrated Bluetooth wireless technology
- MP3 player* and 72 polyphonic sound engine
- Speaker independent name and digit dialing

FEEDBACK POLICY
At Saberpoint Customer Satisfaction is our **Top Priority**.

Our feedback is important to us just as your feedback is to you so we are glad to give positive feedback to our valued customers all the time. If you wish to receive positive feedback from Saberpoint simply leave us a positive feedback once you are completely satisfied with your purchase and our friendly and courteous staff will immediately reciprocate the favor and leave you a positive feedback on your account.

If you are at all unsatisfied with your transaction please make us aware right away so that we can give the matter all the attention it deserves and solve any and all problems or questions until you are completely satisfied. **Click Here** to contact our customer service.

PAYMENT POLICY

- **All Payments should be made within 3 Days of the order.**
- **Sales Tax Applicable in NY**

Figure 3: Description Area

To illustrate, the developed system, called CADBid (Cleaning of Auction Data for Bidding Decision), is placed between the auction website and the bidder. When the bidder reaches an auction page where comparison is needed, a pre-installed bookmark is clicked such that the information of current page is sent to system. The system then queries the auction website on behalf of the bidder, filters the items based on its relevancy and extracts properties for comparison. In addition, CADBid accomplishes description cleaning. Users use these genuine descriptions to make item filtering more precise. At the same time, the bidder compares items much easily by reading the genuine descriptions, not the whole descriptions.

In the experiments, we evaluate the effectiveness of irrelevant item filtering using 5 common product categories with pre-trained models. The bidder can use these pre-trained models immediately. CADBid also enables the bidder to train a new model for other new categories. In addition, we train a generalized classification model for description cleaning. By means of CADBid, the bidder will have a more joyful experience shopping online than before. We propose a statistical learning approach to the task, based on SVM (Support Vector Machines) [6]. Our experimental results indicate that the proposed method performs well. Classification of items got a very high accuracy, as well as high precision and recall. Classification of descriptions got an accuracy which is good enough in solving this kind of problem.

The rest of the paper is organized as follows. In Section 2, we introduce related work. In Section 3, we describe our approach in detail. Section 4 gives our experimental results. We summarize our conclusion and future work in Section 5.

2: RELATED WORK

Our research is related to the search fields of agent for electronic commerce and web page cleaning.

2.1: AGENTS FOR ELECTRONIC COMMERCE

In the research area of agent for electronic commerce, some agent systems were developed to help users find their target item from a large number of items. ShopBot [7] and BiddingBot [8] are classical prototypes which summarize the information of items on electronic commerce sites.

ShopBot focused on products which users want. It automatically searches items in shopping websites and summarizes the characteristics of the items, which helps the user find the cheapest item at one glance. To achieve this goal in a specific website, ShopBot begins with a pre-process which analyzing the website.

For a specific product domain, ShopBot needs some additional domain knowledge, and then it is capable of analyzing this kind of product domain. Domain knowledge contains the examples of property values. ShopBot conducts keyword matching between the examples of property values and the text of the page and learns where the property values are generally described in the page. Unlike ShopBot, CADBid extract genuine descriptions from product page and filter out noise products. This function is crucial for the online-bidding websites where miscellaneous sellers participate in.

BiddingBot supports the user in attending, monitoring, and bidding in multiple auction websites simultaneously. BiddingBot monitors prices of goods in several online auction websites to get reasonable market prices of goods, and uses a new cooperative bidding mechanism to effectively bid in auctions. In general, most of online auctions are classified into common-value auctions. For example, auctions for personal computers and cars are common-value auctions, because we can see a market price as a common valuation among bidders. In the case of a common-value auction, a bidder who wins an auction is the one with the most optimistic information. Therefore, it is an advantage to know real valuations of an item, since a bidder avoids the winner’s curse if she knows a correct valuation (i.e. a market price) of an item. Since it is hard for a bidder to monitor, attend, and bid in multiple auctions simultaneously, the authors use agents to cooperatively monitor, attend, and bid on behalf of the users. At the same time, a market price of an item is predicted.

BiddingBot consists of one leader agent and several bidder agents. Each of bidder agents is assigned to an auction site. Bidder agents cooperatively gather information, monitor, and bid in the multiple auction sites simultaneously. The leader agent facilitates cooperation among bidder agents as a matchmaker, sends the user’s request to the bidder agents, and presents bidding information to the user. In BiddingBot, each of bidder agents is specified to its auction site and can behave as a flexible wrapper, since the different auction sites represent information in different forms. The results of experiments demonstrate that the cooperative bidding mechanism effectively bid in multiple auctions.

In 2004, K. Yukitaka et al proposed NTM-agent [2]. The authors aim at supporting the bidders on auction websites by automatically generating a list which contains the properties of items for comparison. NTM-agent is a powerful agent-based system. The system collects Web pages of items and extracts the items’ properties from the pages. After that, a list which contains the extracted properties is prepared. The authors mentioned two problems. The first problem is that if the system collects items automatically, the results contain the items which are different from those of the user’s target. The second problem is that the formats of descriptions in auction websites are not consistent (There are different formats such as sentences, list items, images, and tables.). Therefore, it is difficult to extract the information from the descriptions by conventional methods of information extraction. The authors proposed methods to solve these problems. For the first, NTM-Agent uses correlation rules filtering the items. These rules are keywords in the titles and descriptions. They are created semi-automatically by a support tool. For the second, NTM-Agent extracts the information by distinguishing the formats. It also learns the property values from examples for the future extraction.

The authors have proved that the filtering method is effective for some product category which has noise items to some extent and the system actually reduce a bidder’s work load to select an item to bid; even so, NTM-agent has two problems. One is the preparation of the domain knowledge and the other is the low accuracy of item filtering. Domain knowledge includes: template for the search result pages, template for the item’s pages, and property names. They are necessary for general extraction but which means the system cannot be easily adapted to a new website. Besides, it uses correlation rules to filter out noise items. CADBid targets the same objective and we overcome the latter problem. The accuracy is strengthened by our sophisticated classifier.

2.2: DATA CLEANING

With the prevalence of web pages, many researches focus on web page cleaning. A web page usually consists of informative content and non-informative content. Informative content is what users are interested in, like a news article in CNN.com. Non-informative content is generated dynamically, containing advertisements, image-maps, plug-ins, logos, counters, search boxes, category information, navigational links, related links, footers and headers, and copyright information. The purpose of web page cleaning is to filter out non-informative content. At the same time, the result of data mining could be reinforced.

The common step is to subdivide the whole web page into smaller semantically homogeneous blocks. The step is called "segmentation." Many approaches have been proposed for automatic page segmentation, which fall into two categories. One mainly depends on HTML DOM trees [4] [5]; the other uses visual cues such as location, color, and separators to simulate how a user understands the semantic structure of a page [9][10][11]. Due to the heterogeneity in description area and JavaScript codes, we usually get wrong visual cues about HTML tags. Finally we decide to segment pages depending on DOM trees.

After segmentation, some researches quantify the degree of importance and give every block a score [3][4]; others view each block as an instance, and define feature sets for classification model [5]. In our work, we view it as a classification problem. Informative block is the block which contains genuine product descriptions. The block which contains ads and instructions is considered non-informative block. The difference between them is the semantics inside the block. We use the tokens as feature set for classification model.

Yi and Liu [3] propose a noise elimination technique based on the following observation: In a website, blocks of noisy content usually have some common contents and presentation styles, while those of main content are often diverse in their actual content and presentation styles. Based on this observation, they propose a tree structure Site Style Tree (SST), which is used to capture the common presentation styles and the actual contents of the pages in a given Web site. Then they introduce an information based measure to determine that given parts of the SST represent noises or main contents. By mapping this page to the SST, noises within given page can be detected and eliminated.

Lin and Ho [4] first partitions a page into several content blocks according to HTML tag <TABLE> in a Web page. Based on the occurrence of the features (terms) in the set of pages, it calculates entropy value of each feature. According to the entropy value of each feature in a content block, the entropy value of the block is defined. By analyzing the information measure, they propose a method to dynamically select the entropy-threshold that partitions blocks into either informative or redundant.

Moreover, our work is related to email data cleaning. Tang et al. [12] proposes a cascaded approach and makes a significant improvement on extraction accuracy when applying to term extraction. The system takes advantage of classification model and defines an excellent feature sets. Here the classification instances are sentences in an email, which are different from blocks in web page cleaning.

3: CADBID SYSTEM

The user has two ways to send information to CADBid. The user can input the information through the web form in CADBid. The other way is very friendly to the user. When the user is surfing on eBay and comparing some items, he can use CADBid through a bookmark which redirects the user to CADBid. CADBid is able to get information of what the user was comparing by some JavaScript codes inside the bookmark.

CADBid provides 5 pre-trained category models. In web form, user can choose one of them which the product belong to. If the user doesn't want to choose the model, CADBid will automatically select the most appropriate model from the five models. The mechanism is default when the user uses CADBid through a bookmark in previous scenario.

Here we use a simple mechanism to achieve this function. We use product name along with one of the five model names as search keywords, and query Google. The model name which gets the most number of results is the most appropriate one.

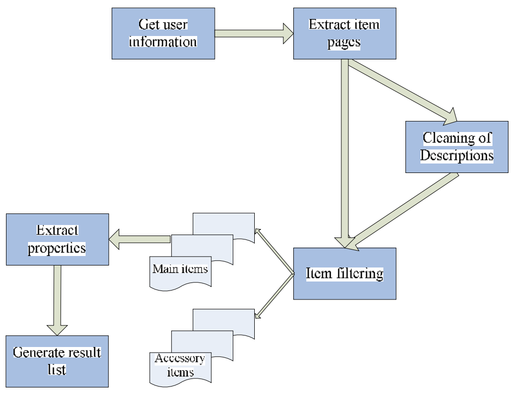


Figure 4 System Architecture

Figure 4 shows system architecture. After getting the information from the user, CADBid uses the product name to query eBay and extracts all the information about the result items in result pages. For example, item title, current price, and time-left are extracted. Compared to descriptions in the detailed page, they are basic features of these result items.

In order to filter items, CADBid can just use the basic features. We define a feature set to do the item filtering according to this basic information. On the other side, CADBid has another path of process to do item filtering. We clean the descriptions in detailed page and use them as additional features in item filtering. The steps include segmenting the description area and using another classification model filtering out unnecessary descriptions (The details about cleaning of descriptions are in Section 3.5). By the assistance of clean descriptions, we hope to get a better result in item filtering compared to basic features.

After item filtering, main products are classified by CADBid. CADBid uses supervised text mining techniques to extract important product properties from the detailed page. In the last step, CADBid generates a list showing these main products with their properties. The user can compare them easily.

3.1: CLASSIFICATION MODEL

We make use of Support Vector Machines (SVM) [6] as the classification model. The SVM implementation we use is from Weka [13]. We use default setup because it works best for our tasks.

3.2: PREPROCESSING

We collect data from eBay because it's the most popular auction website. As mentioned before, we prepare data for 5 product categories and assign 4 popular products for each category (ex: Motorola V3i). We collect 100 product pages for one product. There are totaling 2,000 pages which are downloaded randomly from eBay. The numbers of target items in every product are different. The proportion of main products to accessories is about 1:9. For every page, we extracted title, seller name, current price, and its category. The HTML text of description area is stored in a text file for subsequent description cleaning.

3.3: ITEM FILTERING

In our problem, instances in classification are auctions in eBay. Item filtering consists of two stages: training phase and testing phase. In training phase, we prepare an input table. Input table consists of labeled instances along with their feature values. The label represents whether the auction is target item or not. The feature values are computed based on the feature set we define. SVM model is generated according to the input table. The key issue here is how to define an effective feature set. The feature set is as follows:

- **Price**: The current price of each auction divided by average price.
- **Frequent tokens in title**: We count the frequencies of every token occurred in all titles belonged to the same product category. Stopwords are removed first. Then we propose a sequence of binary values representing frequent words rank 1 to rank 50.
- **Catalog**: An auction belongs to only one node of catalog tree. The catalog tree is compiled by eBay and they hope that each seller choose the correct node selling their products. For example, cell phone should be listed in "Cell Phones / Phones Only", not "Cell Phones / Accessories". However, due to unexpected reason, there are always some auctions listed in wrong node. For every product, we count the occurrences of every node and use the number to represent the node feature.
- **Descriptions**: These features are from raw descriptions or clean descriptions. We will elaborate it in experiments.

3.4: CLEANING OF DESCRIPTIONS

In description area, it's not easy to find out useful information (genuine description) at first glance. Using data mining techniques or NLP techniques to extract useful data from the area isn't easy, too. Various HTML tags and script language adds to the complexity of this area.

Though complex, this area contains significant information which helps the bidder compare items. These genuine descriptions could be used to enhance the accuracy of item filtering in CADBid. In opposition to genuine description, other information in the area is regarded as noises. In order to utilize genuine descriptions in item filtering, we build another classification model to extract them in description area.

HTML page can be parsed into a DOM tree (Document Object Model tree). Every HTML tag in the page means a node in the DOM tree. In HTML's viewpoint, anything between start tag "<xxx>" and end tag "</xxx>" are viewed as inner contents belonged to the tag "xxx". Inner contents may be plain text or other tags. Mapped to tree structure, inner contents are equivalent of children nodes.

According to our observations in tree structure, we design an algorithm to segment the DOM tree into a number of sub-trees. These sub-trees represent different blocks in the description area. Based on the feature set we defined, every block has its own characteristic. We train a SVM model to filtering out unnecessary blocks. With the help of remaining blocks (genuine descriptions), the accuracy of item filtering is improved.

In abstract view, description area can be divided into blocks. Some are product descriptions, others are advertisement, and still others are instructions of transaction. Every block has its own corresponding sub-tree in the DOM tree. By applying our algorithm, we could segment the area into some separate blocks by dividing the DOM tree into some sub-trees. Then we use our classification model to classify these blocks.

Blocks containing genuine description are remained. Other blocks are rejected. We use these remained blocks to improve the item filtering, which is our main objective.

We develop a deep-first algorithm to segment the description area into blocks. First, we extract the HTML text of description area from an auction page. Second, we transform the text into a DOM tree. Here we use CyberNeko HTML parser [14] to complete the task. A DOM tree transformed by the parser is a tree structure. The structure provides many useful functions, like *getChildNodes()*, *getNextSibling()*, and *getAttributes()*. Third, we input the root node of the DOM tree into segmentation program. Finally, the program outputs the blocks information and their properties.

```
Input: root node of DOM tree
Algorithm: getNodeText (Node t)
1     If all children of t are Decorate Tags
2           get all text belong to children of t;
3           merge all text;
4           obtain features from whole text;
5           Return
6     End if
7     If t is a TEXT node
8           get text belong to t;
9           obtain features from the text;
10          Return
11    End if
12    If t is one of {SCRIPT,STYLE} tag
13          Return
14    End if
15    While (Node child of t != NULL)
16          getNodeText(child);
17    End While
```

Figure 5: Algorithm of page segmentation

In a composition, related contents are put together in a paragraph. Like the way we write composition, related contents are put together in an abstract block. From our empirical observation, sellers usually use some HTML tags to separate blocks. We define these HTML tags as Paragraph Tags. For examples, <TABLE> and <P> are Paragraph Tags. Except for these Paragraph Tags, other tags are used to decorate the content. We define these HTML tags Decorate Tags. For examples, <FONT> and <B> are Decorate Tags. In our algorithm, the way we deal with the node depends on the type of its tag.

Let's look at the algorithm in detail. We traverse the DOM tree in a deep-first way. In line 1, we check all the children of node t. If all the children of node t are Decorate Nodes, we consider it a block. We merge all descendant nodes belong to t. All text belong to these descendant nodes are combined into a chunk of text. At the same time, we gather features of the block for subsequent classification. In line 7, the condition is matched while t is a terminal node. We have to extract the text it contains and the features. In line 12, we reject the node whose tag is one of <SCRIPT> and <STYLE>. Script codes and CSS codes are not related to the description. They are not what we need. In line 15, if above conditions are not matched, we use every child of t as input and recursively call the function *getNodeText()*.

After the traverse, we get some blocks. In classification's viewpoint, every block is an instance. We label each instance whether it's positive or negative. The process is done by hands. Positive instance means it's a genuine description. Negative instance means it's a noise. We train a classification model by these labeled instances. Then we use the model to get genuine descriptions from incoming auction pages.

The features we use are as follows:

- **Number of characters:** Total number of true text in the block.
- **Co-occurrence in title and in block:** A block which has token matching the item title is more likely to contain genuine descriptions. The value is the total number of matched token in the block.
- **Occurrence of common tags:** We define 26 common tags and the value is total occurrence of specific tag divided by number of characters.
- **Tokens with high mutual information:** The concept is similar to entropy in information theory [15]. It considers term distribution among the classes and has been found to be particularly effective.

4: EXPERIMENT

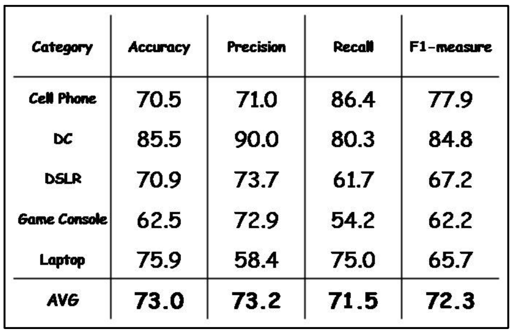
4.1: ITEM FILTERING

In item filtering, we run some setups to see the results while using basic features, features from raw descriptions, and features from clean descriptions. The dataset is the same as we mentioned in Section 3.3. For the data in each category, we run a 4-fold cross-validation.

First, we run an experiment with basic features as a baseline result. The experiment result is in Table 1. The results are different for different product categories. The category "Game Console" gets a worse result than other categories. By our observation, the differences between the four game consoles we assigned are huge. Some are handheld game consoles; the others are large-scale game consoles which are only played at home.

Although the result is not very good, it's sufficient for the bidder to filter out many accessory products. After basic item filtering, the proportion of main products to accessory products is from 1:9 to 3:1. Besides, the processing time is very short because we extract these basic features from list pages, not detailed pages. In terms of user of CADBid, short waiting time implies comfortable using experience.

In Figure 6, we show the comparison among basic features, features from raw descriptions, and features from clean descriptions. The features from raw descriptions or clean descriptions are all possible tokens occurring in the descriptions. Stopwords are removed but the numbers of raw features or clean features are still very big.



Category	Accuracy	Precision	Recall	F1-measure
Cell Phone	70.5	71.0	86.4	77.9
DC	85.5	90.0	80.3	84.8
DSLR	70.9	73.7	61.7	67.2
Game Console	62.5	72.9	54.2	62.2
Laptop	75.9	58.4	75.0	65.7
AVG	73.0	73.2	71.5	72.3

Table 1: Result of basic item filtering

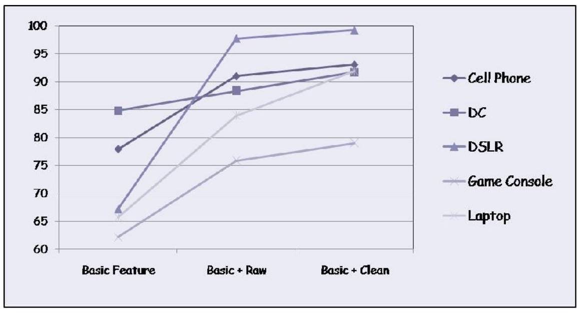


Figure 6: Comparison among three different feature sets

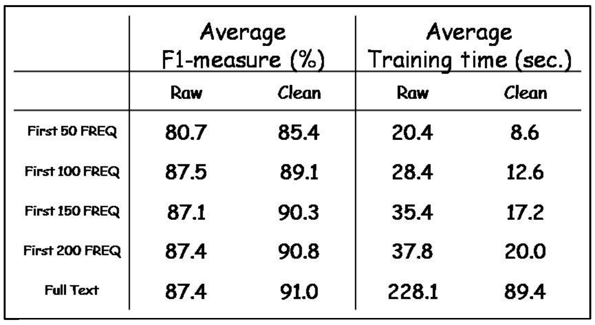
In Figure 6, we see that the five lines all have a positive slope, which means the result is getting better from basic features, raw features, to clean features. From basic features to raw features, we all get an apparent improvement for the five categories. It means that information in descriptions is useful for filtering items. Clean features from clean descriptions are more useful than raw features because some information helpless to the classification model is removed. Here the useless information means ads and instructions of transaction in raw descriptions. By the comparison, we see that our cleaning of descriptions is useful while deciding the features for item filtering.

In spite of using all possible tokens as features gets good result, the processing time leads to a long waiting time. Processing time is equal to training time of classification model plus pre-processing time of data. To avoid long processing time, we need a feature selection.

In setup 2, the difference between raw and clean descriptions is not obvious. Even raw descriptions outperform clean descriptions in some categories, which is not what we suppose to be. But in setup 3, we see that clean descriptions outperform raw descriptions apparently. We see that using TF-IDF value instead of 0/1 affects the performance. In following setup 4-7, we use limited number of feature values. Clean descriptions always outperform raw descriptions.

To speed up the processing time, we simply calculate the frequency of all tokens occurring in raw or clean descriptions and use tokens as features with high frequency. We try four different numbers of features with high frequency. Table 2 shows the result.

Selected feature sets get results which are a little bit lower than result of feature set from full descriptions. As the number of selected features grows, the result is getting better. It's reasonable that longer number of feature set provides more helpful information in classification. When we use the first 200 tokens with high frequency in clean descriptions as features, the average F1-measure is very close to features from all possible tokens in clean descriptions. In terms of processing time, selected feature needs a very short processing time, which is lower than one-sixth of full text in raw descriptions, low than one-fourth of full text in clean descriptions.



	Average F1-measure (%)		Average Training time (sec.)	
	Raw	Clean	Raw	Clean
First 50 FREQ	80.7	85.4	20.4	8.6
First 100 FREQ	87.5	89.1	28.4	12.6
First 150 FREQ	87.1	90.3	35.4	17.2
First 200 FREQ	87.4	90.8	37.8	20.0
Full Text	87.4	91.0	228.1	89.4

Table 2: Performance of descriptions cleaning

In designing CADBid, we think selected features are very useful. Instead of using feature set from full descriptions, we use selected feature set, which leads to a sufficient accuracy in item filtering and a bearable processing time.

4.2: CLEANING OF DESCRIPTIONS

We use 6 setups to see the effectiveness of different feature sets. The dataset is randomly picked from the whole blocks which are segmented from 2,000 pages before. We picked 2,000 blocks as our dataset in description cleaning. Each category provides the same number of training instances. For every setup, we run a 10 cross-validation. The result is in Table 3.

	Precision	Recall	**F1-measure**
Basic	67.2	87.3	**75.9**
Basic+Tag	67.9	87.4	**76.4**
All(MI 100)	71.9	90.8	**80.2**
All(MI 200)	74.9	90.4	**81.9**
All(MI 300)	76.3	89.8	**82.5**
All(MI 400)	77.3	90.1	**83.2**

Table 3: Performance of descriptions cleaning

In the first setup we use basic features. They are "number of characters", "co-occurrence in title and in block", and "occurrence order in description area". In the second setup we use basic features plus tag features. By comparing setup 1 and 2, we see that the tag information is not very helpful. The F1-measure only increases 0.5. We think it's not easy to differentiate genuine descriptions from noise descriptions using tag information because the obvious difference between them is in vocabulary.

Setups 3-6 use above features plus tokens with higher mutual information. We try different number of tokens with the highest mutual information. The value of MI features is 0 or 1 based on the existences of these tokens.

If we focus on the precision, we see that mutual information helps the performance remarkably. With a higher number of MI features, the precision gets higher, but the improvement is getting small while the number of MI features is getting higher. More features mean more information, but the feature set with “MI 100” has already included the most important tokens in descriptions filtering. So the new incoming tokens with high mutual information give less help to the classification model, which leads to a small improvement in precision.

In future work, we plan to combine CADBid with some information extraction techniques. It’s more powerful while able to extract data from multiple auction websites. In addition to auction websites, it’s also possible to make CADBid process data from online shopping websites. It a worthy task to make the users get what they want on e-commerce websites.

5: CONCLUSION

In the research, we focus on the scenario of bidding on auction websites. We have investigated 2 problems: item filtering and descriptions cleaning. We have proposed a 2-step approach. First we use classification model to filter out noise descriptions. Then we use genuine descriptions to differentiate main products from accessories by another classification model. Using these techniques above, we implemented a system CADBid which helps the bidders compare items easily.

Description cleaning is helpful in two points. First, it makes the bidder compare items fast and easily than before. Showing only genuine descriptions, the bidder doesn’t have to pay attention to descriptions unrelated to the item. Second, by the help of clean description, the accuracy of item filtering is improved. At the same time, we don’t need much time to train the model.

Experimental results show that our approach has a high accuracy in item filtering. With a good feature selection mechanism, classification with clean descriptions outperforms that with raw descriptions. In descriptions cleaning, the accuracy is not very good but sufficient for our requirement.

In future work, we plan to combine CADBid with some information extraction techniques. It’s more powerful while able to extract data from multiple auction websites. In addition to auction websites, it’s also possible to make CADBid process data from online shopping websites. It a worthy task to make the users get what they want on e-commerce websites.

6: ACKNOWLEDGEMENT

This paper is sponsored by National Science Council, Taiwan under grant NSC95-2221-E-008-076.

7: REFERENCE

[1] http://www.ebay.com.

[2] K. Yukitaka, H. Yoshinori, and N. Shogo, "Text Mining Agent for Net Auction," in Proceedings of the 2004 ACM symposium on Applied computing: ACM Press, 2004.

[3] L. Yi, B. Liu, and X. Li, "Eliminating Noisy Information in Web Pages for Data Mining," In Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Washington, DC, USA, 2003.

[4] S.-H. Lin and J.-M. Ho, "Discovering Informative Content Blocks from Web Documents," in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining Edmonton, Alberta, Canada: ACM Press, 2002.

[5] S. Debnath, P. Mitra, N. Pal, and C. L. Giles, “Automatic Identification of Informative Sections of Web Pages,” IEEE Transactions on Knowledge and Data Engineering 17, 9, Sep. 2005.

[6] V. Vapnik, Statistical Learning Theory. New York: Springer Verlage, 1998.

[7] B. D. Robert, E. Oren, and S. W. Daniel, "A Scalable Comparison-shopping Agent for the World-Wide Web," in Proceedings of the first international conference on Autonomous agents Marina del Rey, California, United States: ACM Press, 1997.

[8] T. Ito, N. Fukuta, T. Shintani, and K. Sycara, "BiddingBot: A Multiagent Support System for Cooperative Bidding in Multiple Auctions," in Proceedings of ICMAS2000, 2000, pp. 435-436.

[9] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, “Block-based Web Search,” In Proceedings of the 27th Annual international ACM SIGIR Conference on Research and Development in information Retrieval, 2004.

[10] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma, “Learning Block Importance Models for Web Pages,” In Proceedings of the 13th international Conference on World Wide Web, 2004.

[11] P. Xiang, X. Yang, Y. Shi, “Effective Page Segmentation Combining Pattern Analysis and Visual Separators for Browsing on Small Screens,” Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2006.

[12] J. Tang, H. Li, Y. Cao, and Z. Tang, "Email Data Cleaning," in Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining Chicago, Illinois, USA: ACM Press, 2005.

[13] Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/

[14] CyberNeko HTML Parser, http://people.apache.org/~andyc/neko/doc/html/index.html

[15] Y. Yang and J. P. Pedersen. “A Comparative Study on Feature Selection in Text Categorization,” In Proceedings of the Fourteenth International Conference on Machine Learning (ICML’97), pages 412~420, 1997.