International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:1, 2009

Auto Classification for Search Intelligence

Lilac A. E. Al-Safadi

***Abstract*—**This paper proposes an auto-classification algorithm of Web pages using Data mining techniques. We consider the problem of discovering association rules between terms in a set of Web pages belonging to a category in a search engine database, and present an auto-classification algorithm for solving this problem that are fundamentally based on Apriori algorithm. The proposed technique has two phases. The first phase is a training phase where human experts determines the categories of different Web pages, and the supervised Data mining algorithm will combine these categories with appropriate weighted index terms according to the highest supported rules among the most frequent words. The second phase is the categorization phase where a web crawler will crawl through the World Wide Web to build a database categorized according to the result of the data mining approach. This database contains URLs and their categories.

***Keywords*—**Information Processing on the Web, Data Mining, Document Classification.

I. Introduction

EXISTING search engines such as Google (www.google.com), Yahoo (www.yahoo.com) and MSN (www.msn.com) builds and stores huge keyword-based indices that help locate sets of Web pages that contain specific keywords. These often return a long list of search results, ranked by their relevancies to the given query. Web users have to go through the list and examine the titles and short snippets to identify their required results. This is a time consuming task when keywords belong to many categories. For example, "jaguar" can be categorized under animals, cars, music, education, cities and more. In [9], a query for "jaguar" animal found in the 10th, 11th, 32nd and 71st pages of Google results.

Based on these observations, we believe intelligence should be integrated with the Web search engine service to enhance the quality of Web searches. A possible solution to this problem is to classify Web pages and search results into different categories, and to enable users to identify their required category at a glance. [13] Showed that relevant documents tend to be more similar to each other. This work could also contribute to concept-based search engines. Applying a concept-based search engine described in [12] shall include synonyms, super and sub concepts that will perform a parallel search in all domains and that will obtain an even larger set of documents than the search for the keywords alone would return. Classified search engine selects a smaller set of the search engine database index to search in. Thus provide the basis for discovering more relevant documents.

Classified directories organize a usually smaller subset of Web material into a hierarchy of thematic categories: each category lists Web pages deemed relevant to that category. Although Yahoo (www.yahoo.com), Lycos (www.lycos.com) and LookSmart (www.looksmart.com) use human readers to classify Web documents, reduced cost and increased speed make automatic classification highly desirable. A classical document clustering approach, vector space model [6], which represents each document using n-dimensional vector (where n is the number of keywords) also suffers from this problem. By using our approach, the dimension of the keyword representation is highly reduced, because it is associated with the categories not the documents. Typical classification methods use positive and negative examples as training sets, and then assign each document a class label from a set of predefined topic categories based on pre-classified document examples.

In this paper we present an auto-classification method for Web pages. It defines which category a pre-classified document belongs to. The rest of the paper is organized as follows; a formal notation for the auto-classification algorithm is presented in section 2. The proposed algorithms are presented in section 3. The overall structure of the system is illustrated in section 4. Validation experience in section 5, and conclusions will follow in section 6.

II. Auto Classification of Web pages – Formal Notation

Given a database of Web pages D, where each Web page t is a list of terms (appeared simultaneously), Auto-Classification Algorithm is the technique that deals with the discovery of all the rules that correlate the presence of one set of terms with a category.

The following is a formal statement of the problem:

Let I is a Lexicon with a set of terms, I = {i1; i2; . . .; im} be a set of m distinct terms. Let d be a set of n Web pages belonging to category c, where each page t includes a set of terms such that $t \subseteq I$. Associated with each category a unique identifier, called CID. Associated with each Web page is a unique identifier, called its TID. We say that a Web page t contains X, a set of some terms in I, if $X \subseteq t$ and $X \subset I$. Hence, the database D is kept normalized and each database record is in the form <CID, TID, X>. A Web page of k terms is called k-termset.

An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, $X \subseteq t$, $Y \subseteq t$ and $X \cap Y = \phi$; The support

Author is with Department of Information Technology, College of Computer & Information Science, King Saud University, Riyadh, Saudi Arabia. (e-mail: lalsafadi@ksu.edu.sa).

degree is the co-occurrence frequency of set of terms within the category. The greater the degree of support, the more correct the mapping relationship between category and set of terms.

The rule X → Y has support sup in the pages set d if sup% of Web pages in d contain X ∪ Y . Support(X →Y) = freq(X, Y)/n.

The degree of confidence reflects the proportional of Web pages that contain both X and Y in d, to those that contain X only. The rule X → Y holds in the category c with confidence conf if conf% of pages in c that contain X also contain Y. Confidence(X →Y) = freq(X, Y)/freq(X).

Given a set of Web pages d belonging to a category c, the problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support (called minsup) and minimum confidence (called minconf) respectively.

III. The Auto-Classification Stages

The proposed Auto-Classification technique has two phases. The first phase is a training phase where human experts determines the categories of different Web pages, and the supervised Data mining algorithm will combine these categories with appropriate weighted index terms according to the highest supported rules among the most frequent words.

The second phase is the categorization phase where a web crawler will crawl through the World Wide Web to build a database categorized according to the result of the data mining approach. This database contains URLs and their categories.

The Auto-Classification technique that discovers the category in which a set of Web pages belong to involves many stages:

1. Data Collection

To complete the training environment of the system, we need samples training Web pages that sufficiently represent the category. Hence, we collect source data to build up the source database. The source data is a set of categorized and pre-classified training set of pages. Let $T = \{t_1; t_2; \ldots; t_n\}$ be a set of Web pages belonging to category c, such that $T \subseteq D$.

2. Data cleaning

In order to mine the web pages, we need to extract salient terms. Therefore, some basic web page pre-processing is needed. It analyzes the Web pages, identifies the different HTML sections, parses and extracts the text. Extracted strings are normalized for processing. Linguistically, normalization of a word goes through a process known as morphological analysis. It first strips out all suffix and prefixes, stop words, and special characters.

In order to apply Apriori algorithm [19], is should be assumed that the terms are kept in their lexicographic order in each Web page. This will originate a set of salient terms terms $X \subseteq t$ and $X \subset I$ ready for data mining. We use the notation *x[1], x[2], ..., x[k]* where *x[1] < x[2] < ...< x[k]* to represent a *X k-termset*.

3. Data Enrichment

A simple technique for extracting relevant terms is counting their frequencies in a given set of preprocessed documents. In general this approach is based on the assumption that a frequent string in a set of domain-specific extracted terms indicates occurrence of a relevant terms.

Freq(string) is the probability of the string appearance in the Web page *t*. $x_i.freq$ determines strings of high frequency, i.e. greater than or equal *minfreq*. The result is a set of *1-termset* (called L_1) which represent the candidate salient terms, such that $L1 \subseteq X$.

Then we count the support of L_1 candidates $x_i.sup$. This represent is the probability that the term co-exist with the category in the source database and determine the seed set of termset for generating the set of potentially set of terms co-occur within a category.

The above stages will provide us with a data source ready for the terms association extraction.

4. Association rules extraction

The association rule extraction algorithm discovers the termsets, which represent the different associated terms in a category, through performing multiple passes over the data. A number of algorithms has been developed to generate candidate sets such AIS [20] and SETM [14]. We used Apriori algorithms [19] in this work.

Pass *1*, we start with a L_1 termsets found in the previous pass to generate new potentially termsets, (candidates), and count their actual support.

Pass *k*, we start with a L_K termsets found to be large in the previous pass to generate new potentially candidates and count their actual support.

This process continues until no new candidates are found. This determines the correlation of the set of terms and its category. Algorithm 1 below gives the association rule extraction algorithm used during the Training phase. Algorithm 2 gives the classification algorithm used during the crawled page classification phase.

Algorithm 1: Training

// extracting 1-itemset (candidates) from all Web pages belonging to a category *c*
Forall categories *c* in *C* **do begin**
  Forall pages *t* in *c* **do begin**
    Forall terms *x* in *t* **do**
      $X = \{ x \mid x.freq \geq minfreq\}$
  End
  $L_1 = \{ x \mid x.support \geq minsup\}$

// association rules extraction algorithm to discover termset
For ($k = 2$; $L_{k-1} \neq \phi$; k++) **do begin**
  C_k = apriori-gen(L_{k-1}); // generating new candidates using Apriori algorithm [19]
  Forall pages *t* in *d* **do begin**
    C_t = subset(C_k , *t*); // Candidates contained in t

Forall candidates $c \in C_t$ **do** *c:count*++;
End
$L_k = \{c \in C_k \mid c.count \geq minsup\}$
End
Answer = $\cup_k L_k$; // set of rules R
// Repeat the above fo all categories
End

Algorithm 2: Classification

Forall terms *x* in *t* **do**
$L_1 = \{ x \mid x.support \geq minsup\}$

// association rules extraction algorithm to discover termset
For (k = 2; $L_{k-1} \neq \phi$; k++) **do begin**
C_k = *apriori-gen*(L_{k-1}); // generating new candidates using Apriori algorithm [19]
Forall pages *t* in *d* **do begin**
C_t = subset(C_k , t); // Candidates contained in t
Forall candidates c $\in$ C_t **do** c:count++;
End
$L_k = \{c \in C_k \mid c.count \geq minsup\}$
End

// set of termset with highest support
$A = \{L_i \mid max(L_i.support)\}$;
// the category of the Web page based on the i-termset
Answer = *Similarity (A)*;

The similarity function takes as argument L_i, the set of *i* terms in *D*, and the set of i-termset returned from Web page t with the highest support. It returns the category CID of the matching set of items with highest support. The function works as follows

SELECT CID
FROM L_i p, t.L_i q
WHERE p.$term_1$ = q.$term_1$, … p.$term_i$ = q.$term_i$, and max(p.support);

IV. The Structure of the System

The system of the search service comprises four main components that are depicted on Fig. 1. The main components are

- Web crawler
- Preprocessor
- Indexer and categorizer module
- Searcher
- Ranker

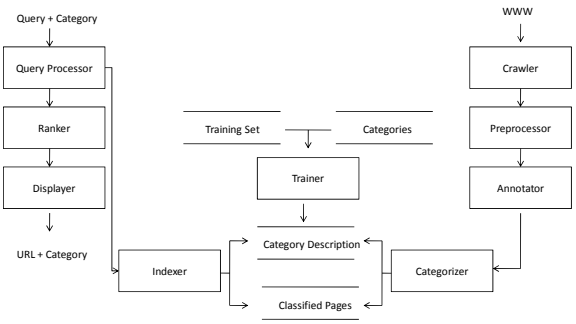

Fig 1 The structure of the Auto-Classifier System

Next we describe the task of each component and how they operate.

- *Web Crawler,* fetches Web pages for parsing and indexing. It follows the links in each page to visit new pages, and eventually visit all the Web pages in the Internet. The Web page Meta information are stored in a repository for future visits. Some pages can be manually submitted to the Crawler to visit.

- *Preprocessor,* performs the data cleaning and data enrichment functions

- *Indexer and Categorizer Module,* The indexer engine creates, maintains the index catalog that serves as the basis of answering search queries. The categorizer engine administrates and determines the category information to support the category-based search of the system. The categorizer component performs supervised machine learning, i.e. it learns the significant words/expressions of each category of the taxonomy by means of sample training documents.

- *Searcher*, it provides a mechanism for keyword-based or concept-based searching. It accepts the user's query, analyzes it, and rewrites the query (if necessary) using the specified thesaurus. Our Intelligent search engine query is imposed against the page index related to the specified domain.

- *Ranker*, sorts the resulted links to be displayed to the user. The Ranker computes the score of each result in the hit list, highlights the summary, sorts and presents the hit list.

V. Validation Experiment

In our experiment we used a sample of 2000 documents from each of 13 categories to train our automatic classifier. Then we evaluated the performance by testing it against a new set of 500 randomly selected documents from each of the categories (not including documents used in the training phase). For both the training and classification we stripped the pages of their html tags.

Our classifier was given no priori information about the semantic content of the 13 categories, simply 2000 examples documents from each. After training our classifier, we are able to automatically produce a list of key words, which are the most distinguishing terms for each category. In Fig. 2, accuracy of the search results by a search engine that uses Vector Space versus our proposed search engine is depicted. The proposed algorithm outperformed the vector space algorithm in all categories in terms of accuracy. More experimentation are being studied right now to be included in future publication.

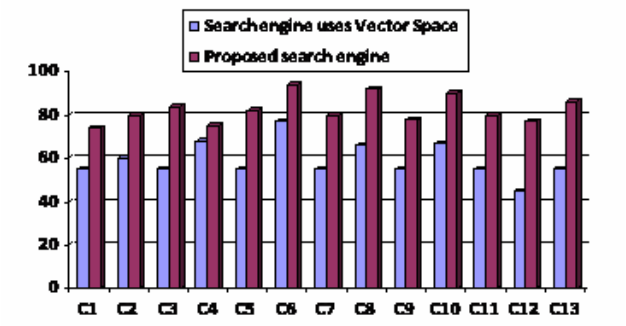

Fig. 2 Accuracy of the search results by a search engine that uses Vector Space versus our proposed search engine

VI. Conclusion

In this paper, a classification algorithm of Web pages into a set of categories using data mining techniques is presented. The proposed technique is based on analyzing relationships between a set of documents and the terms they contain by producing a set of rules relating the category of the document, its terms and their frequencies. After training our classifier, we are able to automatically produce a list of terms, which are the most distinguishing for each category.

References

[1] Kolcz, V. Prabakarmurthi, J.K. Kalita. "Summarization as feature selection for text categorization". Proc. Of CIKM01, 2001.
[2] Z. Broder, S.C. Glassman, and M.S. Manasse, "Syntactic Clustering of the Web," Proceedings of the 6th International World Wide Web Conference, April 1997, pp. 391-404.
[3] Chekuri, M. Goldwasser, P. Raghavan, and E. Upfal, "Web Search Using Automatic Classification," Proceedings of the 6th International World Wide Web Conference, April 1997.
[4] E. Rasmussen, "Chapter 16: Clustering Algorithms," in W. B. Frakes and R. Baeza-Yates, editors, Information Retrieval: Data Structures &Algorithms, Prentice Hall, 1992, pp. 419-442.
[5] G. Salton, editor. "The SMART retrieval system: experiments in automatic document processing," Prentice-Hall Series in Automatic Computation, Englewood Cliffs, New Jersey, 1971, Chapters 14-17.
[6] G. Salton, A. Wong, and C.S. Yang, "A Vector-Space Model for Information Retrieval," Communications of the ACM, vol. 18, no. 11, 1975, pp. 613-620.
[7] H. Chen and S. T. Dumais. Bringing order to the Web: Automatically categorizing search results. Proc. of CHI2000, 2000, 145-152.
[8] H. Mahmood, "CW3S: New Classification Algorithm for World Wide Web Search Engines ",to appear at NITS'08, november 2008, Riyadh, KSA.
[9] H. Zeng, Q. He, Z. Chen, W. Ma and J. Ma, "Learning to cluster Web Search Results", The 27th Annual International ACM SIGIR Conference (SIGIR'2004), July 2004
[10] J. L. Chen, B.Y. Zhou, J. Shi, H.J. Zhang, and Q.F. Wu. Function-based Object Model Towards Website Adaptation, Proc. of WWW10, HK, China, 2001.
[11] J. Pitkow and P. Pirolli, "Mining Longest Repeating Subsequences to Predict World Wide Web Surfing," Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems (USITS'99), Oct 1999, pp.139- 150.
[12] L. Al-Safadi, "Enhanced Arabic Search Engine", The Fifth International Conference on Information Integration and Web-based Applications & Services (iiWAS2003), Jakarta, Indonesia, September 15 - 17, 2003
[13] M. Hearst, J. Pedersen, "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96), Zurich, June 1996.
[14] M. Houtsma and A. Swami. Set-oriented mining of association rules. Research Report RJ 9567, IBM Almaden Research Center, San Jose, California, October 1993.
[15] M. L. Shyu, S.-C. Chen, and C. Haruechaiyasak, "Mining User Access Behavior on the WWW," IEEE International Conference on Systems, Man, and Cybernetics, October 2001, pp. 1717-1722.
[16] M. L. Shyu, S.-C. Chen, C. Haruechaiyasak, C.-M. Shu, and S.-T. Li, "Disjoint Web Document Clustering and Management in Electronic Commerce," Proceedings of the Seventh International Conference on Distributed Multimedia Systems (DMS'01), September 2001.
[17] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: text summarization for Web browsing on handheld devices. Proc. of WWW10, Hong Kong, China, May 2001.
[18] R. Cooley, B. Mobasher, and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web," Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November 1997, pp. 558-567.
[19] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", Proceedings of the 20th VLDB Conference Santiago, Chile, 1994
[20] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C., May 1993.
[21] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced Hypertext Categorization Using Hyperlinks. Proc. of the ACM SIGMOD, 1998.
[22] S. J. Ker and J.-N. Chen. A Text Categorization Based on Summarization Technique. In the 38th Annual Meeting of the Association for Computational Linguistics IR&NLP workshop, Hong Kong, October 3-8, 2000.
[23] S. Miyamoto and K. Nakayama, "Fuzzy Information Retrieval Based on a Fuzzy Pseudothesaurus," IEEE Transactions on Systems, Man, and Cybernetics, vol. 16, no. 2, March/April 1986, pp. 278-282.
[24] T. Joachims. Transductive inference for text classification using support vector machines. Proc. of ICML-99, Bled, Slovenia, June 1999.
[25] Y.J Ko, J.W Park, J.Y. Seo. Automatic Text Categorization using the Importance of Sentences. Proc. of COLING 2002.
[26] Y. Li and R. Gopalan, "Effective Sampling for Mining Association Rules", 17th Australian Joint Conference on Artificial Intelligence Cairns, Australia, December 2004
[27] Y. Ogawa, T. Morita, and K. Kobayashi, "A Fuzzy Document Retrieval System Using the Keyword Connection Matrix and a Learning Method," Fuzzy Sets and Systems, vol. 39, 1991, pp. 163-179.