Automatic Categorization of Web documents Based on Application Ontology

A Thesis Proposal
Presented to the
Department of Computer Science
Brigham Young University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Linus W. Kwong
April 7, 1999

I Introduction

Web users rely on World Wide Web search engines, such as Yahoo! and AltaVista, to retrieve Web documents of interest. Whether a search engine provides categories for a user to click on or a query facility for a user to type in keywords, the Web documents retrieved still suffer from poor precision (i.e., too many irrelevant documents are retrieved) and poor recall (i.e., too many relevant documents are omitted) [CGRU97]. Alternative approaches on categorization of Web documents therefore become necessary.

This thesis proposes the automatic categorization of Web documents with respect to an application ontology[1]. An application ontology [ECLS98] has two parts:

(i) *an ontological model instance* (derived from Conceptual Model). This instance consists of sets of objects, relationships among the objects, and constraints over the objects; and

(ii) *data frames*. A data frame represents each object set in (i) in the form of possible contextual keywords and constants.

This thesis restricts itself to application ontologies which are (i) data rich, i.e., contain a number of identifiable constants, such as dates, names, and account numbers; and (ii) narrow in ontological breadth, i.e., have a relatively small ontology [ECJ+98]. In this thesis, we focus on four application ontologies, namely car advertisements, job advertisements, obituaries, and university course descriptions, which satisfy the restrictions. We focus on retrieving Web documents with multiple records, i.e., each of these records should contain a group of information relevant to a domain of interest [YJ98].

[1] Since an application ontology defines a domain of interest and a (IR) category is a domain of interest, application ontologies and categories will be interchangeably used in this thesis proposal.

In this thesis, we determine the relevance of a Web document with multiple records to a particular application ontology by using two mathematical vector-space based IR models: the Vector Space Model (VSM) and the Clustering Model (CM). An assumption [Salton88] applied to these two IR models is that there exists a set of n different terms which represent a category and a document in the category.

- VSM [Salton88]. The VSM interprets each of the n terms in the category as an axis of an n-dimensional vector space. The VSM represents k Web documents as k n-dimensional vectors and the category as an n-dimensional category vector in the n-dimensional vector space. The coefficients of each of the k n-dimensional vectors (the category vector, respectively) are the frequencies of the n terms in the corresponding Web document (the category vector, respectively).
- CM [SM83]. Like VSM, CM also interprets each of the n terms in the category as an axis of an n-dimensional vector space and represents k Web documents as k n-dimensional vectors in the n-dimensional vector space. However, CM differs from VSM in that CM creates clusters sets of Web documents based on the "similarity" among their corresponding n-dimensional vectors. CM represents each cluster C as an n-dimensional vector, whose coefficients are the average frequencies of the n terms which are found in each of the Web documents in C. This vector is called the term-centroid vector. When the term-centroid vector and the category vector point to the same or nearly the same direction, the Web documents in C (represented by the term-centroid vector) are relevant to the ontology (represented by the category vector).

Similar work on automatically categorizing Web documents has appeared. [ITN96, CGRU97] both submit a query to a search engine and collect a set of documents which the search engine returns. They also define a category by giving a set of n different terms. For each returned document, they determine the frequencies that these terms

appear in the document. The probability that [ITN96, CGRU97] classify a document as belonging to a category depends on the frequencies. Our automatic process is similar, but differs in two ways:

(1) *The creation of the search engine query.* In [ITN96, CGRU97], it is required that a user manually creates a search engine query. However, our categorization program automatically extracts a set of object set names from the ontological model instance in an application ontology. It forms the query which is the logical OR of all the object set names.

(2) *The creation of terms that describe a category.* [CGRU97] define a category by manually extracting terms from a pre-classified set of Web pages from search engines, such as Yahoo! and Infoseek. [ITN96] define a category by using existing information science terminology (subject dictionary), and some terms that describe the category using thesauri. Our approach uses an object-oriented ontology to define a category. We use keywords and keyword-associated constants (which are automatically extracted from the data frames in the ontology) to describe the category.

We conducted an experiment on twenty pre-classified obituary Web documents retrieved from Yahoo!. The experimental results show 90% recall and 97% precision, and our approach enhances the precision of existing search engines in retrieving Web documents. An increase in precision has two consequences: a) it saves users' time when browsing retrieved Web documents; (b) it means retrieved Web documents can be used as input of the data extraction tools proposed in [RL94, KSa97, Sod97, HGMC+97, ECLS98].

II Thesis Statement

This thesis proposes the automatic categorization of Web documents with respect to an application ontology. The approach uses application ontologies, the Vector Space IR Model, and the Clustering IR Model.

III Methods

This thesis proposes the automatic categorization of Web documents with respect to an application ontology. The approach uses application ontologies, the Vector Space IR Model (VSM) and the Clustering IR Model (CM).

Recall that an application ontology has two parts: (i) an ontological model instance, and (ii) data frames. A *low-level object* in a data frame is either lexical or non-lexical. A *lexical object* is an object whose representation is indistinguishable from the object itself, whereas a *non-lexical object* is an object whose representation is distinguishable from the object itself and is represented by its object identifier and its relationships with other lexical and non-lexical objects [Embley98]. A *high-level object* in a data frame is a set of abstract views of information and can include sets of objects, relationships, and constraints. Each object in a data frame has a set of possible keywords and a set of possible constants. The *cardinality constraint* of each object defines the minimum and maximum number of times each keyword (or constant) participate in a relationship between the object and other object.

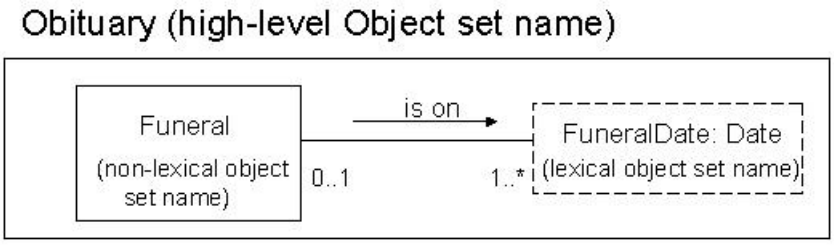

Figure 1: A portion of an ontological model instance of the obituary ontology in graphical form

```
Date matches [16]
   constant -- Month Day, Year
         { extract Month, "\.?\s*(1\d|2\d|0?[1-9]|[12]\d|3[01]),?\s*(1[89]\d\d|20\d\d)"; },

         -- Month, Day
         { extract Month, "\.?\s*(1\d|2\d|0?[1-9]|[12]\d|3[01])"; },

         -- in Year
         { extract "\d+"; context "in\s*(1[89]\d\d|20\d\d)\b"; };
   lexicon  {
          Month case insensitive;
          filename "month.dict";
         };
end;
Birth Date matches [16]
   keyword     "\bborn\b";
end;
Death Date matches [16]
   keyword     "\bdied\b",
               .
               .
               .
end;

Interment Date matches [16]
   keyword     "\binterment\b", "\bburial\b";
end;

Funeral Date matches [16]
   keyword     "\bfuneral(?!\s+(home|director))\b",
               .
               .
               .
end;
Viewing matches [0]
   keyword     "\bviewing\b", "\bmay\s+call\b", "\bvisitations?\b";
end;

Interment [0:1] has Interment Date [1:*];
Viewing [0:1] has Viewing Date [1:*];
Funeral [0:1] is on Funeral Date [1:*];
Deceased Person [0:1] has Death Date [1:*];
Deceased Person [0:1] has Birth Date [1:*];
```

Figure 2: A portion of *data frames* for an ontological model instance of the obituary ontology. Note that "Date" applies to "BirthDate", "DeathDate", "IntermentDate", "ViewingDate", and "FuneralDate".

Consider a simplified ontological model instance of the obituary ontology in Figure 1. "Obituary" is a high-level object set name, "Funeral" is a non-lexical object set name, and "FuneralDate" is a lexical object set name. Suppose a document has a sentence:

"Funeral is on Mar 3, 1998". The sentence generates the relationship (*is on*) between "Funeral" and "FuneralDate", where "Mar 3, 1998" is the constant string for "FuneralDate" specified by the first regular expression under "Date matches [16]" in Figure 2. Furthermore, the cardinality constraint [0..1] to the right of "Funeral" in Figure 1 means that if there is a "Funeral" service, it has one date, whereas the cardinality constraint [1..*] to the left of "FuneralDate" means that a "FuneralDate" may describe one or more "Funeral" services.

In order to compute the coefficients of the vector for the ontology, we define *the number of participation* of each keyword or constant. Informally, this is just the word frequency in the data frame. Specifically, the minimum number of participation of "Date" (a lexical object) is five, e.g., five possible dates, i.e., BirthDate, DeathDate, IntermentDate, ViewingDate, and FuneralDate, in each obituary and each of these dates participates at least one time (see the last five lines of Figure 2). Similarly, the minimum number of participation of the keyword for "FuneralDate" is one. As the ontology does not define the countable maximum number of participation of each object, the number of particpation of each keyword or constant for the object is the minimum value of the cardinality constraint for the object.

Before we further discuss our categorization approach, we describe how VSM and CM determine the relevance of a Web document to a particular application ontology. In VSM, each of the keywords and constants (keyword-associated constants) defined in an application ontology is interpreted as an axis of an n-dimensional vector space. We define CV as a Category Vector for the application ontology. The coefficients of CV are the minimum values of the cardinality constraints of the keywords and constants defined in the ontology. For example, the coefficients in CV of the keyword for "FuneralDate" is 1 and the coefficient in CV of the constant for "Date" is 5.

Consider a Web document D in Figure 3, which contains multiple records of four obituaries:

Stephen Liptak

. . .

Funeral mass will be held at 10:30 a.m. on Tuesday at St. Maximillian Kolbe Church in Toms River, N.J.

. . .

Robert M. Borger

. . .

Funeral services will be held at 11 a.m. on Monday from the Kresge Funeral Home, Route 209 Brodheadsville, wit the Rev. Deborah Scheffey officiating.

. . .

Emily M. Stout

. . .

Funeral services will be immediately following the viewing at 10:30 a.m. on Tuesday at the funeral home.

. . .

William Henry "Pete" Rickards

. . .

A Masonic Service will be held on Thursday at 1 p.m. in the William H. Clark Funeral Home, 1003 Main Street, Stroudsburg.

. . .

Figure 3: Four obituaries recorded on Sunday, July 5, 1998

in *The Pocono Record* (Stroudsburg, PA)

We adopt the heuristics of [YJ98] to calculate the number of multiple records in the document, e.g., 4 is the number of multiple records calculated for D in Figure 3. Therefore there are multiple records of four obituaries. Based on the keywords and constants defined in the obituary ontology of Figure 2, we create a data record for each keyword and for each constant found in M one of the multiple records of D. Each data record has five fields: *1) a constant (the name of a keyword,) 2) the starting position of its corresponding string in the M, 3) the ending position of its corresponding string in M, 4) its length, and 5) its corresponding string found in M*. For example, in Figure 3, the three bolded words “Funeral” which appear in three of the four obituaries are associated with the KEYWORD(FuneralDate). This constitutes three data records for the three strings found. One of the three data records is

KEYWORD(FuneralDate) : 1653|1659|7|Funeral

where KEYWORD(FuneralDate) is the name of a *keyword* for the lexical object FuneralDate; 1653 and 1659 are, respectively, the *starting* and the *ending positions* of the corresponding string for the keyword found in M; 7 is the *length* of the string in M; and "Funeral" is *the corresponding string* for the keyword of FuneralDate.

We use all of the data records to form a data list relative to the document D. The data list is used to compute the average frequencies of the keywords and constants found in the multiple records in a Web document. The average frequency of K, one of the keywords (or constants), is computed as n/m, where n is the frequency of K and m is the number of multiple records in the document. In this example, KEYWORD(FuneralDate) appeared three times in the data list with four multiple records. So, its average frequency is therefore 0.75, i.e., 3/4.

We let DV denote the n-dimensional Document Vector for a Web document with respect to a particular application ontology. Recall that in CM, a cluster C is a set of Web documents that is formed based on the similarity among their corresponding n-dimensional vectors. But we are interested in Web documents with multiple records. Therefore, we must extend CM from individual documents to multiple records in each of these documents.

Let us consider a Web document D with multiple records. We extend CM by constructing S_1 the set of multiple records in D. Only the multiple records in D "represent" the document. We replace DV for the document with the term-centroid vector for S_1. Therefore, the coefficients of DV will now be the average frequencies of keywords and constants found in the set of multiple records in S_1. These frequencies can be computed from the data list.

Our program automatically performs the following process:

1. The program extracts from a particular application ontology the set of all non-lexical object set names L and the set of all high-level object set names H. The program arranges L^2 into a search engine query. The query is submitted to a search engine which returns a set of Web documents.
2. The program applies the multiple record-boundary heuristics of [YJ98] on the set of documents that are returned from the search engine. The program removes from the set of returned documents any document that does not contain multiple records.
3. The program extracts keywords and constants from each remaining document based on the set of keywords and keyword-associated constants defined in the ontology. The keywords and constants that are found in each remaining document yields different records in the data list for the document.
4. The program applies the term-centroid vector of CM to determine the coefficients of DV from the data list for each remaining document.
5. The program applies VSM to create CV for the ontology.
6. Finally, the program computes the acute angle between DV and CV [SM83].

The closer the angle between CV and DV is to zero, the more likely a Web document (represented by DV) is relevant to an application ontology (represented by CV).

[2] The program can also arrange H into a query and submit it to a search engine.

IV Contribution to Computer Science

This thesis proposes the automatic categorization of Web documents with respect to an application ontology. The approach uses application ontologies, the Vector Space IR Model, and the Clustering IR Model. Our approach enhances the precision of existing search engines in retrieving Web documents. An increase in precision has two consequences: a) it saves users' time when browsing retrieved Web documents; and (b) it means retrieved Web documents can be used as input of the data extraction tools proposed in [RL94, KSa97, Sod97, HGMC+97, ECLS98].

V Delimitations of the Thesis

This thesis will not do the following:

- Deal with a Web document unless (1) it is data rich, (2) it is narrow in breadth, and (3) it contains multiple records that might be relevant to a particular application ontology.
- Enhance application ontology.
- Consider Natural Language Processing techniques to determine whether keywords and constants appear in the same sentence in a Web document.

VI Thesis Outline

1 Introduction

1.1 The problem

1.2 Thesis organization

Estimated Length: 3 pages

2 Preliminaries

2.1 Information Retrieval

2.1.1 Term Weighting Methods

2.1.2 Advanced Information Retrieval (IR) Models

2.1.2.1 Vector Space Model

2.1.2.2 Clustering Model

2.2 Related work

Estimated Length: 10 pages

3 Proposed Categorization Approach

3.1 Input queries for Search engines

3.2 Ontological Matching using Conceptual Model

3.3 Determination of relevance using IR Models

3.3.1 Vector Space Model

3.3.2 The Term-Centroid of Clustering Model

Estimated Length: 30 pages

4 Experimental Analysis of the proposed Approach

Estimated Length: 5 pages

5 Conclusions

Estimated Length: 3 pages

VII Thesis Schedule

A tentative schedule of this thesis is as follows:

Literature Search and Reading	August - September, December 1998
	January - February 1999
Chapter 3	April, 1999
Chapter 4	May, 1999
Chapter 1, 2 & 5	June, 1999
Thesis Revision and Defense	June, 1999

VIII Bibliography

[Bri98] Sergery Brin, "Extracting Patterns and Relations from the World Wide Web," in Proceedings of the EDBT '98 Conference, 1998.

> A particular type of data may be scattered across thousands of independent information sources in many different formats. This paper considers the problem of extracting a relation for such a data type from all of these sources automatically. It presents a technique, which exploits the duality between sets of patterns and relations to grow the target relation starting from a small sample.

[CGRU97] Chandra Chekuri, Michael H. Goldwasser, Prabhakar Raghavan, Eli Upfal, "Web Search Using Automatic Classification," in Proceedings of the Sixth International WorldWide Web Conference, April 1997.

> The authors investigate the automatic classification of Web documents into pre-specified categories, with the objective of increasing the precision of Web search. The experiments conducted classify documents into high-level categories of the Yahoo! taxonomy. The results indicate that Web classification and search tools must compensate for artifices such as Web spamming that have resulted from the very existence of such tools.

[CL96] Jim Cowie, Wendy Lehnert, "Information Extraction," Communications of the ACM, Vol. 39, No. 1, January 1996.

> This paper provides an overview of the field of "information extraction", which is a subfield of natural language processing (NLP). The authors take a simpler approach to NLP in syntactic sentence analysis, but focus more on real-world texts.

[CR95] F. Crestani, C.J. van Rijsbergen, "Probability Kinematics in Information Retrieval," in Proceedings of the 18th International Conference on Research and Development in Information Retrieval (ACM SIGIR'95), July 1995.

> This paper discusses the dynamics of probabilistic term weights in different IR retrieval models. Four different models, namely Retrieval by Joint Probability, Retrieval by Conditional Probability, Retrieval by Logical Imaging, and Retrieval by General Logical Imaging, are presented. The former two are classical probabilistic IR models, the latter two are based on a logical technique of evaluating the probability of a conditional called Imaging. The paper also analyses the transfer of probabilities occurring in the representation space at retrieval time for these four models and compare their retrieval performance using classical test collections.

[DADA97] R. Dolin, D. Agrawal, L. Dillion, A. El Abbadi, "Pharos: A Scalable Distributed Architecture for Locating Heterogeneous Information Sources," in

Proceedings of the Conference on Information and Knowledge Management (CIKM'97), 1997.

This paper presents the design of Pharos: a scalable distributed architecture that allows users to locate heterogeneous information sources over the Internet. The system incorporates a hierarchical metadata structure into a multi-level retrieval system. Queries are resolved through an iterative decision-making process.

[ECLS98] David W. Embley, Douglas M. Campbell, Stephen W. Liddle, Randy D. Smith, "Ontology-Based Extraction and Structuring of Information from Data-Rich Unstructured Documents," in Proceedings of the Conference on Information and Knowledge Management (CIKM'98), November 1998, Pages 52-59.

This paper presents an ontology-based system to extracting and structuring information from unstructured documents that are data rich and narrow in ontological breadth. The authors parse the application ontology, which describes the objects, relationships, and constraints in a domain of interest, to generate recognition rules for constants and context keywords and to extract structural and constraint information. Given the generated rules and an unstructured document, the authors apply a recognizer to extract the constants and keywords, and then apply a structure builder to match constant values with attributes, to associate attribute-value pairs as relations, and to populate a generated database schema with the extracted data according to the constraints of the application ontology.

[ECJ+98] David W. Embley, Douglas M. Campbell, Yuan Jiang, Yiu-Kai Ng, Randy D. Smith, Stephen W. Liddle, Dallan Quass, "A Conceptual-Modeling Approach to Extracting Data from the Web," in Proceedings of the 17th International Conference on Conceptual Modeling (ER'98), November 1998, Pages 78-91.

This paper introduces a conceptual-modeling approach to extract and structure data. The approach is based on an ontology – a conceptual model instance – that describes the data of interest, including relationships, lexical appearance, and context keywords. By parsing the ontology, the system can automatically produce a database scheme and recognizers for constants and keywords, and then invoke routines to recognize and extract data from unstructured documents and structure it according to the generated database scheme.

[Embley98] Embley, David W., Object Database Development: Concepts and Principles, Addison Wesley Longman, 1998.

This book illustrates concepts and principles of Conceptual Modeling and how an ontological model instance is created. It defines different kinds of objects, such as lexical objects, non-lexical objects, and high-level objects etc., and explains how these objects are related to one another.

[HGMC+97] Joachim Hammer, Hector Garcia-Molina, Junghoo Cho, Rohan Aranha, Arturo Crespo, "Extracting Semistructured Information the Web," in Proceedings of the First Workshop on Management of Semistructured Data, May 1997.

This paper describes a configurable tool for extracting semistructured data from a set of HTML pages and for converting the extracted information into database objects. The extractor in this paper provides a currently missing link between the Web and the applications, which have no direct access to the Web data.

[IT-AI95] Makoto Iwayama, Takenobu Tokunaga, "Hierarchical Bayesian Clustering for Automatic Text Classification," in Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI95-2). Montreal, Quebec, Canada, 1995. 1322-1327.

Text classification, the grouping of texts into several clusters, has been used as a means of improving both efficiency and the effectiveness of text retrieval/categorization. This paper proposes a hierarchial clustering algorithm that constructs a set of clusters having the maximum Bayesian posterior probability, the probability that the given texts are classified into clusters. This algorithm is called Hierarchical Bayesian Clustering (HBC). HBC has the following advantages: 1) HBC can re-construct the original clusters more accurately than do other non probabilistic algorithms, 2) when a probabilistic text categorization is extended to a cluster-based one, the use of HBC offers better performance than does the use of non-probabilistic algorithms.

[IT-ANLP94] Makoto Iwayama, Takenobu Tokunaga, "A Probabilistic Model for Text Categorization: Based on a Single RandomVariable with Multiple Values," in Proceedings of the Fourth ACL Conference on Applied Natural Language Processing (ANLP 94) (13--15 October 1994 Stuttgart), October 1994.

This paper proposes a new probabilistic model for text categorization, that is based on a Single random Variable with Multiple Values (SVMV). Compared with previous probabilistic models, this model has the following advantages: 1) it considers within-document term frequencies, 2) considers term weighting for target documents, and 3) is less affected by having insufficient training cases. This model is verified superior over the others in the task of categorizing new articles from the "Wall Street Journal".

[IT-IR95] Makoto Iwayama, Takenobu Tokunaga, "Cluster-Based Text Categorization: A Comparison of Category Search Strategies," in Proceedings of the 18th Annual International Conference on Research and Development in Information Retrieval (SIGIR'95), Seattle, WA, USA, 1995. 273-281.

This paper proposes a cluster-based search with a probabilistic clustering algorithm evaluated on two data sets. The efficiency, effectiveness, and noise tolerance of this search strategy were confirmed to be better than those of a full

search, a category-based search, and a cluster-based search with non-probabilistic clustering.

[IT-TR94] Makoto Iwayama, Takenobu Tokunaga, "Text Categorization Based on Weighted Inverse Document Frequency," Technical Report 94-TR0001 in Computer Science Department at Tokyo Institute of Technology, Tokyo, Japan, March 1994

This paper proposes a new term weighting method called weighted inverse document frequency (WIDF). WIDF is an extension of IDF (inverse document frequency) to incorporate the term frequency over the collection of texts. WIDF of a term is given by dividing the frequency of the term in the text by the sum of the frequency of the term over the collection texts. WIDF is applied to the text categorization task and proved to be superior to the other methods. The improvement of accuracy on IDF is 7.4% at the maximum.

[ITN96] Michiaki Iwazume, Hideaki Takeda, and Toyoaki Nishida, "Ontology-based Information Gathering and Text Categorization from the Internet," in Proceedings of the Ninth International Conference in Industrial and Engineering Applications of AI and Expert Systems (IEA/AIE – 96), Pages 305-314, 1996.

This paper describes a new method of information gathering and text categorization using ontologies. A system called IICA (Intelligent Information Collector and Analyzer) was implemented to help people to acquire knowledge from information resources on the wide-area network by gathering information and categorizing texts. Experimental results show that the ontology-based approach enables people to use heterogeneous information resources on the wide-area such as the WWW and the network news.

[KS98] D. Konopnicki, O. Shumueli, "Bringing Database Functionality to the WWW." Submitted for Publication, 1998.

This paper introduces new ideas and mechanisms for "importing" database techniques and functionalities to the WWW. A hierarchical, object oriented, abstract data model for the WWW is proposed to enable the definition of a powerful and optimizable WWW standard query language. Query processing techniques designed for the WWW are a crucial element in harnessing the WWW.

[Ksa97] Daphne Koller, Mehran Sahami, "Hierarchically Classifying Documents Using Very Few Words," in Proceedings of the Fourteenth International Conference on Machine Learning (ML-97), Pages 170-178, Nashville, Tennessee, July 1997.

The proliferation of topic hierarchies for text documents has resulted in a need for tools that automatically classify new documents within such hierarchies. Existing classification schemes which ignore the hierarchical structure and treat the topics as separate classes are often inadequate in text classification where there is a large number of classes and a huge number of relevant features needed to distinguish

between them. This paper proposes an approach that utilizes the hierarchical topic structure to decompose the classification task into a set of simpler problems, one at each node in the classification tree. Each of these smaller problems can be solved accurately by focusing only on a very small set of features, those relevant to the task at hand.

[MLY+98] Weiyi Meng, King-Lup Liu, Clement Yu, Xiaodong Wang, Yuhsi Chang, Naphtali Rishe, "Determining Text Databases to Search in the Internet," in Proceedings of the 24th VLDB Conference, New York, USA, Pages 14-25, August 1998.

Text data in the Internet can be partitioned into many databases naturally. Efficient retrieval of desired data can be achieved if we can accurately predict the usefulness of each database. This paper introduces two new methods for estimating the usefulness of text databases. For a given user query, the usefulness of a text database in this paper is defined to be the number of documents in the database that are sufficiently similar to the query. Such a usefulness measure enables naïve-users to make informed decision about which databases to search. Techniques to determine the best threshold for a given local database are also introduced. The threshold used by a local database to determine whether a document is potentially useful may be different from that used by the global database.

[PPR96] Peter Pirolli, James Pitkow, Ramana Rao, "Silk From a Sow's Ear: Extracting Usable Structures from the Web," in Proceedings of the ACM SIGCHI Conference on Human Factors in Computing, 1996.

This paper presents techniques that utilize both the topology and textual similarity between items as well as usage data collected by servers and page meta-information like title and size. Linear equations and spreading activation models are employed to arrange web pages based upon functional categories, node types, and relevancy.

[PSHD96] Peter Pirolli, Patricia Schank, Marti Hearst, Christine Diehl, "Scatter/Gather Browsing Communicates the Topic Structure of a Very Large Text Collection," in Proceedings of the ACM SIGCHI Conference on Human Factors in Computing, 1996.

Scatter/Gather is a cluster-based browsing technique for large text collections. Users are presented with automatically computed summaries of the contents of clusters of similar documents and provided with a method for navigating through these summaries at different levels of granularity. The aim of this technique is to communicate information about the topic structure of very large collections. This paper provides the study of the effect of Scatter/Gather, a simple pure document retrieval tool, on the incidental learning of topic structure. When compared with simple keyword search, results show that Scatter/Gather induces a more coherent conceptual image of a text collection, a richer vocabulary for constructing search

queries, and communicates the distribution of relevant documents over clusters of documents in the collection.

[RL94] Ellen Riloff, Wendy Lehnert, "Information Extraction as a Basis for High-Precision Text Classification," in ACM Transactions on Information Systems, July 1994, Vol. 12, No. 3, Pages 296-333.

The authors describe an approach to text classification that represents a compromise between traditional word-based techniques and in-depth natural language processing. This approach uses a natural language processing task called information extraction as a basis for high-precision text classification. Three algorithms that use varying amounts of extracted information to classify texts are: (a) the relevancy signatures algorithm, (b) the augmented relevancy signatures algorithm, and (c) the case-based text classification algorithm. (a) uses linguistic phrases, (b) uses phrases and local context, and (c) larger pieces of context. Relevant phrases and contexts are acquired automatically using a training corpus. These algorithms are evaluated on two test sets from MUC-4 corpus. All three algorithms achieved high precision on both test sets, with the augmented relevancy signatures algorithm and the case-based algorithm reaching 100% precision with over 60% recall on one set. These algorithms are also compared on a large collection of 1700 texts and an automated method is described for empirically deriving appropriate threshold values. Results show that information extraction techniques support high-precision text classification and, in general, using more extracted information improves performance.

[SG98] N. Shivakumar, H. Garcia-Molina, "Finding Near-Replicas of Documents and Servers on the Web," Submitted for Publication, 1998.

This paper considers how to efficiently compute the overlap between all pairs of web documents. It reports statistics on how common replication is on the web, and on the cost of computing the above information for a relatively large subset of the web—about 24 million web pages which corresponds to about 150 Gigabytes of textual information.

[SM83] Gerard Salton, Michael J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York 1983.

This is an introductory book on modern information retrieval. It has an analysis of automatic term extraction and weighting. It shows the reader the definition of term-centroid for document classification. It also shows the reader how to measure the similarity of documents using vector similarity functions.

[Salton88] Gerard Salton, Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, New York 1988.

This book has illustrations on advanced information retrieval models, Vector Space Model, Automatic Document Classification using clustering, and Probabilistic Retrieval Model. It also has chapters related to word processing using text editing and formatting, and file access using a variety of searching algorithms.

[Sod97] Stephen Soderland, "Learning to Extract Text-based Information from the World Wide Web," in Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, August 1997.

This paper introduces Webfoot, a preprocessor that parses web pages into logically coherent segments based on page layout curs. Output from Webfoot is then passed on to CRYSTAL, a natural language processing (NLP) system that learns text extraction rules from example. Webfoot and CRYSTAL transform the text into a formal representation that is equivalent to relational database entries.

[YJ98] Yuan Jiang, "Record-Boundary Discovery in Web Documents," Master Thesis at Brigham Young University, December 1998.

This thesis has considered various heuristics on discovering record boundaries in Web documents. These heuristics are the Ontology-Matching Heuristic O, the Repeating-Tag Pattern Heuristic R, the Standard Deviation Heuristic S, the Identifiable "Separator" Tags Heuristic I, and the Highest-Count Tags Heuristic H. Experiments show that each of the combined heuristics of ORSI, ORIH, RSIH, and ORSIH has achieved a successful rate of 100 percent in finding a correct record separator in each of 100 experimental Web documents.

IX Artifacts

The program which implements the retrieval of relevant Web pages using the proposed categorization approach will be written in Perl and Java on an Hewlett Packard C200 workstation.

X Signatures

This proposal, by Linus W. Kwong, is accepted in its present form by the Department of Computer Science of Brigham Young University as satisfying the proposal requirement for the degree of Master of Science.

_____

Yiu-Kai (Dennis) Ng, Committee Chairman

_____

Douglas M. Campbell, Committee Member

_____

Theodore A. Norman, Committee Member

_____

Scott N. Woodfield, Graduate Coordinator