INTELLIGENT RETRIEVAL
Open Journal

Review

ACIRD: Intelligent Internet Document Organization and Retrieval

Shian-Hua Lin; Meng C. Chen; Jan-Ming Ho; Yueh-Ming Huang*
Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan

Corresponding author
Yueh-Ming Huang
Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan

Article information
Received: February 24th, 2023; **Revised:** March 31st, 2023; **Accepted:** April 1st, 2023; **Published:** May 22nd, 2023

Cite this article
Lin S-H, Chen MC, Ho J-M, Huang Y-M. ACIRD: Intelligent internet document organization and retrieval. 2023; 1(1).
doi: https://doi.org/10.70705/ppp.ir.2023.v01.i01.pp23-29

ABSTRACT

Automatic Classifier for the Internet Resource Discoveery (ACIRD) is a smart Internet information system that employs machine learning to categorize and retrieve documents from the Internet. It is presented in this article. The three parts that make up ACIRD are a two-stage search engine, a document classifier, and a knowledge acquisition procedure. Automatically acquiring classification information from classified Internet documents is part of ACIRD's knowledge acquisition methodology. In order to sort freshly acquired documents from the Internet into one or more categories, the document classifier uses its learnt categorization expertise. When compared to human specialists, ACIRD achieves comparable or higher performance in knowledge acquisition and document categorization, according to the experimental findings. In order to help users find information from diverse and large-scale Internet documents, the ACIRD two-phase search engine uses the learned classification knowledge and the provided class lattice to provide results that are hierarchically organized and easy to navigate, rather than the usual flat-ranked document list.

Keywords
Document classification; Data mining; Information retrieval; Search engine.

INTRODUCTION

As it has developed into a primary means of communication and information dissemination, the Internet's meteoric rise has altered traditional ways of life. Nevertheless, the information overflow phenomena has been brought about by the vast amounts of data available on the Internet. There are now a plethora of subject directories and search engines accessible online to help with this issue. Websites like AltaVista and InfoSeek may pull documents from the Internet in reaction to a user's search. On the other hand, subject directories like Yahoo! make it possible to search for relevant information by navigating a hierarchical topic structure. Search engines are made to help you find what you're looking for in a big document collection. Search engines may get thousands of items for a query with only one or two terms1 due to the enormous amount of content accessible on the Internet. As an example, consider 2. Lots of issues remain unresolved, even though a lot of money and effort have gone into this [3]. While many studies have focused on incorporating these new technologies into existing digital library information retrieval systems, very little has examined the architectural and life cycle perspectives of semantic and intelligent artificial challenges. Overwhelming amounts of data make it difficult to conduct targeted searches, even as search engines have become more efficient. In contrast to similar initiatives, we provide methods that make use of information in ontology search and expert systems, and we construct contextual profiles based on ontologies [4]. The characteristics of LO usage and user attitudes suggest that case-based retrieval, an ontology application, might be a suitable option.

OSearch results for the term "education and university" returned 87,368,493 results from AltaVista, 7,379,086 results from Infoseek, and 237,902 results from WebCrawler. Using a small number of keywords to rank several documents makes it exceedingly improbable that the resulting order would satisfy the user's information demands. Therefore, in order to get the needed information, the user has to fetch a bunch of boring papers. Use of relevance feedback [32] has been used by a number of search engines to either broaden or narrow the query depending on the user-selected content. Since it is very difficult to understand the user's actual purpose from the chosen content, relevancy feedback may not be useful.

The discrepancy between expected retrieval outcomes and actual ones is exacerbated by the conceptual divide between document creators and consumers. Web developers and users may use several phrases and terminology to portray the same idea, or the same term to characterize different things, due to the vast cultural and linguistic

diversity on the Internet. Consequently, users may not always get the papers they want via term-based search engines, and they often get thousands of pages that aren't relevant. In spite of the fact that both "airline schedule" and "flight schedule" are thought to signify the same thing, term-based search engines will not match the two. Maybe looking it up in a thesaurus may help. Another issue that might emerge is when a word, like "bank," can have many meanings depending on the context. One solution to this issue is to create a thesaurus for each individual domain. The Internet is diverse and ever-changing, making it impossible for a static thesaurus to keep up with the changing meanings of words in this context.

The current state of subject directory systems is hindered by the need to manually classify freshly gathered documents. As an example, the biggest Internet directory system, Yahoo!, has about 150 editors required to categorize web pages3, and its subject hierarchy has over 1.2 million connections. Directory systems have a much smaller collection of documents compared to search engines' databases. Directory systems aren't concerned with database performance or capacity but rather with categorizing online content according to relevant subjects.

Based on the conventional wisdom about information retrieval metrics like recall and precision, subject directories and search engines both have poor recall rates due to their relatively tiny databases and low accuracy rates due to the large number of results they return. The effective management and retrieval of Internet content requires the organization of documents according to a set of classes in order to achieve balanced precision/recall rates and to enable users to quickly locate essential materials [14]. With the help of its creators, the ACIRD4 system was able to accomplish [21, 22, 23].

speedy and accurate retrieval of online documents. Classified materials teach the system how to categorize. In addition, it explores the latent semantics of words by mining their association rules, and it refines its understanding of class lattice classifications by inferring from these relationships. The system employs a two-stage search technique that gives the user a hierarchically navigable view to aid with the Internet search.

This paper is structured as follows for the remainder. Related works are reviewed in Section 2. The purpose of ACIRD is defined in Section 3. We define key terms and provide the conceptual model in Section 4. The ACIRD learning methodology is extensively covered in Section 5. To back up the choices made during ACIRD's design, Section 6 shows results from tests with automated document categorization. Following this, Section 7 presents the two-stage search procedure. Section 8 concludes by outlining the work's contributions and potential future directions.

2. Related Work

This section reviews works related to this study on Internet information retrieval, document classification and data mining.

Internet Information Retrieval

The primary goal of much of the prior research on IR systems was to find ways to make retrieval faster and more accurate by using term-based indexing [8, 11, 28, 37] and query reformulation [...]. Using a pre-built dictionary, stop words, and stemming rules, term-based document processing first extracts terms from documents [10, 28, 30]. After the phrases have been retrieved, the weights of the terms are determined using a commonly used approach termed TF-IDF (or variants thereof) [31, 33]. Words and their weights may therefore stand in for a document. A query's and a document's similarity measure is the cosine of the product of their respective term vectors in a multi-dimensional vector space. The obtained documents are shown as a ranked list according to the measure, which indicates the level of relevancy of the documents and queries.

On the other hand, there is an indexing method that uses strings and all possible substrings instead of terms. This method is great for searches that involve strings of any length, like address matching, and character-based searches for languages like Chinese and Japanese. The string-based indexing method has a much larger storage need compared to the term-based indexing method. Furthermore, retrieval times are longer due to their complex data structures. However, when it comes to recovering

When it comes to Internet information finding activities, the string-based indexing strategy isn't the best fit since users often provide more generalized descriptions rather than precise texts. For better performance of many search functions, including range searching, most significant and frequent searching, regular expression searching, prefix searching, etc., numerous studies have built string-based indexing technologies, such as PAT-tree [4] and signature files [7]. On the other hand, you won't often see these search features online.

Search engines nowadays use a wide range of IR approaches; yet, there are notable distinctions between them with respect to indexing, representation, querying, and implementation.

Organizing data by index. Internet robots (also referred to as spiders or crawlers) or human users submit HTML documents (i.e., web pages) to search engines. To facilitate efficient retrieval, search engines, like traditional IR systems, index collections of words or phrases. A few of search engines make an effort to decipher and catalog ideas contained inside texts. For instance, Excite5 is aware of the connection between "senior citizens" and "elderly people," two connected notions. Search engines are able to give more weight to phrases based on the importance of tags thanks to the rich HTML structure.

Stand in. Full text indexing is used by the majority of search engines to quickly match documents with queries. In standard IR systems, documents are represented as a collection of term-weight pairings. Web pages are represented as term-weight indexes by most subject directory systems, which also include keyword search functionalities. Furthermore, pages are organized into a subject hierarchy that is manually created and updated.

Looking for information. Search engines use a variety of features to narrow down the vast array of search results. To get more specific results, you may use Boolean operators, which are provided by the majority of search engines. Additional features that may be used to enhance search results include precise word matching, sorting pages

according to associated sites, and limiting search to certain sites.

Putting it all together. Unlike IR systems, which operate in a static context, Internet search engines and subject directory systems must adapt to the ever-changing nature of the Internet. Systems that can handle the frequent creation, modification, and deletion of web pages need storage structures that can adapt and indexing algorithms that are efficient. Internet web page collecting faces yet another difficult challenge with the implementation of clever Internet robots.

The Internet is now home to hundreds of search engines that use IR technology. Renown search engines are known for their lightning-fast responses and comprehensive indexes. As a rule, the majority of people who engines take a page out of IR's playbook when it comes to indexing and ranking, and they bolster their own performance using high-tech hardware and software. Not when search engines provide no matching documents, but when they return too many, user happiness takes a nosedive. Readers interested in the present state of the most major online search engines should consult Search Engine Watch6.

Document Classification and Data Mining

There are a lot of methods for document categorization, but the two broad categories are manual and automated. Due to the enormous volume of papers available on the Internet, manual document categorization is impractical due to the high costs and lengthy processing times involved. Classification knowledge may be automatically learnt from training texts or collected from domain experts for automated classification [3]. It takes a lot of time and effort to acquire information from domain experts, but it's worth it in the end. In addition, there's a chance that the information we've gathered isn't comprehensive enough to be useful without resorting to complex models and theories. Automatically gained categorization information from training documents, on the other hand, is efficient but has accuracy limitations according to the learning model and training data used.

Information retrieval has been the primary emphasis of many text classification research [3, 9, 14, 15, 16, 17, 19, 20, 35]. Internet HTML [18] pages, not generic texts, are the subject of this effort, so the term "document classification" is used instead of "text categorization" in this context. Automated document grouping is what document categorization is all about. Several research have tackled this problem by using methods such as text clustering [12, 19], text classification [3, 17, 20], text filtering [25], relevance feedback [32], and similarity-based document retrieval [35]. As an example, ExpNet [35] finds the optimal category for the input content by using similarity measurement as the approach for ranking categories. In order to choose documents for filtering according to content and user interests, SIFTER [25] employs a vector space model for document representation, unsupervised learning for document classification, and reinforcement learning for user modeling. A k-nearest-neighbor method using belief scores as the distance measure, Bayesian independence classifiers, and relevance feedback are the three classification methods used by INQUERY [17]. Based on the probability overlap between documents and document clusters, Goldszmidt and Sahami suggested document clustering [12].

Algorithms created in earlier machine learning research have found widespread use in domains as diverse as healthcare and the financial sector. In particular, popular algorithms such as ID3 [26], C4.5 [27],

The document classification issue has been solved by using CN2 [5] and the AQ algorithm [24] on structured training data rather than non-structured textual data. Along similar lines, a plethora of document classification methods use feature sets to define texts before applying algorithms such rule-based induction algorithms [3], k-nearest-neighbor technique [9, 11], Bayesian classifiers [19], and mixed approaches (e.g. INQUERY [17]). These systems fail to take into account the variety of documents in terms used and their meanings since they are too focused on the document categorization process and learning algorithms. Assuming fixed semantics, the described feature in many learning applications is an attribute-value pair. But the meaning of a feature changes depending on the domain. Consider the domains "computer" and "food," where the document characteristic "apple" takes on distinct connotations.

In order to find significant relationships among transaction elements, mining association rules [1, 2, 34] are used. Finding the best way to organize items at a supermarket so that consumers can easily collect their goods is a common use of mining item associations. This research delves into the document's feature semantics by means of mining association rules.

3. The ACIRD System

Automatically collecting and categorizing online documents for better administration and retrieval is what ACIRD [21, 22, 23] is all about. Initially, ACIRD is devoted to enhancing the labor-intensive and costly manual categorization method that is used by several search engines on the Internet. To automatically categorize freshly acquired Internet documents, ACIRD makes use of classification expertise gained from papers that have been manually categorized. The two-phase searching process, made possible by classification information and the provided class lattice, displays the search results in a hierarchical perspective rather than a ranked document list, as is typical in traditional Internet document retrieval. In order to achieve ACIRD's overarching goal—auto-learning, auto-classification, and two-phase searching—all design choices, including measurement measures, are carefully considered. The next parts go into depth on each part of ACIRD, while this section gives an outline of the whole.

Figure 1 provides a schematic representation of the ACIRD process. Training data consists of Internet documents that have been manually given classes, and the domain expert supplies a class lattice that serves as the document domain's worldview. Class lattice classes' classification knowledge, or class indexes, is generated by the Classification Learner using training data. Documents are automatically retrieved from the Internet via the Internet Robot, and their characteristics are extracted during the Preprocessing Process. Once the Document Classifier receives incoming documents, it assigns them to the most suitable class(ies). Internet users who do ACIRD queries trigger the Two-Phase

Search Engine matches the queries with knowledge of documents, and classes and presents a hierarchical view to the users to facilitate the information discovery job. In this study, we focus on Internet HTML documents only, referred to herein as objects. Each object

is assigned a unique ID and parsed into a document index, a set of terms with weights, which is stored in the database. The set of term and weight pairs form the feature vector representing the object knowledge. Inverted indexes pointing to the occurring objects from terms are generated for efficient access during learning and accessing.

Expert
Internet User

Fig. 1. The Major Components and Workflow in ACIRD.

The given class lattice presents the worldview to ACIRD. In the lattice, each node represents a class, and every parent node is a super set of its child node. Nodes with no parent nodes besides the universal node are referred to as the most general nodes, and nodes with no child node besides the null node as most specific nodes. The automatic learning and classification process of ACIRD consists of two phases: a training phase and a testing phase. In the training phase, the training data consist of a set of manually classified document s. The learning process learns the classification knowledge in the sequence from the most specific classes to the most general classes of the given class lattice. For the most specific classes, the classification knowledge is generalized from the knowledge of objects in the class. For the other classes, the knowledge is generalized from their child classes and direct objects (i.e. the objects belong to the class, but not to any of its child classes). After the knowledge of classification is learned, the technique of mining association rules is employed to discover term associations inside each class so as to enhance the classification knowledge. As term

associations highly depend on the class domain, the best scope to apply term associations to refine the classification knowledge is a single class. Our previous study [22] demonstrated that the mined term association can enhance the term semantics dramatically. In the testing phase, the classifier employs the learned classification knowledge to assign classes to the test documents (usually the newly collected documents), and the assignments are compared with the classes assigned by human experts to verify the quality of the learned knowledge.

ACIRD provides a two-phase search engine that allows users to efficiently and effectively retrieve interesting documents via interactive navigation of the returned class hierarchy, rather than via a sequence of ranked documents. During the two-phase search, each user query string is parsed and formulated as a sequence of terms, called query term vector. Similarity matching based on the vector space model is applied to determine the relevance between the query and the classes in the class lattice as well as stored objects. Note that both class and object knowledge are also represented as term vectors. During the first phase, a class-level search is performed in which the query term vector is used to determine the qualified classes that form a shrunken view of the class lattice. If the user decides to further explore a qualified class in the returned class hierarchy, the query term vector is again employed to calculate the relevance of the subclasses of the selected class. When the user decides to explore the objects in a class, which is called object-level search, ACIRD matches and retrieves qualified documents in the class. The two-phase search not only reduces the search domain, but also presents a hierarchical conceptual view to aid the user in locating interesting information.

4. ACIRD Conceptual Model

In this section, we define the terminology used herein and introduce the conceptual model of ACIRD [22]. An entity is denoted by a lower case letter, and a set or series of entities by an upper case letter. For example, let c denote a class and C represent a set of classes. In the following, we describe the system entities with their notations in parentheses beginning from the high level concepts and continuing to the low level ones.

5. ACIRD Learning Model

In this section, we describe the learning model of ACIRD in detail. In the training phase, ACIRD adopts supervised learning techniques and treats previously classified documents as training objects. The testing phase is described in Section 6. ACIRD applies machine learning techniques to learn classification knowledge as shown in Fig. 2. The learning method is applied to each class of ACIRD lattice from the most specific classes to the most general ones. Each document is preprocessed into a weighted term vector. The dimension of the vector is then reduced by the Feature Selection Process in order to reduce the complexity of learning. For the most specific class, knowledge of the class

Knowc

is generalized from the knowledge of all the objects in the class, which can be represented by the Term Support Graph (TSG). For classes other than the most specific classes, the learning process is the same except that the initial weighted term vectors originate from its subclasses and direct objects. The mining association algorithm is then applied to mine associations of terms in TSG, which can be represented by the Term Association Graph (TAG). Combining TSG and TAG derives the Term each iteration of the refinement process, some terms may be promoted. As the promoted terms may be used to promote other terms, the promotion process is applied recursively until the stable state is reached.

5.1 Preprocessing Process and Knowledge Representation

The preprocessing process consists of two parsers, the HTM L Parser and Term Parser. The HTML parser parses an object into paragraphs and determines their weights by judging the ir associated HTML tags. The Term Parser partitions the paragraphs into sentences and extracts terms from sentences. The Term Parser also calculates term supports using the weight assigned by the HTML Parser and the term frequency.

HTML Parser

An HTML document consists of paragraphs in which the associated HTML tags [18] indicate their importance and provide meta- level information. Web developers highlight the contents using HTML tags, such as titles or headings. In addition, META tags allow developers to add extra information such as “CLASSIFICATIONS” and “KEYWORDS” to the document. Apparently, the implications

of tags must be considered while indexing documents. In ACIRD, human experts assign and adjust the weights of HTML tags by observing the outcomes of numerous experiments in order to improve the classification accuracy. HTML tags are classified into four types:

- Informative. Paragraphs enclosed by tags, such as CLASSIFICATION and KEYWORD in META, TITLE, Hn, B, I, and U, consist of either the meta knowledge of the documents or significant contents provided to users. Thus, the informative tags have the highest weights.
- Skippable. Tags, such as BR and P, do not affect the semantics of the document and are omitted.
- Uninformative. Contents enclosed by tags, such as AREA, COL, SCRIPT, and COMMENT, are invisible to users. Thus, these tags and their corresponding contents are excluded.
- Statistical. Contents enclosed by tags, such as !DOCTPYE, APPLET, OBJECT, SCRIPT, etc., are stored in a database for statistical purposes.

The HTML Parser is implemented with two stacks: one for HTML tags and the other for paragraphs.

The algorithm is executed in one document scan with computational complexity O(Knowo) .

Term Parser

The Term Parser partitions a paragraph into sentences, extracts terms from the sentences, and counts the term frequency(TF) of each term. After a term t is extracted from an object o , the support value supt ,o is measured based on TF and HTML weight, as defined in equation (5.1). This value, normalized in the range of [0, 1], indicates the importance of a term in representing the object:

Since a sentence may have more than one tag, the maximum weight of the tags is used to calculate the term support. In ACIRD, TF and the maximum tag weight are used to calculate the term support instead of using the TF IDF weighting approach. The Inverted Document Frequency (IDF), designed to enhance the discriminating capability of high frequency terms, is not critical in our hierarchical learning model and two-phase search discovery model. In ACIRD, a high frequency term is considered to represent its class and may be generalized to classification knowledge of its parent class, instead of being used as a discriminator of objects in the class [31].

Designed to handle multi- lingual documents, ACIRD currently supports English and Chinese only. With English, each extracted term is stemmed [31]. With a character-based language like Chinese, a sentence is segmented into meaningful multi-character terms. As there may be no apparent stop characters in a sentence, the Term Parser uses a pre-constructed term base structured as a B-tree [6] to quickly match and extract meaningful terms. The Term Parser extracts Chinese terms based on the heuristics of "long term first" to resolve ambiguity. That is, for two terms, one of which is a part of the other, the Term Parser extracts the longer one. In addition, the rules for Chinese term segmentation are provided to handle segmentations ambiguity between conflicting candidate terms. The complexity of term extraction is O(n2) , where n is the length of the input sentence.

8. Conclusions and Future Work

In this work, we introduced ACIRD, an online system for managing and accessing class documents. Machine learning and data mining algorithms may provide accurate classification expertise, as shown by our findings. Online documents may be automatically and accurately classified into classes using the classifier's knowledge of classification. Using the categorization information as a meta-index, the search domain may be shrunk to effectively retrieve potentially desired content, according to analysis of the user query log. Also, the results of a class-level search provide a clear categorization. In order to find the documents they need, users may link their queries with the given classes, browse those classes, and then do an object-level search. By doing so, the system aids users in finding information across a vast array of publications found on the Internet.

It will be necessary to upgrade the learning model in the future so it can adapt to the ever-changing Internet. Additionally, there is potential for further improvement in the classification accuracy of both the learning techniques and the classifier. Additional research and investigation into a few connected matters would be beneficial. For instance, it would be great if future research could look at ways to automate the construction of a thesaurus that matches the semantics of words in a particular domain by expanding the usage of term association mining in classes. In addition, by examining the user query log, the system may acquire new vocabulary that is not already in thesauruses, such as "MP3," "ICQ," and "CGI," which will help to broaden ACIRD's term base.

REFERENCES

[1] Presented at the ACM SIGMOD International Conference on Management of Data in May 1993, the paper "Mining Association Rules between Sets of Items in Large Databases" was written by R. Agrawal, T. Imielinski, and A. Swami.

(1) "Fast Algorithms for Mining Association Rules" was published in the Proceedings of the 20th International Conference on VLDB in September 1994 and was written by R. Agrawal and R. Srikant.

3. "Automated Learning of Decision Rules for Text Categorization" by C. Apte, F. Damerau, and S. M. Weiss was published in the July 1994 issue of the ACM Transactions on Information Systems, volume 12, issue 3, and ran from pages 233-251.

The paper "PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval" was presented at the 1997 ACM SIGIR International Conference on Information Retrieval.

Article cited as "The CN2 Induction Algorithm" in the Machine Learning Journal (vol. 3, no. 4, 1989, pp. 261-283) by P. Clark and T. Niblett.

The authors of "Introduction to Algorithms" (MIT Press, 1990) are T. H. Cormen, C. E. Leiserson, and R. L. Rivest.

[7] "Implementing Ranking Strategies Using Text Signatures" (Vol. 6, No. 1, Jan. 1998, pp. 42-62) by W. B. Croft and P. Savino in the ACM Transactions on Office Information Systems.

[8] "Optimizations for Dynamic Inverted Index Maintenance" was presented at the 13th International Conference on Research and Development in Information Retrieval in 1990 by D. Cutting and J. Pedersen.

In 1973, John Wiley & Sons published a book titled "Pattern Classification and Scene Analysis" by R. O. Duda and P. E. Hart.

[10] "Information Retrieval—Data Structures & Algorithms" by W. B. Frakes and R. Baeza-Yates, published by Prentice Hall in 1992.

"Models for Retrieval with Probabilistic Indexing" by N. Fuhr was published in 1989 in the journal Information Processing and Management (Vol. 25, No. 1) and can be found on pages 55 to 72.

http://robotics.stanford.edu/users/sahami/papers-dir/gm-clustering.ps is a reference to a work by M. Goldszmidt and M. Sahami titled "A Probabilistic Approach to Full-Text Document Clustering" (TR ITAD-433-MS-98-044, SRI International, 1998).

UMass Technical Report 94-17, authored by Y. F. Jing and W. B. Croft, may be found online at http://cobar.cs.umass.edu/info/psfiles/irpubs/jingcroftassocthes.ps.gz.

"The Use of Automatically-Obtained Classifications for Information Retrieval" (K. S. Jones and D. M. Jackson, 1970, pp. 175-201, Information Processing and Management (IP&M), Vol. 5, 1970).

[15] "Automatic Term Classification and Retrieval" by K. S. Jones and R. M. Needham was published in 1968 in the journal Information Processing and Management, volume 4, issue 1, pages 91–100.

In their 1996 UMass Computer Science Technical Report, IR-78, T. Kalt and W. B. Croft presented "A New Probabilistic Model of Text Classification and Retrieval." The paper may be found online at http://cobar.cs.umass.edu/info/psfiles/irpubs/ir.html.

Referenced in [17] "Combining Classifiers in Text Categorization" by L. S. Larkey and W.B. Croft, published in 1996 at ACM SIGIR' 96, pages 289-297. Web address: http://andrew2.andrew.cmu.edu/rfc/rfc1866.html; author: T. Berners-Lee; page number: 18.

The paper "An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task" was presented at the ACM SIGIR'92 conference in 1992 and can be found on pages 37-50.

"Training Text Classifiers by Uncertainty Sampling" was published in 1994 at ACM SIGIR and was co-authored by William Gale and D. Lewis.

In December 1996, at the International Symposium on Multi-technology and Information Processing (ISMIP'96), S. H. Lin, M. C. Chen, J. M. Ho, and Y. M. Huang presented their work titled "The Design of an Automatic Classifier for Internet Resource Discovery" (pp. 181-188).

In their 1998 paper "Extracting Classification Knowledge of Internet Documents: A semantics Approach," S. H. Lin, C. S. Shih, M. C. Chen, J. M. Ho, M. T. Kao, and Y. M. Huang presented their findings at the ACM SIGIR conference.

Citation: "A Collaborative Internet Documents Access Scheme Using ACIRD" by S. H. Lin, C. S. Shih, M. C. Chen, J. M. Ho, M. T. Kao, and Y. M. Huang, published in 1998 at the International Computer Symposium on Software Engineering and Database Systems (ICS'98).

The AQ15 Inductive Learning System: An Overview and Experiments was published in 1986 by the Department of Computer Science at the University of Illinois in Urbana as Technical Report ISG 86-20, UIUCDCS-R-86-1260. The authors of this report are R. S. Michalski, I. Mozetic, and J. Hong.

"A Multilevel Approach to Intelligent Information Filtering: Model, System, and Evaluation" (Vol. 15, No. 4, October 1997, pp. 368-399), written by J. Mostafa, S. Mukhopadhyay, W. Lam, and M. Palakal, was published in the ACM Transactions on Information Systems.

The article "Induction of Decision Trees" by J. R. Quinlan was published in 1989 in Machine Learning, Volume 1, pages 261-283.

C4.5: Programs for Machine Learning by J. R. Quinlan, San Mateo, CA: Morgan Kaufmann Publishers, 1993 [27].

G. Salton, "Automatic Information Organization and Retrieval," McGraw-Hill, 1968, p. 28.

[29] "An Evaluation of Term Dependence Models in Information Retrieval," LNCS 146, 1983, pp. 151-173, by G. Salton, C. Buckley, and C. T. Yu. "Introduction to Modern Information Retrieval," McGraw-Hill, 1983, by G. Salton and M. J. McGill.

In 1989, Addison-Wesley published G. Salton's "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer." (21).

[32] "Improving Retrieval Performance by Relevance Feedback" (A.S.I.S. Journal, Vol. 41, No. 4, 1990, pp. 188-297), written by G. Salton and C. Buckley.

The paper "New Techniques for Best-Match Retrieval" was published in the ACM Transactions on Office Information Systems in January 1990 and can be found on pages 140–158.

The paper "Mining Quantitative Association Rules in Large Relational Tables" was presented at the 1996 ACM SIGMOD International Conference on Management of Data by R. Srikant and R. Agrawal.

The paper "Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval" was presented at the 1994 ACM SIGIR conference and can be found on pages 13-22.

[36] "A Framework for Effective Retrieval" by C. T. Yu, W. Meng, and S. Park was published in 1989 in the ACM Transactions on Database Systems (Vol. 14, No. 2) and can be found on pages 147–167.

[37] "A World Wide Web Resource Discovery System" (WWW Journal, Vol. 1, No. 1, Winter 1996), B. Yuwono, S. L. Y. Lam, J. H. Ying, and D.