ANADOLU ÜNİVERSİTESİ BİLİM VE TEKNOLOJİ DERGİSİ
ANADOLU UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
Cilt/Vol. : 9-Sayı/No: 1 : 73-86 (2008)

ARAŞTIRMA MAKALESİ /**RESEARCH ARTICLE**

INFORMATION EXTRACTION FROM E-COMMERCE WEBSITES USING SEQUENTIAL WORD GROUP FREQUENCIES

Ferkan KAPLANSEREN [1], Tatyana YAKHNO [2], M. Kemal ŞIŞ

ABSTRACT

In this paper we present an approach how to extract information from e-commerce web sites. Automatic information extraction is applied to e-commerce web sites to construct a description of products. This description contains set of features of the product and their possible values. We implement a new algorithm based on sequential word group frequencies and syntactical rules to extract the semantics. Results are presented and interpreted for future works toward designing an e-commerce shopping agent.

Keywords: Semantics, Sequential word group frequency, Information extraction, E-commerce agent.

ARDIŞIK KELİME GRUP FREKANSLARINI KULLANARAK E-TİCARET WEB SİTELERİNDEN BİLGİ ÇIKARIMI

ÖZ

Bu çalışmada e-ticaret web sitelerinden bilgi çıkarımının nasıl yapılacağına dair bir yaklaşımı sunuyoruz. Otomatik bilgi çıkarımı, e-ticaret web siteleri üzerinde uygulanarak ürünlerin tanımlanması sağlanır. Bu ürün tanımlaması, ürünün özellikler kümesini ve muhtemel değerlerini içerir. Anlam çıkarımını sağlamak için ardışık kelime grubu frekanslarına ve söz dizimsel kurallara dayanan yeni bir algoritma gerçekleştirilmiştir. Sonuçlar gelecek çalışmalarda bir e-ticaret alışveriş ajanının tasarımı için sunulmuş ve yorumlanmıştır.

Anahtar kelimeler: Anlam, Ardışık kelime grubu frekansı, Bilgi çıkarımı, E-ticaret ajan.

1. INTRODUCTION

The World Wide Web (WWW) is now the largest source for Information Extraction (IE) but, because of its structure, it is difficult to use this information in a systematic way. The scope of the data in web pages consists of text, video, images, sounds and etc. Much of the information is embedded in text documents and relies on human interpretation and processing, at the same slow pace as before (Hui and Yu, 2005). According to Moens (2006) the use of the term extraction implies that the semantic target information is explicitly present in a text's linguistic organization, i.e., that it is readily available in the lexical elements (words and word groups),

[1,] Dokuz Eylül Üniversitesi İşletme Fakültesi, İşletme Bölümü 35160 Buca, İZMİR.
Fax: 0.232.453. 50.62, **E-mail:** ferkan.kaplanseren@deu.edu.tr

Received: 28 August 2007; **Revised:** 11 January 2008; **Accepted:** 15 February 2008

the grammatical constructions (phrases, sentences, temporal expressions, etc.) and the pragmatic ordering and rhetorical structure (paragraphs, chapters, etc.) of the source text. Relation between IE and the fields of Natural Language Processing (NLP), Knowledge Acquisition (KA), Machine Learning (ML) and Data Mining (DM) of artificial intelligence is important to process the text in WWW.

Statistics and layout of html pages play a strong role in processing the unstructured text. Filatova and Hatzivassiloglou (2003) present a statistical system that detects, extracts, and labels atomic events at the sentence level without using any prior world or lexical knowledge. Burget (2005) proposes an alternative information extraction method that is based on modeling the visual appearance of the document. Burget (2004) also explains a hierarchical model for web documents that simply presents some structured data such as price lists, timetables, contact information, etc. Objective of Mukherjee, Yang, Tan and Ramakrishnan (2003) is to take a HTML document generated by a template and automatically discover and generate a semantic partition tree where each partition will consist of items related to a semantic concept. Hidden Markov Model (HMM) is one of the main methods that uses the probability for lexical analysis. Bikel, Schwartz and Weischedel (1999) present IdentiFinder, a HMM that learns to recognize and classify names, dates, times, and numerical quantities. Borkar, Deshmukh and Sarawagi (2001) present a method for automatically segmenting unformatted text records into structured elements. Their tool enhances on HMM to build a powerful probabilistic model that corroborates multiple sources of information including, the sequence of elements, their length distribution, distinguishing words from the vocabulary and an optional external data dictionary.

On the other hand, most web users expect more human-like intelligence from their web applications. Recently, e-commerce is one of the most important and money making engine of web based applications. The e-commerce platform will eventually provide a platform where manufacturers, suppliers and customers can collaborate (Chaudhury and Kilboer, 2002). The expectation may be realized, apart from other factors, by subscribing more participants in this collaboration. The point of interest being e-commerce, trusting one's personal computer (it's application in reality) is a matter of intelligent e-commerce agent" that can extract the features of desired commodity.

E-commerce shopping (buyer) agents help customers to find the products and services from internet. According to Leung and He (2002), one of the things that an ecommerce agent can do is to monitor and retrieve useful information, and make transactions on behalf of their owners or analyze data in the global markets. Shopping bots (buyer agents) agent of eBay like eBay (http://www.ebay.com) find products and services that customers are searching for. Jango (http://www.jango.com) and adOne (http://www.addone.com) are other examples for e-commerce agents (Chaudhury and Kilboer, 2002). Jango does comparative price shopping. It is a wrapper visiting several sites for product queries and obtains prices. "adOne" searches classified advertisements like used cars. Another shopping bot that is used by akakce (online web store) is called as bilbot (http://www.akakce.com) makes price comparison (Karataş, 2001). Ahmed, Vadrevu and Davulcu (2006) improved a system called Datarover which can automatically crawl and extract all products from online catalogues. Their purpose is to transform an online catalogue into database of categorized products. Crescenzi, Mecca and Merialdo (2001) investigate techniques for extracting data from HTML sites through the use of automatically generated wrappers. Their software RoadRunner tokenizes, compares and aligns the HTML token sequences tag by tag. Arasu and Garcia-Molina (2003) study the problem of automatically extracting the database values from such template-generated web pages without any learning examples or other similar human input.

Semantic web project has been around for over a decade, but it seems that this project will not be realized for near future. Managing the unstructured information of internet implies discovering, organizing and monitoring the relevant knowledge to share it with other applications or users. To extract the semantics from text documents will be a matter unless web pages are designed in a standard format. Therefore, deriving semantics from unstructured or semistructured web will be as important as designing a semantic web.

In this study, we propose a new algorithm based on information extraction and knowledge discovery from e-commerce web pages. We study lexical contextual relations to show that relative frequencies of sequential word groups including n-word(s) allow agent to derive semantics from a collection of e-commerce web pages. For that reason, the input set of this study is the set of web pages, $WP = \{wp_1, wp_2, \ldots, wp_n\}$, from e-commerce web sites in internet. The purpose is to produce the output which is the definition of a product in terms of possible features and values. A product is defined as a set of features like $P = \{F_1, F_2, F_3, \ldots, F_n\}$ and a feature is defined as a set of values like $F_i = \{V_1, V_2, V_3, \ldots, V_k\}$. We developed a software called FERbot (Feature Extractor Recognizer bot) to evaluate our approach. FERbot learns the features of products automatically by extracting sequential word groups (*SWG*) from web pages and finds out the values of these features to automatically construct a knowledge base for products. This algorithm can be considered as an alternative for the current statistical and machine learning techniques for information detection and classification. For this reason, usage of the sequential word group frequencies is the main difference of this study from the other agent based applications. Also, text miners can use our algorithm for knowledge discovery within the concept of statistical data mining techniques to operate the information extracted from texts.

The paper is organized as follows. The details of our algorithm how to extract information and build a knowledge base are explained in section 2. We present the architecture of FERbot section 3. Case studies and results are presented in section 3.1. In Section 3.2 we present the comparison of results with similar works. Finally in section 4 we state our conclusion and possible future extensions of this study.

2. INFORMATION EXTRACTION FOR BUILDING KNOWLEDGE BASE

Every product is defined by its features. An advertisement about a product in an e-commerce site firstly includes price of that product and some additional information like size, brand, power and etc. Because every e-commerce web page does not include the same standard notation for information about same product, it will be interesting to define the general characteristics of that product. Every product has some features and every feature has some values. For example, a car has some features like color, fuel type and motor power, etc. A color feature of a car has some values like red, blue and white.

If we denote a product by P and if F_i and V_j refer to the i^{th} feature and j^{th} value, respectively, we define a product as a set of features, $P = \{F_1, \ldots, F_n\}$ and a feature as a set of values, $F_i = \{V_{i_1}, \ldots, V_{i_m}\}$, $i, j, n, m \in Z^+$ Finally, a product can be represented as $P = \{F_1 = \{V_{1_1}, \ldots, V_{1_k}\}, \ldots, F_n = \{V_{n_1}, \ldots, V_{n_t}\}\}$, $k, t \in Z^+$. Every product may have different number of features and every feature may have different number of values. For example, feature and value representation of product "car" can be shown as:

Car = {color={red, blue, green},fuel type={benzine, diesel, lpg}, production year = {1946, 2004, 2005, 2006}}.

According to Grishman (1997), the process of information extraction has two major parts; first the system extracts individual "facts" from the text of a document through local text analysis. Second, it integrates these facts, producing larger facts or new facts (through) inference. Stevenson (2006) emphasizes the importance of usage of multiple sentences for facts instead of single sentence. Karanikas (2002) thinks that primary objective of the feature extraction operation is to identify facts and relations in text. Some knowledge engineers try to construct knowledge bases from web data based on information extraction using machine learning (Craven, DiPasquo, Freitag, McCallum, Mitchell, Nigam and Slattery, 2000). To collect knowledge about a product, Lee (2004) works with specially designed interfaces to allow domain experts easily embed their professional knowledge in the system. Gomes and Segami (2007) study the automatic construction of databases from texts for problem solvers and querying text databases in natural language, in their research. The system that is offered by Shamsfard and Barforoush (2004) starts from a small ontology kernel and constructs the ontology through text understanding automatically.

Assume that we have a set of e-commerce web pages WP. $WP = \{wp_1, wp_2, \ldots, wp_n\}$, $n \in Z^+$. Every wp_i is a pair of words and keywords, $wp_i = <W_i, K_i>$, where $W_i = [w_{i_1}, \ldots, w_{i_p}]$ and $K_i = \{k_{i_1}, \ldots, k_{i_s}\}$, $i, p, s \in Z^+$. W_i is the list of words where every w_{i_j} appears in the body part of i^{th} html page, K_i is the set of keywords of html pages where every k_{i_j} is defined in meta-keyword-tag of i^{th} html page.

If we combine all lists of words, W_{i}, successively we create a new list of words, $S=[W_1,W_2,W_3,...,W_n]$. Order of words is important in list S and a word can appear more than one time in this list. If we collect all sets of keywords, K_{i}, in a single set, we create a new set of keywords, $K=\{K_1,K_2,K_3,...,K_n\}$. We analyze the list S is in section 2.1 to find the features and values of a product and we work over the set K in section 2.2 to refine these features and values.

2.1. Sequential Word Groups

In our study, frequencies of n-words groups are used to extract features and values from the list S. We assume that firstly the name of the feature appears and then the value of this feature appears in a text to explain a product. Also we assume that a feature can be represented by a single word or by a sequence of words and a value can be represented by a single word.

Let $SWG_{\mathrm{i}}^{\mathrm{r}}$ denote a sequential word group consisting of r words, starting from i^{th} word of list S.

The list of sequential word groups that have only 1-word can be shown by SWG^1 where $SWG^1=[SWG_1^1, \dots, SWG_n^1]=S$.

The list of sequential word groups that have 2-words can be shown as $SWG^2=[SWG_1^2, \dots, SWG_n^2]$ where $SWG_{\mathrm{i}}^2 = w_{\mathrm{i}}^2 \; w_{\mathrm{i+1}}^2$

Finally, the list of sequential word groups that have r-words can be shown as $SWG^{\mathrm{r}}=[SWG_1^{\mathrm{r}},...,SWG_n^{\mathrm{r}}]$ where $SWG_{\mathrm{i}}^{\mathrm{r}} = w_{\mathrm{i}}^{\mathrm{r}} \; w_{\mathrm{i+1}}^{\mathrm{r}} \; ... \; w_{\mathrm{i+r-1}}^{\mathrm{r}}$.

We analyze the list S to find the frequencies, $f_{SWG_i^n}$, of sequential word groups. We keep sequential word groups including r-word(s) and their frequencies in different text files. These text files are basically arrays of word(s) and their frequencies.

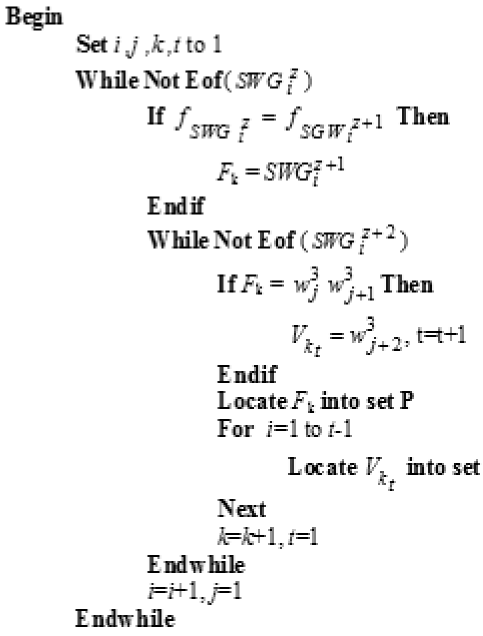
A feature of a product can only be a single word or it can be a noun phrase or an adjective phrase. For example, features of a car are color, fuel type, motor power and etc. For extracting features represented by two-words, we use three lists SWG^1, SWG^2 and SWG^3. We also use frequencies of every SWG, $f_{SWG_i^n}$, for comparison of frequencies of different SWGs.

Let us consider that first n unique SWGs of SWG^1 which have the highest frequencies are the features of the product P. Empirically n can be set to an integer number between 50 and 100. We compare frequency of first n unique *SWG of* SWG^1 and frequencies of *SWG of* SWG^2 beginning with the same word. If they are approximately equal, the feature is represented by two-words instead of single word. Formally, if $f_{SWG_i^1} = f_{SWG_j^2} + t$ then $F_{\mathrm{k}} = SWG_j^2 = w_j^2 \; w_{j+1}^2$ where F_{k} shows a feature of product $P = \{F_1 , \dots , F_{\mathrm{n}} \}$, $SWG_j^2 = SWG_i^1 \; w_{j+1}^2 = w_j^2 \; w_{j+1}^2$ and t is an integer number to show that frequencies are approximately equal.

After selecting a feature, it is time to find the values of this feature F_{k}. For every SWG from SWG^3, if the first two words of SWG are equal to the feature, $F_{\mathrm{k},}$ then the third word that comes after these two words is set to a value of this feature, F_{k}. Formally, If $SWG_j^3 = F_{\mathrm{k}} \; w_{j+2}^3$ then $V_{k_t} = w_{j+2}^3$ where $F_{\mathrm{k}} = w_j^3 \; w_{j+1}^3$ and V_{k_t} is one of the values of feature F_{k} , $V_{k_t} \in F=\{V_1 , V_2, V_3 ,,, V_{\mathrm{n}} \}$. This process continues until all features that include two-words and values of these features are found.

To find the features including one-word, again we take into account of n unique SWG of SWG^1 which have the highest frequencies. We eliminate the *SWGs* which are used for any feature including two words or used for a value of these features. Formally, if $SWG_i^1 \notin P$ or $SWG_i^1 \notin F$ then $F_{\mathrm{k}} = SWG_i^1$ where F_{k} becomes a feature of product P. Values of this feature F_{k} becomes the second words of SWG^2, beginning with the feature F_{k}. Formally, if $SWG_j^2 = F_k \; w_{j+1}^2$ then $V_{k_t} = w_{j+1}^2$ where $F_{\mathrm{k}} = w_j^2$ and V_{k_t} is one of the values of feature F_{k} ,

$V_{k_t} \in F$. This process continues until to all features that include one-word phrases and values of these features are found. A description of a product *P*, $P = \{ F_1 = \{V_{1_1},...,V_{1_k}\},...,F_n = \{V_{n_1},...,V_{n_t}\} \}$, is completed ignoring features including more than two-words. Figure 1 shows how to find the features including more than two words and values of this feature.



```
Begin
    Set $i, j, k, t$ to 1
    While Not Eof($SWG_i^z$)
        If $f_{SWG_i^z} = f_{SGW_i^{z+1}}$ Then
            $F_k = SWG_i^{z+1}$
        Endif
        While Not Eof ($SWG_i^{z+2}$)
            If $F_k = w_j^3\ w_{j+1}^3$ Then
                $V_{k_t} = w_{j+2}^3$, t=t+1
            Endif
            Locate $F_k$ into set P
            For $i$=1 to $t$-1
                Locate $V_{k_t}$ into set $F_k$
            Next
            $k$=$k$+1, $t$=1
        Endwhile
        $i$=$i$+1, $j$=1
    Endwhile
```

Figure 1: Pseudo code of finding features including more than two words and values of these features.

2.2. Defining Mostly Used Keywords to Refine Features and Values

Features and values of a product can be refined by using the set of keywords, $K=\{k_1, k_2, \ldots, k_n\}$, of html pages where k_i appears in meta-keyword-tag of html page. $D = [d_{ij}]_{mxn}$ is a binary matrix shows which web pages use the keyword, k_i , in meta-keyword-tag. *m* shows number of keywords and *n* shows number of web pages. A keyword may include more than one word. Value of each entry of matrix *D* can be computed by the following conditions.

$$d_{ij} = \begin{cases} 1 \ , \text{ if } i^{\text{th}} \text{ keyword occurs in } j^{\text{th}} \text{ document} \\ 0 \ , \text{ if } i^{\text{th}} \text{ keyword doesn't occur in } j^{\text{th}} \text{ document} \end{cases}$$

For example, in the following *D* matrix, d_{21} has a value of 1 and means that second keyword of set *K* is the keyword of the first web page, formally, $k_2 \in wp_1$.

$$D = \begin{bmatrix} 0 & 1 & 1 & ... & 0 \\ 1 & 1 & 1 & ... & 1 \\ 1 & 0 & 0 & ... & 1 \\ ... & ... & ... & ... & . \\ ... & ... & ... & ... & . \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}_{mxn}$$

Number of 1s in a row gives the frequency, f_{k_i}, of i^{th} keyword, k_i, used in web pages, *wp*. Frequency of a keyword can be computed as $f_{k_i} = \sum_{j=1}^{n} d_{ij}$.

All frequencies of keywords are computed and then keywords are eliminated from the set *K* if they have a frequency under the standard deviation of the average of keyword frequencies. A new set which is the subset of *K* can be shown as $K' = \{ k_i \mid \text{if } f_{k_i} > (A - \sigma) \}$ where *A* is the average of the frequencies of keywords and σ is the standard deviation of the frequencies of keywords. *A* is represented by the formula

$$A = \frac{\sum_{i=1}^{m} f_{k_i}}{m}$$

Members of the set *P* are compared with the members of set K'. The words in the intersection set of these sets define the importance of the features and values. If $k_i \in K'$ and $k_i \in P$ then k_i is certainly a feature of product *P*. If $k_i \in K'$ and $k_i \in F$ then k_i is certainly a value of feature *F*.

3. ARCHITECTURE OF FERBOT

To demonstrate that sequential word group frequencies are effectively used for information extraction we developed a software called FERbot (Feature Extractor Recognizer Bot). FERbot classifies features and their values into different categories using the algorithm specified in Figure 1 which is based on sequential word frequencies. There are three main parts of FERbot system as shown in Figure 2, where IG represents the Information Gathering part, UI represents the User Interface part and KM represents the Knowledge Modelling part.

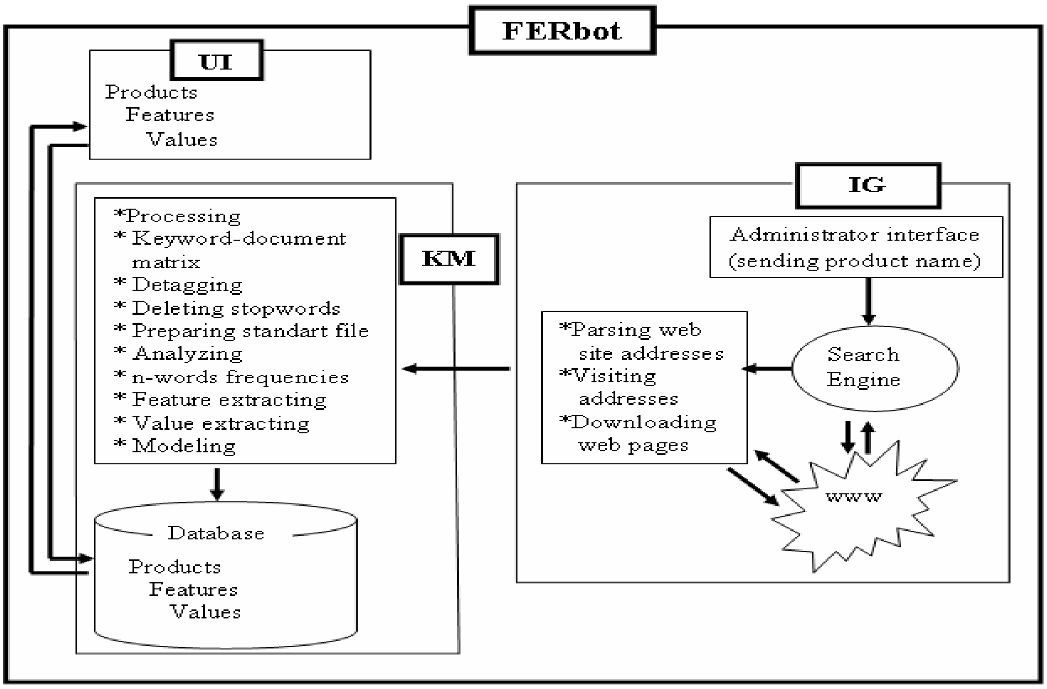


Figure 2: Architecture of FERbot: IG (Information Gathering), KM (Knowledge Modeling), UI (User Interface).

At the Information Gathering part, FERbot visits internet pages and stores them into a local machine to work over these web pages. The main questions for IG are "Which sites must be visited?" and "Which information must be collected from each site?" Everyday the number of the web sites to be visited is increasing. Processing every document in WWW is often expensive and unnecessary. Chuang and Chien (2005) fight with this huge amount of data by combining their study with search process of search engines to extract features from the retrieved highly ranked search-result snippets for each text segment. Their aim is to organize text segments into hierarchical structure of topic classes. Can (2006) summarizes the information explosion problem and illustrates some recent developments for information retrieval research of Turkish Language. He emphasizes some methods like vector space models, term frequency, inverse document frequency and document query matching for more effective information retrieval against the information explosion. FERbot knows which page to visit by filtering the addresses of web pages from Google's main page. To obtain more specific result page, FERbot sends the product name as query to Google Search engine by customizing some searching options. To visit just e-commerce sites that sell products, web site addresses including domain name extensions like edu, mil, org, net, gov, aero, biz, coop, info, pro and files having file extension names like pdf, doc, ppt, xls, jpg, jpeg, gif, pict, bmp, tif, png, ico are eliminated from the result page.

Google applies Latent Semantic Index (LSI) to every query. LSI can be used to find relevant documents that may not even share any search terms provided by the user (Berry, Dumais and O'Brien, 1995). Google search results include both the product names and the relevant terms of product names. Every search-result of Google includes the web site name, web site address and a small summary including the queried word(s). Google uses different styles to insert hyperlinks for web site names, addresses and summaries. To leave alone the links showing the web site address of the query, Google's search results are parsed. Parsed hyperlinks of web site addresses are written into a text file to keep them for reuse. Every link includes an identifier showing its order. The identifier can be considered as the pointer of that link. Sometimes Google produces less than 1000 results. At that situation the highest numbered pointer is the total results. Also it is possible to see same addresses at the result page. After downloading links, same web site addresses are deleted.

According to Google Search Engine results, most popular 1000 links are refined and ready to be visited. Since FERbot is a centralized system, it begins to download web sites from the first link to the last one. There are some reasons that prevent FERbot to download web pages. First, some web pages are protected for downloading. Second, some web pages are not available and third, some downloadings are not completed. At these situations FERbot skips the web page and downloads the next web page.

At the Knowledge Modelling part which is the most important, FERbot parses these downloaded html files to have a structured form for data. After downloading all web pages, to find the frequencies of sequential word groups of the collected data, FERbot applies a regular process. All html files have to be detagged. FERbot cleans all java, php, asp and etc. scripts, and deletes all markups of html codes. FERbot writes all text without markups and scripts into a text file having word-space-word format. Cleaning mark ups of html files is not the only thing to get a standard file to work over it. FERbot changes all letters into lower case in a document to make that document be more standardized. Also, FERbot converts all text formats into a same text format and deletes all punctuation marks. There are some stopwords in a language to link sentences or phrases. For example in Turkish language they are "için, kim, bu" and etc. Cleaning these stopwords decreases the work on analysing whole data. FERbot cleans approximately 180 types of stopwords of Turkish language from documents. Finally, FERbot use the algorithm based on sequential word group frequencies and keyword-document matrix.

At the user interface part, FERbot allows customers to present their queries. Customers may select the query from the list or they may write their queries. After accepting query, related answer is retrieved from database and sent back to user.

3.1. Case Studies of Sequential Word Group Frequencies for Information Extraction

FERbot collects the data from the web to extract the features of the product before presenting it to the customers. FERbot applies a regular process (collecting information, knowledge modeling and presenting data) for every product. Let us illustrate the description for the product "araba" (which means car in Turkish) that FERbot retrieves.

The first step of FERbot is to find the web pages related to "araba" from the Internet using Google search engine. Every search-result of Google includes the web site name, web site address and a small summary including the queried word "araba". Google uses different styles to insert hyperlinks for web site names, addresses and summaries. To leave alone the links showing the web site address of the query "araba", Google's search results are parsed. Hyperlinks of web page addresses about "araba" are written into a text file to visit and download them one by one.

FERbot creates a collection which consists of 925 web pages related to "araba", to find the frequencies of sequential n-word(s) groups of these web pages. FERbot applies the regular process which is mentioned in section 2 and section 3. After this process, FERbot creates an input file called S which is the list of words obtained from collected web pages.

FERbot analyzes the input file and writes all one-word, two-words, six-words groups (for example a two-words group means two sequential words considering order is important) and their frequencies into separate text files. Sections of three of these files about "araba" are presented in Table 1.

Table1. Sections of SWG^1,SWG^2,SWG^3 and their frequencies, f_{SWG^1}, f_{SWG^2}, f_{SWG^3}

A Section from the file of SWG^1 and f_{SWG^1}		A Section from the file of SWG^2 and f_{SWG^2}		A Section from the file of SWG^3 and f_{SWG^3}	
SWG_i^1	$f_{SWG_i^1}$	SWG_i^2	$f_{SWG_i^2}$	SWG_i^3	$f_{SWG_i^3}$
motor	2358	motor gücü	2214	motor gücü 100	163
motora	1	motor i6	1	motor gücü 101	16
motorbike	2	motor ima	3	motor gücü 102	13
motorbisiklet	4	motor kategoride	1	motor gücü 103	15
………	……	………	……	………	……

Table 2. Values and features of product “araba”

FEATURE	VALUE
'ikinci el' (second hand)	'araç'(vehicle), 'araba'(car), 'arabalar'(cars), 'bilgisayar' (computer), 'cep'(mobile), 'ford' (ford), 'ikinci' (second), 'mercedes' (mercedes), 'opel'(opel), 'oto'(auto), 'otomobil' (automobile), 'otomotiv' (automotive), 'peugeot' (peugeot), 'renault' (renault), 'satılık'(for sale)
'motor gücü' (motor power)	'100' '101' '105' '110' '115' '120' '125' '130' '136' '140' '150' '163' '180' '190' '200' '220' '225' '235' '250' '300' '60' '65' '70' '75'
'yakıt tipi' (fuel type)	'benzin'(benzine) 'dizel'(diesel) 'lpg'(lpg)
'silindir hacmi' (cylinder volume)	'1300' '1400' '1490' '1500' '1598' '1600' '1800' '1998' '2000' '2400' '2500' '2700' '2800' 'cm'
'yılı' (production date)	'1963' '1964' '1973' '1987' '1989' '1990' '1991' '1992' '1993' '1994' '1995' '1996' '1997' '1998' '1999' '2000' '2001' '2002' '2003' '2004' '2005' '2006'
'renk' (color)	'çelik'(steel), 'ateş'(fire), 'bej'(beige), 'beyaz'(white), 'bordo' (maroon), 'füme'(smoke-colored), 'gökyüzü'(sky), 'gümüş' (silver), 'gri'(grey), 'kırmızı'(red), 'lacivert'(dark blue), 'mavi' (blue), 'metalik'(metallic), 'nil'(Nile green), 'sarı'(yellow), 'siyah' (black), 'yeşil'(green)
'ytl' (currency unit)	'1' '10' '11' '12' '13' '14' '15"16' '17' '18' '19' '20' '21' '22' '23"24' '25' '26 '27' '28' '3' '30' '4' '5'

FERbot analyzes the input file and writes all one-word, two-words, six-words groups (for example a two-words group means two sequential words considering order is important) and their frequencies into separate text files. Sections of three of these files about “araba” are presented in Table 1.

FERbot considers the first sixty unique words which have highest frequency as the most related words for product “araba”. It searches relevant features among these sixty words (to get better solutions number sixty can be increased). FERbot extracts the features which includes two-words by using the algorithm mentioned in Figure 2. For an example, the frequency of motor, f SWG_i^1, and the frequency of two-words group beginning with the word “motor”, f SWG_i^2, are compared, if they are approximately equal, feature name becomes two-words group (motor gücü) instead of “motor”. The frequency of “motor gücü”, f SWG_i^2, and the frequency of three-words group beginning with “motor gücü” , f SWG_i^3, are compared, if they are not equal or not approximately equal, feature name remains same as “motor gücü” and values of this feature becomes last words (..,100,101,102,103,…) of three-words group, SWG_i^3. Words having high frequencies that do not take a place in the “two-words feature group” are considered as single word feature and values of them were associated like values of two-words groups.

FERbot searches meta-keyword-tags to see the relation between the product name “araba” and keywords of web pages about “araba” as mentioned in Section 2. FERbot considers keywords having a frequency higher than the average of keyword frequencies are potential features or values of the product “araba”. These potential features and values are compared with the results that are found by the new algorithm to refine the results.

The refined results about features and values of “araba” are presented in Table 2 where first column represents the main features of the product “araba” and second column represents the values of the features. Also frequencies of these features and values are kept in another file. By changing the threshold of any frequency, number of features and the range of the features (number of values) can be increased or decreased.

To show the performance of FERbot, the features and values of the concept “Renault” and “BMW” which are different types of product “araba” (car) are presented in the Table 3 and Table 4. These results are obtained for the same thresholds from independent domain resources. We used a summary of the same features in both tables for better comparisons. “Yakıt tipi” which means “fuel type” is the feature of both cars that have different values. This is the expected situation for all cars. We know that “BMW” has more powerful motors and higher cylinder volumes than Renault has. This situation is observed for the values of features “motor power” and “cylinder volume” as shown in Table 3 and Table 4. Renk (Color) features of these two kinds of cars have almost the same values but the frequencies may change. FERbot may extract meaningful information about the relations between different products which are in the same category as seen in Table 3 and Table 4. For example: both car types have the same values for the feature “yakıt tipi”. For this reason these products are strongly related. Also, FERbot may show the range of the values of these products at the same time.

Table 3. Values and features of product “Renault”.

FEATURE	VALUE
motor gücü	100, 103, 105, 110, 115, 120, 135, 140, 60, 65, 70, 72, 75, 80, 85, 90, 95, 98
yakıt tipi	benzin, dizel, lpg
Silindir hacmi	1149, 1200, 1300, 1390, 1397, 1398, 1400, 1461, 1490, 1500, 1590, 1598, 1600, 1700, 1721, 1870, 1898, 1900, 1995, 1998, 2000, 2500
Renk	çelik, ateş, bej, beyaz,bordo, fume, gümüş, gri, kırmızı, lacivert, mavi, sarı

We evaluated our approach for some different types of products such as, BOOK, TELEVISION, HOLIDAY, VILLA, PRIVATE LESSON etc. to compare results. We evaluate our approach for independent domains. Whenever we use domain specific web pages for searching the features and values of a product, the results becomes excellent like the results of “villa” from the domain http://www.emlak.turyap.com or “özel ders” from the domain http://izmir.ozelders.com/ogretmen/. Results of the products “villa” and “özel ders” can be seen in Table 5, Table 6 and Table 7. Also, if the domain of any product is semistructured then our approach produces perfect results.

Table 4. Values and features of product “BMW”.

FEATURE	VALUE
motor gücü	100, 101, 105, 110, 115, 120, 125, 130, 136, 140, 163, 180, 190, 200, 220, 225, 235, 250, 300, 60,65
yakıt tipi	benzin, dizel, lpg
silindir hacmi	1596, 1598, 1600, 1796, 1798, 1800, 1991, 1995, 1998, 2000, 2200, 2498, 2500, 2800, 2993, 2998, 3000, 4400, 4500
renk	çelik, ateş, beyaz, bordo, füme, gümüş, gri, il, kırmızı, lacivert, mavi, metalik, opsiyonları, seçenekleri, siyah, yeşil

Table 5. Values and features of product “villa”.

FEATURE	VALUE
'oda sayısı' (number of rooms)	'2' '3' '4' '5' '6'
'salon sayısı' (number of salons)	'1' '2'
'kiralık' (for rent)	'emlak' (estate) 'emlaklar' (estates) 'villa' (villa)
'site' (site)	'haritası' (map) 'içinde' (inside)
'kat' (number of floors)	'1' '2' '3' '4'
'bahçe' (garden)	'içinde' (inside) 'var' (exist)
'bölge' (region)	'çengelköy' 'bolu' 'guzelyali' 'marmaris'
'mahalle' (district)	'akpinar' 'bahcelievler' 'merkez'

Table 6. Values and features of product “villa” with a different treshold.

FEATURE	VALUE
'oda sayısı' (number of rooms)	'2' '3' '4' '5' '6'
'salon sayısı' (number of salons)	'1' '2'
'kiralık' (for rent)	'emlak' (estate) 'emlaklar' (estates) 'villa' (villa)
'adres' (address)	
'fiyat' (cost)	
'ilçe' (town)	
'kat' (number of floors)	'1' '2' '3' '4'
'mutfak' (kitchen)	
'banyo' (bathroom)	
'bahçe' (garden)	'var' (exist)
'mahalle' (district)	'Merkez'

Table 5 and Table 6 are obtained for different thresholds of frequencies of values of features. When we decrease the threshold of frequency we obtain the results in Table 6 which are better feature values than results in Table 5.

Table 7 shows another case study of FER-bot that is studied for the product “özel ders” which means “private course” in English. FER-bot uses web pages only from the domain http://izmir.ozelders.com/ogretmen/. Results are quite satisfactory.

Table 7. Values and features of product “özel ders”.

'lisans'(university)	'anadolu' 'celal bayar 'ege' '9 eylül' 'istanbul' 'izmir' 'uludağ'
'yaş'(age)	20' '21' '22' '23' '24' '25' '26' '27' '28' '29' '32' '33' …….
'cinsiyet'(gender)	'bay'(male) 'bayan'(female)
'isim'(name)	'elif' 'emrah' 'erkan' 'pınar'………
'meslek'(job)	'öğrenci'(student) 'öğretmen'(teacher)
'eğitim durumu'(education)	'üniversite'(university) 'lise'(high school)
'grup dersi'(group lesson)	'evet'(yes) 'hayır' (no)
'saat ücreti' (cost for an hour)	'20' '25' '30' '40'
'ders mekanı'(place of lesson)	'öğrencinin evi' (house of student) 'etüd'(etude) 'grup'(group)
'bulunduğu şehir' (city of teacher)	'bornova' 'buca' 'karşıyaka'

3.2. Comparison of results with similar works

Information agents generally rely on wrappers to extract information from semi structured web pages (a document is semi structured if the location of the relevant information can be described based on a concise). Each wrapper consists of a set of extraction rules and the code required to apply those rules (Muslea, 1999). According to Seo, Yang and Choi (2001), the information extraction systems usually rely on extraction rules tailored to a particular information source, generally called wrappers, in order to cope with structural heterogeneity inherent in many different sources. They define a wrapper as a program or a rule that understands information provided by a specific source and translates it into a regular form that can be reused by other agents.

Table 8. Existences of features of "araba" for different wrappers and FERbot.

Feature	**Wrapper for gittigidiyor.com**	**Wrapper for sahibinden.com**	**Wrapper for araba.com**	**FERbot**
Marka	1	1	0	0
Model Yılı	1	1	1	1
Renk	1	1	1	1
Kilometre	1	1	1	0
Motor Hacmi	1	1	1	1
Yakıt Tipi	1	1	1	1
Motor Gücü	1	1	1	1
Model	1	0	0	0
Model Uzantısı	1	0	0	0
Araç Tipi	1	0	0	0
İç Renk	1	0	0	0
Durumu	1	1	0	0
Vites	0	1	1	0
Kasa Tipi	0	1	0	0
4 x 4	0	1	0	0
Durumu	0	1	0	0
Türü	0	1	0	0
Takaslı	0	1	0	0
Kimden	0	1	0	0

FERbot is a kind of information extraction system which has the same purpose as wrappers have. The first advantage of FERbot is its own algorithm. It can be used for domain specific or domain independent resources. Second advantage is the ability to work with the labeled or unlabeled web pages. The input of FERbot can be any kind of web pages (They are designed or not designed by XML). FERbot understands the features of products by finding their frequencies. FERbot uses labels to extract the keywords of web pages to construct the keyword-document matrix for refining the results of features of products. Finally, we may say that most of the wrappers are constructed for domain dependent and labeled documents but FERbot does not imply these kinds of restrictions.

In Table 8, first column shows the features of "araba" that are obtained as a result of construction of wrappers for different domains. 1 means that the related feature is recognized by the wrapper of that domain or FERbot. 0 means that the related feature is not recognized by that wrapper or FERbot. FERbot generates the features from different domains.

Table 9 shows the correlation matrix of wrappers and FERbot. As seen in Table 9, FERbot features have a strong positive relation with features of wrapper for araba.com. Also, Results of FERbot are mostly related features with the features of wrapper for gittigidiyor.com.

In Table 10, FERbot generates the features from the domains izmir.ozelders.com/ogretmen and emlak.turyap.com.tr. When the features are compared it is seen that FERbot automatically generates most of the features that are captured by the domain dependent wrappers.

Table 9. Correlation matrix of wrappers and FERbot.

	Wrapper for gittigidiyor.com	**Wrapper for sahibinden.com**	**Wrapper for araba.com**	**FERbot**
Wrapper for gittigidiyor.com	1			
Wrapper for sahibinden.com	-0.394405319	1		
Wrapper for araba.com	0.357142857	0.394405319	1	
FERbot	0.456435465	0.3086067	0.782460796	1

Table 10. Domain dependent Features for wrappers and FERbot.

Wrapper for izmir.ozelders.com/ogretmen	**FERbot**		**Wrapper for emlak.turyap.com.tr**	**FERbot**
isim:	isim		oda sayısı	'oda sayısı'
yaş:	yaş		salon sayısı	'salon sayısı'
cinsiyet:	cinsiyet		mahalle	'mahalle'
meslek:	meslek		bahçe	'bahçe'
saat ücreti:	saat ücreti		ilçe	'ilçe'
grup dersi:	grup dersi		fiyat	'fiyat'
bulunduğu şehir:	'bulunduğu şehir		mutfak	'mutfak'
ders mekanı:	ders mekanı		banyo	'banyo'
irtibat telefonu:	lisans		toplam kat	'kat'
verdiği dersler	eğitim durumu		site içinde	'site'
			alan	'bölge'
			ısıtma	'kiralık'
			kod	'adres'
			güvenlik	
			yapım yılı	
			balkon sayısı	
Consistency: 80%			**Consistency: 72%**	

4. CONCLUSIONS

In this study, we proposed a new algorithm based on information extraction and knowledge discovery from e-commerce web pages. We developed a software called FERbot (Feature Extractor Recognizer bot) to evaluate this new algorithm. FERbot learns the features of products automatically by extracting sequential word groups from hundreds of web pages and finds out the values of these features to automatically construct a knowledge base for products. Finally, in section 3 we presented the results of experiments about products “araba”, “Renault”, “BMW”, “villa” and “özel ders”. Our algorithm produces perfect results if the input set is mostly consist of semi-structured web pages instead of unstructured web pages. Comparisons of results of FERbot and other wrappers are defined at the end of Section 3.

This work has shown that sequential word group (single, two-words,…., *n*-words) frequencies are effective to derive semantics from e-commerce sites to define product features and feature values. We demonstrated that structure and semantics of a text can be systematically extracted by lexical analysis. We presented a new algorithm to use the relative frequencies of sequential word groups for information extraction. This algorithm is an alternative for the current statistical and machine learning techniques for information detection and classification. Also, text miners can use our algorithm for knowledge discovery within the concept of statistical data mining techniques to operate the information extracted from texts.

If intelligent agents especially “shop-bots that derive semantics from semantic web sites” have capability to understand the non-standard web sites, they will complete their job to find the product features at the same time. Such agents may suggest new product features as important keywords to be taken into account for refined searches. If proposed new features are accepted by the user, the agent will search and shop in Internet accordingly. These agents searching the desired product in the Internet, may record possible new features at the same time and inform the central server from which they are dispatched and where the product feature files are maintained.

Our algorithm and work in this paper for extracting new features of products from e-shopping web-sites will be realized as an agent based application. One part of e-commerce agents deal with the subject of information gathering which is the act of collecting information. This is the main idea of our study which is collecting information from Web and extracting meaningful relations between products and their properties to inform customers. This will also present new possibilities towards information extracting from structured or unstructured html web sites.

Most of the wrappers are constructed for domain specific resources. For that reason, they get the semi structured or structured web pages as input set to extract the information. They mainly segment the page layout and extract the labeled information. We have searched how FERbot produces different results according to the type of resource of web pages. FERbot is ready for every kind of web pages but, our algorithm produces perfect results if the input set is mostly full of semi-structured web pages instead of unstructured web pages.

5. ACKNOWLEDGMENTS

We would like to thank Dr. Adil Alpkocak for his useful comments and remarks.

REFERENCES

Ahmed S, T., Vadrevu S., Davulcu H. (2006). DataRover: An Automated System for Extracting Product Information, In Advances in Web Intelligence and Data Mining, Volume 23, pp. 1-10, Springer Berlin/Heidelberg Publishers.

Arasu A., Garcia-Molina H. (2003). Extracting structured data from Web pages, ACM SIGMOD, pp. 337 – 348.

Arasu A., Garcia-Molina H. (2003). Extracting structured data from Web pages, ACM SIGMOD, pp. 337 – 348.

Bikel D. M. Schwartz R., Weischedel R. M., (1999). An algorithm that learns what's in a name. *Machine Learning*. 34, 211-231.

Borkar V. Deshmukh K., Sarawagi S., (2001). Automatic segmentation of text into structured records, International Conference on Management of Data Archive Proceedings of the 2001 ACM SIGMOD international conference on Management of data, ISSN:0163-5808, pp. 175 – 186.

Burget R. (2004). Hierarchies in HTML Documents: Linking Text to Concepts, 15th International Workshop on Database and Expert Systems Applications, pp.186-190.

Burget R. (2005). Visual HTML Document Modeling for Information Extraction, CZ, FEI VŠB, ISBN 80-248-0864-1, Ostrava pp.17-24.

Can F. (2006). Turkish Information Retrieval: Past Changes Future, Advances in Information Systems, Springer Berlin Heidelberg New York, pp.13-22.

Chaudhury A. Kilboer J., (2002). E-business and E-commerce Infrastructure, Technologies Supporting the E-business Initiative (Chapter 1), McGraw Hill.

Chuang S. L., Chien L.F. (2005). Taxonomy generation for text segments:A practical web-based approach, ACM Transactions on Information Systems, pp. 363-396.

Craven M., DiPasquo D., Freitag D., McCallum A., Mitchell T., Nigam K., Slattery S. (2000). Learning to construct knowledge bases from the World Wide Web, Artificial Intelligence, 118, pp. 69–113.

Crescenzi R., Mecca G., Merialdo P. (2001). RoadRunner: Towards Automatic Data Extraction from Large Web Sites, 27th International Conference on Very Large Data Bases.

Filatova E., Hatzivassiloglou V. (2003). Domain-Independent Detection, Extraction, and Labeling of Atomic Events, Proceedings of the Fourth International Conference on Recent Advances in Natural Language Processing.

Gomes G., Segami C. (2007). Semantic Interpretation and Knowledge Extraction, Knowledge-Based Systems, Volume 20, Issue 1, pp. 51-60.

Grishman R. (1997). Information Extraction: Techniques and Challenges, Materials for Information Extraction, Springer-Verlag.

Hui B., Yu E. (2005). Extracting conceptual relationships from specialized documents. *Data & Knowledge Engineering*. 54 (1), 29-55.

Karanikas H., Theodoulidis B. (2002). Knowledge Discovery in Text and Text Mining Software, Technical report, UMIST-CRIM, Manchester.

Karataş K. (2001). The Anatomy of an Internet Robot : bilBot®.. Meteksan Sistem ve Bilgisayar Teknolojileri A.Ş. Bilkent, Ankara, Türkiye'de Internet konferansları VII., http://inet tr.org.tr/inetconf7/program/ 97.html.

Lee W.P. (2004). Applying domain knowledge and social information to product analysis and recommendations: an agent-based decision support system. *Expert systems*. 21 (3), 138-148.

Leung H., He M. (2002). Agents in E-commerce: State of the Art, Knowledge and Information Systems, pp. 257-282.

Moens M.F. (2006). Information Extraction: Algorithms and Prospects in a Retrieval Context, (Chapter 1), pp. 1-22, Springer.

Mukherjee S. Yang G., Tan W., Ramakrishnan I.V., (2003). Automatic Discovery of Semantic Structures in HTML Documents, 7th International Conference on Document Analysis and Recognition.

Muslea I. (1999). Extraction Patterns for Information Extraction Tasks: A Survey, The AAAI-99, Workshop on Machine Learning for Information Extraction.

Seo H., Yang J., Choi J. (2001). Building Intelligent Systems for Mining Information Extraction Rules from Web Pages by Using Domain Knowledge, IEEE International Symposium on Industrial Electronics, pp. 322-327, Pusan, Korea.

Shamsfard M., Bardoroush A.A. (2004). Learning Ontologies from Natural Language Texts. *International Journal of Human-Computer Studies*. 60 (1) 17-63.

Stevenson M. (2006). Fact distribution in Information Extraction, Language Resources and Evaluation, Springer, pp. 183-201.

Ferkan KAPLANSEREN, Lisans eğitimini Ege Üniversitesi Matematik Bölümü'nde, yüksek lisans ve doktora eğitimini ise Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü'nde tamamlamıştır. Çalışmalarını akıllı ajanlar, elektronik ticaret ve yapay zeka üzerine yoğunlaştırmıştır. D.E.Ü. İşletme Fakültesi'nde öğretim görevlisi olarak görev yapmaktadır.

TatyanaYAKHNO, Novosibirsk State University, Computational Linguistic bölümü mezunudur. Aynı bölümde yüksek lisansını tamamlamıştır. Doktora eğitimini "Siberian Branch of Russian Academy of Sciences" akedemisi bilgisayar merkezinde tamamlamıştır. Çalışma alanları arasında yapay zeka, bilgi gösterimi ve uzman sistemler, mantıksal programlama, kısıtsal programlama, çoklu ajan sistemleri ve evrimsel algoritmalar konuları yer alır. D.E.Ü. Bilgisayar Mühendisliği Bölümü'nde Prof. Dr. olarak görev yapmaktadır.

M. Kemal ŞİŞ, İstanbul Teknik Üniversitesi Elektrik Mühendisliği Bölümü'nde lisans ve yüksek lisans eğitimini tamamlamıştır. Doktora eğitimini Polytechnic University (NewYork A.B.D.) de bitirmiştir.Çalıştığı konular arasında elektronik ticaret, sayısal ses işleme, bilgisayar ağları ve quantum hesaplamaları yer alır. D.E.Ü. Bilgisayar Mühendisliği Bölümü'nde Öğr. Gör. Dr. olarak görev yapmaktadır.