Implementation of An Information Extraction System using Boolean Model

Khaing Zar Mon, Thinn Thu Naing
University of Computer Studies, Yangon
khaingzarmon09@gmail.com

Abstract

The rapid growth of on-line information has created a surge of interest in tools that are able to extract information from on-line documents. Information extraction (IE) is the process of pulling desired pieces of information out of a document. IE systems are complementary to Information Retrieval (IR) systems and can make information retrieval more precise. This system intends to implement an information extraction system using Boolean Model and it will display the relevant information according to user's preference. Boolean model was the first operational Information Retrieval model. It has the advantages of ease and efficiency of the implementation, simplicity of the inverted file structure, and clarity of the model. In Boolean Model, user's request is converted to Boolean algebra (i.e., 0 and 1) and Boolean queries give inclusion or exclusion of documents. As a case study, this system will implement based on sale system of computer and accessories.

1. Introduction

Information Extraction (IE) is the name given to any process which selectively structures and combines data which is found, explicitly stated or implied, in one or more texts [5]. Information retrieval (IR) techniques identify documents from a larger collection which are (hopefully) relevant with respect to some query. Information extraction (IE) techniques process a document to identify pre-specified entities and the relationships between them. Put another way, IR retrieves relevant documents from collections, and IE extracts relevant information from documents. The two techniques are therefore complementary, and their use in combination has the potential to create a powerful tool in text processing, allowing, for example, the automated construction of repositories of structured information from large free-text collections.

Information (or document) retrieval systems deal with the representation, organization, and accessing of information items, documents or representatives of documents. IR identifies documents which match a query as presented to the system and which may, or may not, contain the desired information. Boolean model is one of the approaches that use in IR. A Boolean query is constructed from atomic query terms (words or phrases) using the logical operators AND, OR and NOT [11].

With the rapid expansion of the internet, there is no need to go to the shopping center or mall, and showroom for buying computer or searching information about the computer. As the role of internet is increased, the use of on-line shopping in everyday life becomes very popular. This system aims to help the user in saving time, energy and money. This system uses Boolean model equations to retrieve the laptop computer information that is relevant to the users' preferences. In this system, the buyer can choose the product to meet his/her price and can also see another type of product from different sellers.

The rest of the paper is organized as follows: the background history of information extraction and information retrieval is described in the next section. Section 3 represents the concept of Boolean Model and the architecture of the proposed system is described in section 4. And then, section 5 describes the evaluation of the proposed system. Finally, the conclusion of this paper and references are described in section 6 and 7 respectively.

2. History of Information Extraction and Information Retrieval

2.1. Information Extraction

Information extraction (IE) is the process of pulling desired pieces of information out of a document [13]. IE technology automatically skims through text and identifies predefined types of information. An IE application is usually configured with the information type, and often location in the text, so the information then becomes available for further processing, human or automatic. A typical use of Information Extraction is to "extract" this information and put it in a database format. And Information Extraction (IE) is also used today to support and enhance other kinds of text handling applications such as Information Retrieval, Question

Answering, Text Categorization, and even Machine Translation of Natural Languages.

Information Extraction technology recognizes predefined types of information in text, without the specific instances of those types having to be defined in advance of time. Systems today use one of three strategies for finding the specified information in text – human developed patterns to find key information, based on grammatical structures and domain expressions, patterns of key information automatically learned from manually tagged examples in the text, and hybrid systems. Most current systems are in fact based almost entirely on human developed patterns. Some of these are experimenting with automated learning in combination with manually developed patterns and can be considered hybrid. On the whole, IE technology has reached levels of accuracy and adaptability to new tasks that make it useful in operational settings [12].

IE systems are complementary to IR systems. They analyze pre-selected but unrestricted texts in order to identify pre-specified entities, events, and relationships, i.e., it acts as does the "user" of an IR system in identifying the required information within a retrieved document. Information extraction is a non-trivial task as there are many ways of expressing the same fact, and in addition, information may be distributed across several sentences.

IE techniques are computationally intensive and therefore it is necessary to ensure as far as is possible that documents input to an IE system are likely to fall within the domain of the specific IE system. Thus, it is natural to think of ways in which an IR system can be used as a front-end to an IE system to retrieve, from the source collection, documents which are relevant to an extraction scenario. Coupling IR and IE in this way is not a novel idea [11].

2.2. Information Retrieval

The meaning of the term information retrieval (IR) can be very broad. Some defines Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers) [1]. And other defines IR is concerned with the process involved in the representation, storage, searching and finding of information which is relevant to a requirement for information desired by a human user [9]. In fact, the primary goal of an IR system is to retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible [10].

The field of IR also covers supporting users in browsing or filtering document collections or further processing a set of retrieved documents. And then IR identifies documents which match a query as presented to the system and which may, or may not, contain the desired information. Furthermore, IR systems can be classified into four basic models: Boolean model, probabilistic model, vector space model, and linguistic model [14]. This paper implements the role of Boolean Model in IR system and how they applied in IE system.

3. Boolean Model

Boolean model was the first operational Information Retrieval model. In this model, documents are indexed with a set of terms, and queries are terms combined with the logical operators AND, OR and NOT [6]. The result obtained from the processing of a query is a set of documents that totally match with it, i.e., only two possibilities are considered for each document: to be or not to be relevant for the user's needs, represented by the user query [8].

And furthermore, the conventional Boolean retrieval systems cannot provide ranked retrieval output in order of query-document similarity [2]. Such a ranked output is useful because controls are now available over the size of the retrieved document set, and iterative retrieval strategies based on successive query reformulations are simplified. Therefore, this paper uses the combination of Boolean Model and Extended Boolean Model to support ranking facility. The Extended Boolean Model utilizes document term weights reflecting the importance of individual terms in a document to calculate query-document similarities [7].

The advantages of the Boolean Model is that it has ease and efficiency of the implementation, simplicity of the inverted file structure, and clarity of the model [4].

Although the Boolean system has been popular in operational situations, the system has been criticized for several reasons: (1) the construction of a query formulation is too difficult for the user, (2) it does not support ranked output in any order of presumed importance to the user, (3) the size of the retrieved documents is difficult to control, and (4) there is no provision for term weighting either the documents or the queries [6].

4. The Architecture of the Proposed System

The electronic commerce is a major trend on the Web nowadays and one which has benefited millions of people. As the role of internet is increased, the people do not need to spend their time and energy for shopping. This system aims to build up the user preferences in on-line shopping.

In this system, the visited user is firstly checked by the system, whether he/she is member or not. If not, the user must be registered first. And the customer can ask for their necessities to the system. The system alters the user's request to Boolean algebra (i.e., 0 and 1), and match the request using Boolean Model equations. After matching, the relevant information is retrieved from the database with the order of the similarity result. And the final result is showed to the user. The overview of the system is shown in Figure 1 and detailed design is shown in Figure 2.

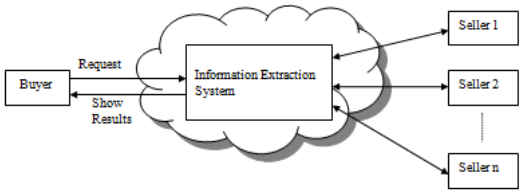

Figure 1: Overview of the proposed system

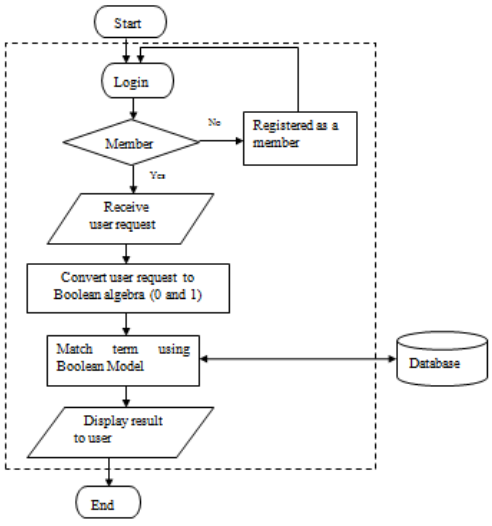

Figure 2: Detailed Design of the proposed system

5. Evaluation of the Proposed System

This section describes the calculation of the Boolean Model. And the sample database of laptop information is shown in Table 1.

In this system, the users need to enter the first query that is price and purpose of use. Assume the user enter the price between 600000 and 800000, and enter 'Premium' for purpose. The system retrieves laptop information from the database related to the user's first query. And then the user requests the message: "((Intel Core i3 OR Intel Core2 Duo) AND 2GB) AND (NOT Acer)" to the system. And the database collection consists of three documents (D=3) with the following content:

- **D_1:** Acer, Aspire TimeLine 4810T, Premium, Intel Core2 Solo, SU3500, 1.4GHz, 2GB, DDR3, 1066MHz, 320GB HDD,SATA,14"HD LED LCD, Gb LAN, Intel Wi-Fi Link 802.11b/g/n, Bluetooth, 6-Cells, DVD-RW DL, Webcam, 755000
- **D_2:** Compaq, PreSurio CQ41-203TU, Premium, Intel Core i3, 330M, 2.13GHz, 2GB, DDR3, 1066MHz, 320GB HDD, SATA, 14.1" WXGA 16:9 Wide, LAN, Intel Wi-Fi Link 802.11b/g/n, Bluetooth, 6-Cells, DVD-RW Lightscribe, Webcam, 600000
- **D_3:** SUZUKI, Kuiper 1412PKS, Premium, Intel Core2 Duo, T6500,2.1GHz,3GB, DDR2, 800MHz, 320GB HDD, SATA, 14.1" Widescreen, LAN, Intel Wi-Fi Link 802.11b/g/n, Bluetooth, 6-Cells, DVD-RW DL, 1.3M Webcam, 698700

The weight of each term is calculated by the following equations [3]:

$$W_{i,j} = tf_{norm\ i,j} * idf_{norm\ i}$$

where $tf_{norm\ i,j} = \frac{tf_{i,j}}{tf_{max\ i,j}}$ and $idf_{norm\ i} = \frac{idf_i}{idf_{max\ g}}$

$$D = 3;\ IDF = \log (D/df_i)$$

Where

$W_{i,j}$	= weight of term i in document j
$tf_{norm\ i,j}$	= normalized term frequency of term i in document j
$tf_{i,j}$	= term frequency of term i in document j
$tf_{max\ i,j}$	= maximum term frequency of term i in document j
idf_i	= inverse document frequency of term i in collection c
$idf_{max\ g}$	= maximum inverse document frequency of a generic term g in the collection c
$idf_{norm\ i}$	= normalized inverse document frequency of term i in the collection c

And the calculation of weight value of each term is shown in Table 2.

Table 1: Sample list of laptop computer in database

Type	Model	Purpose	Processor	Memory	Hard Disk	Display	Accessories	Price (Ks)
Acer	Aspire TimeLine 4810T	Premium	Intel Core2 Solo,SU3500, 1.4GHz	2GB,DDR3,1 066MHz	320GB HDD, SATA	14"HD LED LCD	Gb LAN,Intel Wi-Fi Link 802.11b/g/n,Bluetooth,6-Cells, DVD-RW DL,Webcam	755000
ASUS	F82Q	Innovator	Intel Core2 Duo,T5900, 2.2GHz	2GB,DDR2,8 00MHz	320GB HDD, SATA	14"HD LED	Gb LAN,Intel Wi-Fi Link 802.11b/g/n,Bluetooth,6-Cells, DVD-RW DL,1.3M Webcam	598000
Compaq	CQ45-401TX	Premium	Intel Core2 Duo,P8600, 2.4GHz	2GB,DDR2,8 00MHz	320GB HDD, SATA	14.1"HD WXGA	LAN,Wi-Fi,Bluetooth,6-Cells, DVD-RW LightScribe,Webcam	1199000
Compaq	PreSurio CQ41-203TU	Premium	Intel Core i3,330M,2.13GHz	2GB,DDR3,1 066MHz	320GB HDD, SATA	14.1" WXGA 16:9 Wide	LAN, Intel Wi-Fi Link 802.11b/g/n,Bluetooth,6-Cells, DVD-RW LightScribe,Webcam	600000
DELL	Vostro 1320	Premium	Intel Core2 Duo,P7570, 2.26GHz	4GB,DDR2, Null	320GB HDD, SATA	13.3" LED	LAN,Wi-Fi,Bluetooth,6-Cells, DVD-RW DL,Webcam	1090000
Gateway	NV 4802t	Innovator	Intel Core2 Duo,T6500, 2.1GHz	2GB,DDR2,6 67MHz	320GB HDD, Null	14"HD TFT LCD	Gb LAN,Intel Wi-Fi Link 802.11b/g/n,Bluetooth,6-Cells, DVD-RW DL,Webcam	945000
HP	G 60	Innovator	Intel Core2 Duo,T7300, 2.0GHz	4GB,DDR3, Null	320GB HDD, SATA	15.6" LED	LAN,Wi-Fi,Null,6-Cells, DVD-RW DL,Webcam	940000
SUZUKI	Kuiper 1412 PKS	Premium	Intel Core2 Duo,T6500, 2.1GHz	3GB,DDR2,8 00MHz	320GB HDD, SATA	14.1" Widescreen	LAN,Intel Wi-Fi Link 802.11b/g/n,Bluetooth,6-Cells, DVD-RW DL,1.3M Webcam	698700
...	...	...	...	...	...	...	...	...

Table 2: Calculation of the weight of documents

Terms	Q	Counts, $tf_{i,j}$			$tf_{norm\ i,j}$			df_i	D/df_i	idf_i	$idf_{norm\ i}$	$W_{i,j}$		
		D_1	D_2	D_3	D_1	D_2	D_3					D_1	D_2	D_3
600000	0	0	1	0	0	1	0	1	3/1=3	0.4771	1	0	1	0
698700	0	0	0	1	0	0	1	1	3/1=3	0.4771	1	0	0	1
755000	0	1	0	0	1	0	0	1	3/1=3	0.4771	1	1	0	0
2GB	1	1	1	0	1	1	0	2	3/2=1.5	0.1761	0.3691	0.3691	0.3691	0
320GB HDD	0	1	1	1	1	1	1	3	3/3=1	0	0	0	0	0
1066MHz	0	1	1	0	1	1	0	2	3/2=1.5	0.1761	0.3691	0.3691	0.3691	0
1.3M Webcam	0	0	0	1	0	0	1	1	3/1=3	0.4771	1	0	0	1
14.1" Widescreen	0	0	0	1	0	0	1	1	3/1=3	0.4771	1	0	0	1
Acer	1	1	0	0	1	0	0	1	3/1=3	0.4771	1	1	0	0
Compaq	0	0	1	0	0	1	0	1	3/1=3	0.4771	1	0	1	0
DVD-RW DL	0	1	0	1	1	0	1	2	3/2=1.5	0.1761	0.3691	0.3691	0	0.3691
Intel Core2 Duo	1	0	0	1	0	0	1	1	3/1=3	0.4771	1	0	0	1
Intel Core2 Solo	0	1	0	0	1	0	0	1	3/1=3	0.4771	1	1	0	0
Intel Core i3	1	0	1	0	0	1	0	1	3/1=3	0.4771	1	0	1	0
Intel Wi-Fi Link 802.11b/g/n	0	1	1	1	1	1	1	3	3/3=1	0	0	0	0	0
Kuiper 1412PKS	0	0	0	1	0	0	1	1	3/1=3	0.4771	1	0	0	1
LAN	0	0	1	1	0	1	1	2	3/2=1.5	0.1761	0.3691	0	0.3691	0.3691
PreSurio CQ41-203TU	0	0	1	0	0	1	0	1	3/1=3	0.4771	1	0	1	0
SUZUKI	0	0	0	1	0	0	1	1	3/1=3	0.4771	1	0	0	1
T6500	0	0	0	1	0	0	1	1	3/1=3	0.4771	1	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

5.1. Euclidean Distance

After the calculation of the weight there is need to find the Euclidean Distance. This has important implications for multiple term queries. Consider, for example, a query Q consisting of two terms, k1 and k2. A term space representation of this query contains two extreme points:

- one at (1, 1) corresponding to documents completely relevant; i.e., containing both terms.
- one at (0, 0) corresponding to documents completely irrelevant; i.e., containing neither terms.

Therefore the maximum Euclidean Distance or maximum displacement (d_{max}) between the two points is:

$$d_{max} = \sqrt{(1-0)^2 + (1-0)^2} = \sqrt{2} = 1.4142$$

For **AND** queries, a similarity measure can be computed by subtracting the Euclidean Distance between the (1, 1) and (W_{k1}, W_{k2}) points from the maximum distance, d_{max}

$$d_{AND} = d_{max} - \sqrt{(1-W_{k1})^2 + (1-W_{k2})^2}$$

And **OR** query of the Euclidean Distance of a document at (W_{k1}, W_{k2}) must be d<1.4142. The equation is as follows [3]:

$$d_{OR} = d_{max} - \sqrt{(W_{k1} - 0)^2 + (W_{k2} - 0)^2}$$

5.2. Normalized Similarity Scores

To compare similarity scores for a variety of scenarios, there is need to normalize all distances by dividing by d_{max}. Thus, for AND and OR queries we obtain the following equations [3]:

$$Sim\ (Q_{k1\ AND\ k2}, D) = 1 - \sqrt{\frac{(1 - W_{k1})^2 + (1 - W_{k2})^2}{2}}$$

$$Sim\ (Q_{k1\ OR\ k2}, D) = 1 - \sqrt{\frac{W_{k1}^2 + W_{k2}^2}{2}}$$

For user query: "((Intel Core i3 **OR** Intel Core2 Duo) **AND** 2GB) **AND** (**NOT** Acer)" and documents, the system firstly calculate the inner **OR** query parenthesis and then calculate **AND** query and finally calculate **NOT** query.

For document D_1, the query "Intel Core i3 **OR** Intel Core2 Duo": k_1 = Intel Core i3 and k_2 = Intel Core2 Duo,

$$d_{OR} = d_{max} - \sqrt{(W_{k1} - 0)^2 + (W_{k2} - 0)^2}$$

$$= \sqrt{2} - \sqrt{(0 - 0)^2 + (0 - 0)^2} = 1.4142$$

$$Sim\ (Q_{k1\ OR\ k2}, D_1) = 1 - \sqrt{\frac{W_{k1}^2 + W_{k2}^2}{2}}$$

$$= 1 - \sqrt{\frac{(0-0)^2 + (0-0)^2}{2}} = 1$$

For query "(Intel Core i3 **OR** Intel Core2 Duo) **AND** 2GB": k_1= Intel Core i3 **OR** Intel Core2 Duo and k_2 = 2GB,

$$d_{AND} = d_{max} - \sqrt{(1 - W_{k1})^2 + (1 - W_{k2})^2}$$

$$= \sqrt{2} - \sqrt{(1 - 1)^2 + (1 - 0.3691)^2}$$

$$= 1.4142 - 0.6309 = 0.7833$$

$$Sim\ (Q_{k1\ AND\ k2}, D) = 1 - \sqrt{\frac{(1 - W_{k1})^2 + (1 - W_{k2})^2}{2}}$$

$$= 1 - \sqrt{\frac{(1 - 1)^2 + (1 - 0.3691)^2}{2}}$$

$$= 1 - \sqrt{0.3980} = 0.3691$$

For query "((Intel Core i3 **OR** Intel Core2 Duo) **AND** 2GB) **AND** (**NOT** Acer) ": k_1= ((Intel Core i3 **OR** Intel Core2 Duo) **AND** 2GB) and k_2= **NOT** Acer,

where $W_{k2} = 1 - W_{Acer} = 1 - 1 = 0$

$$d_{AND} = d_{max} - \sqrt{(1 - W_{k1})^2 + (1 - W_{k2})^2}$$

$$= \sqrt{2} - \sqrt{(1 - 0.3691)^2 + (1 - 0)^2}$$

$$= 1.4142 - 1.1824 = 0.2318$$

$$Sim\ (Q_{k1\ AND\ k2}, D) = 1 - \sqrt{\frac{(1 - W_{k1})^2 + (1 - W_{k2})^2}{2}}$$

$$= 1 - \sqrt{\frac{(1 - 0.3691)^2 + (1 - 0)^2}{2}}$$

$$= 1 - \sqrt{0.6990} = 1 - 0.8361$$

$$= 0.1639$$

And then, the system calculates the similarity of D_2 and D_3 as like above. Table 3 shows the result of documents (D_1, D_2, and D_3).

Table 3: Similarity results sorted in descending order

Document	Brand	Model No	Similarity Result
D_2	Compaq	PreSurio CQ41-203TU	0.5262
D_3	SUZUKI	Kuiper 1412PKS	0.3876
D_1	Acer	Aspire TimeLine 4810T	0.1639

6. Conclusion

Information Extraction is an emerging technology. It is characteristic of emerging technologies that their best applications are not always understood immediately, whether by developers or by users. The development and evolution of the important uses for emerging technologies, and their acceptance, require considerable support by technologists, willingness to experiment on their part, and collaboration between user and technologist. IE systems are complementary to Information Retrieval (IR) systems and can make information retrieval more precise. This paper describes how IR system can be used as a front-end to an IE system, and how Boolean Model is applied in IR system. The main purpose of this system is to build up the user preferences in on-line shopping.

7. References

[1] Christopher D. Manning, Prabhakar Raghavan, and Hinirich Schutze, "Introduction to Information Retrieval".

[2] David E. Losada and Alvaro Barreiro, " Using a Belief Revision Operator for Document Ranking in Extended Boolean Models".

[3] Dr. E. Garcia, "The Extended Boolean Model".

[4] Elizabeth D. Liddy, "Document Retrieval, Automatic".

[5] Jim Cowie and Yorick Wilks, "Information Extraction".

[6] Jongpill Choi, Minkoo Kim, and Vijay V. Raghavan, "Adaptive relevance feedback method of extended Boolean model using hierarchical clustering techniques".

[7] Joon Ho Lee, "Properties of Extended Boolean Models in Information Retrieval".

[8] O. Cordon, F. de Moya, and C. Zarco, "A GA-P Algorithm to Automatically Formulate Extended Boolean Queries for a Fuzzy Information Retrieval System".

[9] Peter Ingwersen, "Information Retrieval Interaction".

[10] Ricardo Baeza-Yates and Berthier Ribeiro-Neto, "Modern Information Retrieval".

[11] Robert Gaizauskas and Alexander M. Robertson, "Coupling Information Retrieval and Information Extraction: A New Text Technology for Gathering Information from the Web".

[12] Sarah M. Taylor, "Improving Analysis with Information Extraction Technology".

[13] Tina Eliassi-Rad and Jude Shavlik, "A Theory-Refinement Approach to Information Extraction".

[14] Ting-Peng Liang, Yung-Fang Yang, Deng-Neng Chen, and Yi-Cheng Ku, "A semantic-expansion approach to personalized knowledge recommendation".