# Performing Binary-Categorization on Multiple-Record Web Documents Using Information Retrieval Models and Application Ontologies

Linus W. Kwong
Yiu-Kai Ng
Department of Computer Science
Brigham Young University
Provo, Utah 84602, U.S.A.
{ng, kwongl}@cs.byu.edu

**Abstract**

To retrieve Web documents of interest, most of the Web users rely on Web search engines. All existing search engines provide query facility for users to search for the desired documents using search-engine keywords. However, when a search engine retrieves a long list of Web documents, the user might need to browse through each retrieved document in order to determine which document is of interest. We observe that there are two kinds of problems involved in the retrieval of Web documents: (1) an inappropriate selection of keywords specified by the user; and (2) poor precision in the retrieved Web documents. In solving these problems, we propose an automatic binary-categorization method that is applicable for recognizing multiple-record Web documents of interest, which appear often in advertisement Web pages. Our categorization method uses application ontologies and is based on two information retrieval models, the Vector Space Model (VSM) and the Clustering Model (CM). We analyze and cull Web documents to just those applicable to a particular application ontology. The culling analysis (i) uses CM to find a virtual centroid for the records in a Web document, (ii) computes a vector in a multi-dimensional space for this centroid, and (iii) compares the vector with the predefined ontology vector of the same multi-dimensional space using VSM, which we consider the magnitudes of the vectors, as well as the angle between them. Our experimental results show that we have achieved an average of 90% recall and 97% precision in recognizing Web documents belonged to the same category (i.e., domain of interest). Thus our categorization discards very few documents it should have kept and keeps very few it should have discarded.

**Keywords**: Information retrieval, application ontologies, binary-categorization, vector-space model, clustering model, WWW

# 1 Introduction

The World Wide Web contains abundant repositories of information written in Web documents. To retrieve Web documents of interest, most common Web users who are not knowledgeable in constructing sophisticated Web queries rely on search engines, which provide query facilities based on keywords. Partially because keyword queries posted by common Web users, even if the queries are in the form of a Boolean expression, are not precise, search engines often return long lists of Web documents as answers to these queries. If a search engine returns many irrelevant Web documents, that is, if its precision is poor, it fails to accomplish its task [CGRU97].

We observe that precision is often poor largely because users lack opportunities to specify more exactly what is wanted. In this paper we propose an approach to alleviate poor precision that allows users to specify more precisely what is wanted. We base our approach on application ontologies [ENX01], which are conceptual-model snippets [SDUS98] of standard ontologies [Bun79]; and we use a binary-categorization technique to classify Web documents to a subset applicable to a particular ontology. After an application ontology has been established, our binary-categorization process is fully automatic; anecdotal evidence suggests that a reasonable application ontology of the kind we use can be created in one or two dozen person-hours. Since our approach does have this associated cost, it is not appropriate for all Web categorization. It is appropriate, however, for application inquiries that are either more long-lasting than fleeting desires to surf for something interesting or that fall into common applications for which the cost of application-ontology development can be amortized over large user groups. We believe that many, if not most, serious inquiries fall into these two categories. Furthermore, our categorization approach is applicable to Web documents regardless of their structures, formats, and contents, which is essential in the dynamic environment of the Web.

Our work reported here is also partly motivated by our success in using application ontologies to extract information from multiple-record Web documents. These Web documents usually contain a sequence of chunks of textual information, each of these chunks constituting a "record" [ECJ+99]. Thus a record represents an instance of data item specified in an given application ontology. For example, in a multiple-record job-ads [1] Web document, there are a number of job-ads, each of which often includes degree/skill requirements and contact information, such as email address, phone, fax, and Web URL address. Each job ad is thus treated as a "record." Hence, a Web document that contains multiple job-ads is a multiple-record job-ads Web document. We also consider categories defined by application ontologies that are data-rich and narrow in ontological breadth. In the applications we have tried—automobile want-ads, obituaries, job-ads, real estate, stocks, musical instruments, precious gems, games, personals, and computer monitors—we have achieved good fact-extraction recall (75-100%) and very good fact-extraction precision (ordinarily 90% or better) [ENX01]. The reason why we consider application ontologies is that they describe particular data of interest and capture the semantic relationships among data objects using a simple conceptual modeling approach. Furthermore, if we can recognize documents applicable to an ontology (as proposed in this paper), we can apply the downstream process,

[1] From now on, whenever we use the term "job ad(s)", we really mean "computer job ad(s)".

which we have already developed, to extract, structure, and archive in databases information found in these application-specific Web documents. Thus the contributions of this work have the potential to be more far-reaching than just the salient contribution of significantly increasing recall and precision for recognizing application-specific Web documents.

Our approach to document categorization is related to text classification [BB63], i.e., each application ontology can be a class. However, our work fundamentally differs from other text-classification work [LSC96, KS97, BGG+99]. Text-classification systems usually attempt to place articles, such as newspaper articles, in predefined classes according to the subject matter of the article, whereas our approach seeks to do "high-precision" text classification (with similarities to [RL94]) in which we not only determine whether a listing of ads, such as the classified ads in a newspaper, contains any ads that include any element as predefined in an application ontology, but we also determine whether particular elements of interest are also present in each ad. We further assume that a subsequent process can extract the information and create a database record for each ad. Furthermore, our binary-categorization approach is an enhancement of our earlier work [ENX01], in which three different heuristics were proposed in recognizing multiple-record Web documents that belong to a particular application ontology. The experimental results show that our approach performs at least as good as, if not much better than, [ENX01]. Our approach is more elegant than the one adopted by [ENX01] because it is simplier and more cost-effective in solving the same problem due to the simplicity of the Vector Space Model and Clustering Model, two information-retrieval models adopted by our categorization approach.

The specifics of our approach, which we present in this paper, are as follows.

1. We specify a conceptual-model ontology for our application of interest. Section 2.1 briefly describes the model we use for specifying application ontologies and provides an example to which we refer throughout the paper to illustrate our ideas.

2. Given a list of Web documents, for example, retrieved by a keyword query submitted to a search engine, we first identify those documents that contain multiple records—i.e., chunks of (semi-)structured or unstructured data with data elements corresponding to the application ontology. To identify multiple records, we use a technique as discussed in Section 3.

3. Using the application ontology, we identify constants and keywords for the specified objects (i.e., specified items of interest, such as name, price, location, etc.) of the application that often appears within the records in a relevant Web document of the list, and we count how many we have for each object in each record. Using the Clustering Model (CM) in information retrieval, we then compute the $n$-dimensional ($n \geq 1$) centroid vector for each Web document $doc_D$, where $n$ is the number of objects (dimensions) specified in the application ontology. Section 4.3 describes this process.

4. Based on the constraints of the ontology, which provide the expected number of constants and keywords for each object of interest per record, we compute the expected $n$-dimensional vector for the application ontology. This task is performed only once for each application ontology. Section 4.2 describes this process.

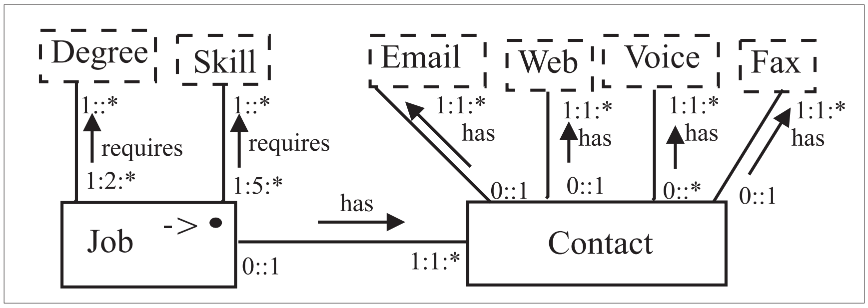



Figure 1: The graphical representation of the job-ads application ontology.

5. Finally, we measure the relevance of $doc_D$ in Step 3 for an application ontology using an extended Vector Space Model (eVSM). eVSM considers not only the difference in direction, which is the similarity metric of VSM, but also the difference in magnitudes of the vectors computed in Steps 3 and 4. When there are "small" differences in both directions and magnitudes, we consider $doc_D$ to be relevant for the application. Section 5 describes these details.

In Section 6 we discuss our experimental results—which are in the 90% range for both recall and precision for the chosen application ontologies. In Section 7, we give a conclusion.

# 2 Preliminaries

## 2.1 Application ontology

An application ontology is a conceptual model instance that describes a real-world application of interest and consists of two components: (i) an *ontological model instance*, which describes sets of objects, sets of relationships among objects, and constraints over object and relationship sets; and (ii) *data frames*, which formally define the constants and keywords of each object set in part (i). A data frame for an object set defines the lexical appearance of *constant* objects, such as $50,000, 70%, and "software development," within the object set, which can be an integer, a string of characters, a Boolean value, etc. *Keyword* objects, on the other hand, indicate the *presence* of an object in a lexical object set. For example, "died" and "passed away" may be context keywords for *Death Date*; "buried" may be a context keyword for *Interment*.

Figure 1 shows a graphical representation of the `job-ads` application ontology. Figure 2 shows part of a textual representation for this ontology, including the name of the ontology ("Jobs" on line 1), the object and relationship sets and the constraints (lines 2-9), and some partial data frames (lines 10-50). A data frame for an object set specifies the object-set name (e.g., "Skill" on line 10) and the keyword `matches` to designate that the structure is a data frame followed by regular expressions for defining `constants` that appear as values in the data frame (if any) and `keywords`. The system we have built accepts application ontologies such as the one depicted in Figure 2, but often we refer to the graphical abstraction of an ontology such as the one depicted in Figure 1.

```
1: Application Ontology: Jobs;
2: Job [→ object];
3: Job [1:5:*] requires Skill [1::*];
4: Job [1:2:*] requires Degree [1::*];
5: Job [0::1] has Contact [1:1:*];
6: Contact [0::1] has Email [1:1:*];
7: Contact [0::1] has Web [1:1:*];
8: Contact [0::*] has Voice [1:1:*];
9: Contact [0::1] has Fax [1:1:*];
10: Skill matches
11: constant
12: – Languages
13:     { extract "\bbasic\b";},
14:     ...
15: – Databases
16:     { extract "\bsql\b";},
17:     ...
18: – Operating systems
19:     { extract "\bunix\b";},
20:     ...
21: – Web & networking tools
22:     { extract "\bweb\b";},
23:     ...
24: – Other software & Hardware
25:     { extract "\bperl\b";},
26:     ...
27: end;
28:
29: Degree matches
30: constant
31: – College/university
32:     ...
33: – Certificate
34:     ...
35: end;
36:
37: Email matches
38:     ...
39: Web matches
40:     ...
41: Voice matches
42: constant
43: { extract "(\d{3}([-/]\s*...";},
44:
45: keyword "\bph(one)?\b";
46: end;
47:
48: Fax matches
49:
50: keyword "\bfa?x\b";
51: end;
```

Figure 2: Jobs.onto, the job-ads application ontology.

An object set in an application ontology represents a set of *objects*, which may either be *lexical* [2] or *non-lexical.* [3] We describe the constant lexical objects and keywords for an object set by regular expressions using Perl syntax. Applying to a document, the `extract` (`keyword`, respectively) clause in a data frame identifies a string in the document that matches a regular expression to be extracted as a constant (`keyword`, respectively). The regular expression "\bph(one)?\b" for the `keyword` clause on line 45 locates strings such as "ph," "Ph," and "Phone" in a document and associates them as keywords for the object set *Voice* in a job ad. The nonlexical object set *Job* is designated as the *object set of interest*, denoted by "[→ object]" on line 2.

A line connecting rectangles in Figure 1 represents a relationship set among the connected object sets. A labeled reading-direction arrow of a relationship-set, along with the names of the connected object sets, provides a name for a relationship set. *Job requires Degree*, for example, names the relationship set between *Job* and *Degree* in Figure 1. In the textual representation these names represent both the relationship set and the connected object sets. Line 4 in Figure 2 represents the relationship set *Job* (a non-lexical object set) *requires Degree* (a lexical object set).

The *min:ave:max* triples near the connections between object sets and relationship sets in Figure 1 are *participation constraints. Min* designates the minimum number of times an object in the object set can participate in the relationship set; *ave*, if it exists, designates the average number of times the object is expected to participate in the relationship set; and *max* designates the maximum number of times the object can participate in the relationship

[2]A lexical object is identified by dashed boxes, as shown in Figure 1; or data frames with constant declarations, as shown in Figure 2.

[3]A non-lexical object is identified by solid boxes, as shown in Figure 1; or data frames without constant declarations in the textual representation of the application ontology.

set, with '*' designating an unknown but finite number of times that is larger than the *min*. The participation constraint on *Job* for *Job requires Skill*, for example, specifies that a job require at least one skill, that a job normally require 5 skills on the average, and that there be no specified maximum for the number of skills a job may require.

For any application ontologies that we have created, we obtained participation constraints by examining corresponding multiple-record documents extracted from different regions covering the United States. From each of these documents we simply counted by hand and obtained minimum, average, and maximum values for each object set in each relationship set and normalized the values for the corresponding application ontology.

## 2.2 Information retrieval models

According to [CLvR98], the most significant problem in Information Retrieval (IR) is to find the set of relevant documents that are sought by the user among a large collection of documents. In order to solve this problem, we represent each document in the collection by such properties as selected keywords and constants that appear in the document. Hereafter, we determine the relevance of the document to a user's request by using different IR models. Well-known IR models include the Vector Space Model (VSM) and the Clustering Model (CM) [BYRN99].

In VSM a document $D$ (a query $Q$, respectively) can be represented as a vector using the selected terms whose numbers of occurrences in $D$ ($Q$, respectively) determine the coefficients of the vector. Selected terms of $D$ must be carefully chosen to represent $D$ using the *term weighting system* [Sal89]. In this system, the importance of a term in a document is measured by its *term weight*. We adopt the *term frequency* of a term as the term weight in our binary-categorization approach.

*Term frequency* is a measure of the weight of a term $t$ in a document $d$, denoted $TF(d, t)$, which is determined by the number of occurrences of $t$ in $d$. Terms do not occur with uniformly high frequencies in $d$. The weight of $t$ in $d$, denoted $W(d, t)$, is $TF(d, t)$. According to [Sal89], if $W(d, t) > T$, where $T$ is a threshold frequency, $t$ is chosen to partially represent $d$, and the larger $W(d, t)$ is, the higher the recall ratio[4] of $d$ is.

In this paper, we adopt the following two IR models for categorizing relevant documents with respect to an application ontology:

The Vector Space Model (VSM). VSM is one of the IR models that use vectors to represent documents (queries created by users, respectively) in an $n$-dimensional vector space, where $n \geq 1$. This model is used to rank a given set of documents according to various degrees of similarities among the documents with respect to a particular query based on the matching terms in the respective vectors [SM83]. The degree of similarity between a document and a query is measured by using a similarity function defined in VSM that will be further explained.

[4]The recall ratio of documents is the number of relevant documents retrieved from a collection divided by the total number of relevant documents in the collection.

The Clustering Model (CM). Similar to VSM, the Clustering Model (CM) uses vectors to represent documents and queries in a vector space. CM, however, creates clusters of documents to define different categories of documents. To represent a cluster $C$, CM constructs an $n$-dimensional *term-centroid vector* $V$, where $V$ and any document/query vector in VSM are in the same $n$-dimensional vector space, where $n \geq 1$. Unlike VSM, which uses vector to represent a document, $CM$ uses $V$ to represent a cluster of documents. The $j$th coefficient of $V$ is the *average weight* of the $j$th ($j \geq 1$) term in all the documents of $C$. The *average weight* of the $j$th term in $C$ is computed by the sum of the weights of the $j$th term in all documents in $C$ divided by the number of documents in $C$.

# 3 Record extraction

In our binary-categorization approach, one of the first steps in obtaining documents of interest from the Web is done by identifying documents containing multiple records using our record-boundary detection approach [EJN99]. A record is a chunk of data that represents an instance of (some) terms specified in a given application ontology. Our record-boundary detection approach is applicable to both Web documents created by using an automated tool and to documents written manually. When an automated tool is used, there is a high likelihood that the use of tags in documents is consistent across records. For Web documents with multiple records that are generated manually, it is usually true that the creator of these documents follows certain patterns, although these patterns are not necessarily rigid or well-formed. To deal with inconsistency within a Web document and variation across individually consistent documents, we use multiple heuristics to identify record separators as defined in our detection approach. The experimental results of applying different heuristics (discussed below) on randomly chosen sets of Web documents show a high success rate in determining the correct record separators [EJN99]. The results support our claim that the record-boundary detection approach is reliable and general.

To discover the record separator, we independently rank the candidate tags [5] in a Web document, such as an HTML, XML, or SGML document, using five different heuristics: *highest count* heuristic (ranking the candidate tags on the number of occurrences), *identifiable separator* heuristic (using a predetermined list of ranked record separator tags), *standard deviation* heuristic (assuming that records are similar in size. Candidate tags are ranked in order of the standard deviation of included plain text between identical tags), and *repeating pattern* heuristic (i.e., divisions between records are formed by several tags that consistently occur in the same order), and *ontology matching* heuristic (based on the ontological content of a record such that our ontology tells which fields of a record are expected to occur once and only once). The detailed discussion of using these heuristics in detecting record boundary that allows us to determine the number of records in a Web document can be found in [EJN99].

[5] A tag plays the role of record separator.

# 4 Application-Ontology vector

After a set of multiple-record Web documents is determined, we proceed to find out which documents actually belong to the application ontology of interest. In our binary-categorization approach, we adopt the VSM and CM models and treat a query and documents in VSM as an ontology and multiple-record Web documents, respectively. Hence, given an application ontology, we first determine the terms and their corresponding weights by using the predefined constants, keywords, and the cardinality (participation) constraints of their lexical object sets in the ontology. Using these terms and their weights, we are able to construct the ontology vector of the application ontology (as discussed in Section 4.2).

## 4.1 Automatic extraction of constants, keywords, and names

We have developed a tool to automatically extract constants and keywords, which define the corresponding lexical object set, to construct the ontology vector from an application ontology file. Since there can be more than one constant or keyword of a lexical object set, one or more entries in the ontology file are used to define constants and keywords of a lexical object set in the ontology.

In the ontology file, the lexical values of *constants* in a lexical object set are expressed by regular expressions, each of which is embedded by two double quotes. Each of these regular expressions is prefixed by the string "extract" and is located underneath the line(s) with the label "**constant**." The lexical values of *keywords* in the data frame of a lexical object set are also expressed by regular expressions, each of which is also inserted between two double quotes. These regular expressions, however, are prefixed by the label "**keyword**." Consider the application ontology, jobs.ont, in Figure 2 again. Different notations in a regular expression are used to define the lexical values of a constant or a context keyword. For example, the expression "\bbasic\b" on line 13 defines the constant string "*basic*," which stands for the programming language *Basic*. The expression "$\backslash bfa?x\backslash b$" on line 50 defines the keywords of FAX. "\bbasic\b" and "\bsql\b" (on line 16) define the lexical values of *basic* and $SQL$, respectively, that belong to the lexical object set $Skill$.

The constant on line 43 and the keyword on line 45 define the data frame of the lexical object set "Voice" between lines 41 and 46.[6] The relationship set between an object set and another object set is also defined in the ontology file, such as "Job [1:5:*] requires Skill [1::*]" on line 3 in Figure 2.

Since there exist a number of lexical values of constants and keywords that define an object set, we extract all the lexical values that represent constants and keywords of an object set from the application ontology file, and use each lexical object set name or keyword as the name of a coefficient in the ontology vector. For example, we extract "$Skill$," "$Voice$," and "$KEYWORD(Fax)$" as three different names in the ontology vector of the job-ads application ontology.

[6] Recall that constants and keywords in a data frame together define an object set. Hence, the data frame of an object set in an ontology file constitutes the definition of the object set. If there are $n$ different object sets in an application ontology, then there are $n$ distinct data frames in the corresponding ontology file.

## 4.2 Ontology vector (OV) construction

Based on the object sets and the cardinality constraints of the object sets specified in an application ontology file, we construct the corresponding ontology vector $OV$. The construction process of $OV$ includes identifying (i) the *names* of the coefficients of $OV$, which are lexical object set names or the keywords of lexical object set names (as discussed in Section 4.1), and (ii) the coefficients of $OV$. The coefficients of $OV$ represent an instance of the ontology. We determine the *coefficients* of $OV$ by using the cardinality constraints of the object sets to which constants and keywords belong. Given the relationship set $R$ associated with an object set $O$, the *expected number* of participations of $O$ in $R$ is determined by $[min : average : max]$ of $O$, which denotes the minimum, the average, and the maximum number of participations of $O$ in $R$. (In the ontology file, $[min : average : max]$ of $O$ in $R$ is specified on the same line that includes the definition of $R$.) If the *average* participation constraint of $O$ in $R$ exists, then the expected number of participations of $O$ in $R$ is the *average*. Otherwise, if $max$ is not a '*', then the expected number of participations of $O$ in $R$ is $max$. By default, the expected number is $min$.

Consider again the ontology file in Figure 2. The relationship set "*Job*[1:5:*] requires that *Skill*[1::*]" on line 3 be the relationship set between the object sets *Job* and *Skill*. Since the *average* value in [1:5:*] to the right of Job exists, it is chosen as the expected number of participations of Job in the relationship set "Job requires Skill." This means that on the average, five different skills are listed for each job ad in a Web document. (Note that the minimum and maximum values, i.e., 1 and * as shown in [1::*] to the right of skill in Figure 2, indicate that if a particular skill appears in any job ad, it appears at least once and up to a finite number of times among all the job-ads in a Web document.)

Given the job-ads application ontology file in Figure 2, the ontology vector $OV$ of the application ontology is constructed as follows. The identified names of the coefficients in $OV$ are $Skill$, $Degree$, $Email$, $Voice$, $Web$, $Fax$, $KEYWORD(Fax)$, and $KEYWORD(Voice)$. Among all these names, $Email$, $Voice$, $Web$, $Fax$, $KEYWORD(Fax)$, and $KEYWORD(Voice)$ are constants and keywords whose corresponding lexical object sets are in the relationship sets with the non-lexical object set *Contact*. The maximum value of the cardinality constraint of *Contact* in each of these relationship sets is 1, and the average values of cardinality constraints of its connected object sets and keywords are all 1s. On the other hand, the average values of the cardinality constraint of *Job* in the relationship sets with $Skill$ and $Degree$ are 5 and 2, respectively. The data frames of these two lexical object sets are defined between lines 10 and 27 for $Skill$ and between lines 29 and 35 for $Degree$ as shown in Figure 2. Hence, the weights (i.e., the expected number) of $Skill$, $Degree$, $Email$, $Voice$, $Fax$, and $Web$, and the keyword $KEYWORD(Fax)$ and $KEYWORD(Voice)$ are 5, 2, 1, 1, 1, 1, 1, and 1, respectively, which are the coefficients of $OV$ for the job-ads application ontology.

## 4.3 Document vector ($DV$)

The Vector Space Model (VSM) represents a document using an $n$-dimensional document vector, $n \geq 1$. In our binary-categorization approach, each multiple-record Web document is represented as a vector in VSM. After constructing the ontology vector $OV$ for a particular application ontology $O$, we proceed to build the document vector $DV$ to represent each multiple-record Web document $D$ and determine whether $D$ is relevant to $O$. (Note that we construct $OV$ only one, even though we must construct the document vector for each multiple-record Web document one by one in order to compare it with $OV$ to determine its similarity with $O$.)

The order of the names and their coefficients of $DV$ corresponds to the order of names and their coefficients of $OV$. We determine each coefficient of $DV$ by the weight of its corresponding lexical object set in $DV$, which is based on the assumption that each multiple record in $D$ is treated as a document in a cluster as defined in the Clustering Model ($CM$). We construct $DV$ by extracting keywords and constants in $D$ according to the keywords and constants, which define the lexical object sets whose names are specified in $OV$, and compute the coefficients of $DV$ based on $CM$. [7]

We define a *term* in a multiple-record Web document as either a constant or keyword in a lexical object set. The more frequently the anticipated constants and keywords occur in a Web document $D$, the larger the magnitude of its document vector ($DV$) is. In our binary-categorization approach, the *weight* of a (keyword in a) lexical object set in $DV$ is determined by the *frequencies* with which its corresponding keywords and constants appear in a Web document. The weight of a (keyword in a) lexical object set in $DV$ is independent of the weights of other (keywords in) lexical object sets in $DV$. We further illustrate below how to use term frequencies to determine the weight of a (keyword in a) lexical object set in $DV$.

Let us consider the lexical object set "*FuneralDate*" in a simplified obituary application ontology with its only keyword, *Funeral*. Let $D$ be the document with four obituaries, as shown in Figure 3. In the figure, the first record $a$ is specified between lines 1 and 4, the second record $b$ is between lines 5 and 9, the third record $c$ is between lines 10 and 13, and the fourth record $d$ is between lines 14 and 18. The (bolded) keyword "**Funeral**" (in *FuneralDate*) appears once in each of the records $a$, $b$, and $c$. We compute the weight of *KEYWORD(FuneralDate)* in $DV$ based on its frequency of occurrence in $D$, which is 3. Since there are four obituaries in $D$, the weight of *KEYWORD(FuneralDate)* in $DV$ is computed by using the average frequency of appearance of "Funeral" in $a$, $b$, $c$, and $d$, i.e., 3/4 or 0.75, which is close to the weight of *KEYWORD(FuneralDate)*, i.e., 1, in $OV$ of the simplified obituary ontology.

Since we consider each Web document $D$ with multiple records, we adopt CM to create the cluster, which is the set of multiple records in $D$, to represent $D$. We use the *term-centroid vector* to represent the cluster $C$. Each coefficient in the *term-centroid vector* is the average frequency of appearance of particular keywords or constants belonging to the corresponding (keyword in a) lexical object set. This average frequency is computed as $m/n$, where $m$ is the frequency of appearance of a keyword or constant in $C$ and $n$ is the

[7] Note that normalizing the frequency of keywords and constants belonged to the same lexical object set that appear in each document is unnecessary. This is because the degree of similarity between the document and the predefined ontology is straightly based on the frequence of appearance of elements in the corresponding lexical object sets.

1: David Larson
2: …
3: **Funeral** mass will be held at 10:30 a.m. on Tuesday at St. Maximillian Kolbe Church in Toms River, N.J.
4: …
5: Linda W. Boyd
6: …
7: **Funeral** services will be held at 11 a.m. on Monday from the Kresge Funeral Home, Route 209 Brodheadsville, wit the Rev.
8: Stephen Oaks officiating.
9: …
10: Scott E. Jacks
11: …
12: **Funeral** services will be immediately following the viewing at 10:30 a.m. on Tuesday at the funeral home.
13: …
14: Patricia Beth Anderson
15: …
16: A Masonic Service will be held on Thursday at 1 p.m. in the William H. Clark Funeral Home, 1003 Main Street,
17: Stroudsburg.
18: …

Figure 3: Obituaries recorded on Sunday, July 5, 1998 issue of The *Pocono Record* (Stroudsburg, PA). Note that names of the deceased persons have been modified to protect the identity of each deceased person.

number of multiple records in $D$. Hence we compute the weight of a (keyword in a) lexical object set in the document vector $DV$, i.e., a coefficient of $DV$, as the average frequency of appearance of its corresponding constants or keyword appeared in $C$, which yields the corresponding coefficient of $DV$.

Consider the multiple-record Web document $D$ that includes 18 job ads in Figure 4. The names of the coefficients of $DV$ for the job-ads application ontology are $Skill$, $Degree$, $Email$, $Voice$, $Web$, $Fax$, $KEYWORD(Fax)$, and $KEYWORD(Voice)$, which are the same as in the ontology vector $OV$ (see Section 4.2). The coefficients of $DV$ are the average frequencies of keywords and constants defined in the corresponding (keywords in) lexical object sets found in $D$ and as shown in Table 1.

# 5 Document relevance

We determine the relevance of a multiple-record Web document to an application ontology by measuring the similarity of their corresponding vectors $DV$ and $OV$, respectively. $DV$ = $OV$ if they point to the same *direction* and have the same *magnitude*.

[SM83] claims that in VSM when $DV$ and $OV$ are exactly the same, $DV$ and $OV$ are *superimposed*, and the *angle* between them is zero. We understand that the superimposition of $DV$ and $OV$ does not necessarily mean that the magnitudes of $DV$ and $OV$ are exactly the same. The superimposition of $DV$ and $OV$ and the zero degree of the acute angle between the two vectors only indicate that $DV$ and $OV$ point to the same direction. Hence we can say that VSM measures the similarity between a document and a category (i.e., an application ontology) solely based on the difference in directions between their corresponding vectors. We, however, determine the similarity between a document and an ontology

1: US-TX-DFW Electrical Project Manager, 315
2: . . .
3: JOB DESCRIPTION:
4: * Field inspection and verification of electrically-related issues.
5: * Coordinate purchased material delivery.
6: * Keep track of schedule on multiple jobs.
7: * Organize and file project papers.
8: * Some web-design work.
9: . . .
10: SKILL REQUIREMENTS:
11: * At least 5 years related experience
12: * Familiarity with Autocad 14, **Microsoft Word, Excel**, PowerPoint
13: * Electrical background preferred
14: * Knowledge of **Microsoft Access** and FrontPage helpful
15: . . .
16: EDUCATION:
17: Must be a high school graduate. **College degree** would be an asset.
18: LOCATION:
19: Dallas, Texas
20: POSITION TYPE:
21: Contract, but no CORP to CORP
22: PAY RATE: $26-28 hourly (flexible) as W2
23: CONTRACT LENGTH:
24: 11 months
25: MISC:
26: Customer requires Drug Test & Background Check
27: US Citizen or permanent resident.
28: . . .
29: RESPOND TO:
30: Please indicate the JOB NUMBER(S) you are responding to when sending your resume.
31: Organization: Universal CADworks Inc.
32: Contact: UCAD Staffing
33: **Phone: 972-480-9644**
34: **FAX: 972-480-9640**
35: Email: **jobs@ucad.net**
36: Visit our Web Site at: **www.ucad.net**
37: . . .

Figure 4: One of the 18 job-ads listed by the Universal CADworks.

using the difference of their magnitudes as well as the difference of their directions.

We demonstrate the need to consider both the differences in magnitude and direction using the following example. Let $O$ be an application ontology with its corresponding category vector $OV$ whose coefficients are 20 and 10, and let $DV$ be the document vector of Web document $D$ whose coefficients are 10 and 5, which are the frequencies of occurrence of the two corresponding terms in $D$. Based on the similarity metric of VSM [SM83], $OV$ and $DV$ point to the same direction, since the cosine of the acute angle between $OV$ and $DV$ is $1 = (20 \times 10 + 10 \times 5) \,/\, \sqrt{(20^2 + 10^2) \times (10^2 + 5^2)}$. (The formal definition of the cosine of the acute angle is given in Section 5.1.) $D$ is classified as relevant to $O$ according to VSM. However, the magnitudes between $OV$ and $DV$ reveal that the length of $DV$ is 50% of the length of $OV$, i.e. $\sqrt{(10^2 + 5^2)} = 1/2 \times \sqrt{(20^2 + 10^2)}$. This can be interpreted as that the content of document $D$ is equivalent to 50% of the expected content in ontology $O$, and hence we should not categorize $D$ into $O$ if the threshold value of relevance is set to be greater than 0.5.

We consider an Extended Vector Space Model (eVSM), which measures the similarity between a multiple-record Web document $D$ and an ontology $O$ based on the differences in direction and magnitude between their corresponding vectors, $DV$ and $OV$. eVSM is applicable in determining the relevance of $D$ to $O$ if the following two assumptions hold: 1) there exists a tool, such as the combined heuristic approach [EJN99], that can accurately compute the *number* of multiple records in $D$, and 2) the coefficients of $DV$ are the coefficients of the term-centroid vector for $D$ as computed in CM. Both assumptions hold in our binary-categorization approach. In the following sections, we explain how to

| Names of the coefficients of $DV$ | Constants & keywords in the *Electrical Project Manager* job-ads (see Line 1 in Figure 4 | Corresponding line number in Figure 4 | Frequencies of appearance of constants & keywords in the record | Frequencies of appearance of constants & keywords in all 18 job-ads | Average frequencies of appearance of constants & keywords in all 18 job-ads |
|---|---|---|---|---|---|
| Skill | Microsoft Word, Excel, Microsoft Access | 12, 14 | 3 | 50 | 2.78 |
| Degree | College Degree | 17 | 1 | 17 | 0.94 |
| Email | Jobs@ucad.net | 35 | 1 | 18 | 1 |
| Voice | 972-480-9644 | 33 | 1 | 17 | 0.94 |
| Web | www.ucad.net | 36 | 1 | 17 | 0.94 |
| Fax | 972-480-9640 | 34 | 1 | 17 | 0.94 |
| KEYWORD (Fax) | Fax | 33 | 1 | 17 | 0.94 |
| KEYWORD (Voice) | Phone | 34 | 1 | 18 | 1 |

Table 1: The average frequencies of constants and keywords found in 18 job-ads as partially shown in Figure 4.

measure the difference in directions between $DV$ and $OV$ by adopting the similarity metric of VSM. We also explain how to measure the difference in magnitudes as well.

### 5.1 Difference in directions between $DV$ and $OV$

We use the similarity metric of VSM in eVSM to measure the difference in directions between $DV$ and $OV$. In VSM, the similarity between a document $D$ and a category (i.e., an application ontology) $O$ is measured by the difference in directions between their corresponding $DV$ and $OV$, which can be computed by the acute angle between $DV$ and $OV$. [SM83] defines the Similarity Cosine function, which calculates the acute angle as $SIM_d(O, D) = \cos\theta = P/N$, where $P$ is the inner product of $DV$ and $OV$, and $N$ is the product of the lengths of $DV$ and $OV$.

In our binary-categorization approach, we adopt $SIM_d(O, D)$, whose value is bound between 0 and 1, as the cosine measure of an acute angle, i.e., $\cos\theta$, between $OV$ and $DV$ in an $n$-dimensional vector space. The value of the similarity function $SIM_d(O, D)$ for $OV$ and $DV$ is used as the first filter to eliminate $D$, a multiple-record Web document, that is categorized as irrelevant to the ontology $O$. When $\theta \cong 0$, $\cos\theta \cong 1$. To determine the fault-tolerance ratio of the measured $\theta$, we adopt a threshold $\hbar_d$ for $SIM_d(O, D)$. Based on our experimental results (see Section 6) on four different ontologies, which achieve high precision and recall in retrieving multiple-record Web documents relevant to one of the four ontologies, we set $\hbar_d$ to be 0.83. Hence, if $SIM_d(O, D) \geq 0.83$, then the constraint on the similarity of directions between their corresponding $DV$ and $OV$ is satisfied, and the document will be further considered for their similarity in magnitudes, denoted $SIM_m(O, D)$.

Consider Figure 4 again on the multiple-record Web document $D$ with 18 job-ads. The similarity between the job-ads application ontology $O$ and $D$ is first determined by the difference in directions between their corresponding vectors $OV$ and $DV$. Recall that in Section 4.2 the coefficients of $OV$ for the job-ads application ontology have been calculated as 5, 2, 1, 1, 1, 1, 1, and 1, whereas the coefficients of $DV$ for $D$ are 2.78, 0.94, 1, 0.94, 0.94, 0.94, 0.94, and 1 (see the last column of Table 1). The inner product of the coefficients of $OV$ and $DV$, which is $21.54 = 5 \times 2.78 + 2 \times 0.94 + 1 \times 1 + 1 \times 0.94 + 1 \times 0.94 + 1 \times 0.94 + 1 \times 0.94 + 1 \times 1$. The length of $OV$ is $5.91608 = \sqrt{(5^2 + 2^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2)}$, whereas the length of $DV$ is 3.7612 $= \sqrt{(2.78^2 + 0.94^2 + 1^2 + 0.94^2 + 0.94^2 + 0.94^2 + 0.94^2 + 1^2)}$. Thus, $SIM_d(O, D)$ is 0.968 $= 21.54 / (5.91608 \times 3.7612)$. Since $SIM_d(O, D) \geq 0.83$, the lengths of $OV$ and $DV$ will further be considered.

## 5.2 Difference in magnitudes between $DV$ and $OV$

In addition to measuring the difference in directions of the two vectors $DV$ and $OV$ using the VSM similarity metric, eVSM also considers the measurement on the similarity of their corresponding multiple-record Web document $D$ to the application ontology $O$ based on the difference in their magnitudes. We define $SIM_m(O, D)$ as the absolute value of $M_{DV}/M_{OV}$, i.e., $|M_{DV}/M_{OV}|$, where $M_{DV}$ is the length of $DV$ and $M_{OV}$ is the length of $OV$. A threshold $\hbar_m$ for $SIM_m(O, D)$ is adopted to determine the "acceptance" similarity in magnitudes between $OV$ and $DV$. Based on our experimental results (see Section 6), we set $\hbar_m = 0.55$. Hence, if $SIM_d(O, D) \geq 0.83$ and $SIM_m(O, D) \geq 0.55$, we claim that $D$ is relevant to $O$.

Consider Figure 4 again for the multiple-record Web document $D$ with 18 job-ads and the job-ads application ontology $O$. Since $SIM_d(O, D) = 0.968 \geq \hbar_d (= 0.83)$ as computed earlier, $D$ is further considered for its relevance to $O$. According to the length of $DV$ (i.e., 3.7612) and the length of $OV$ (i.e., 5.9161), $SIM_m(O, D) = |3.7612 / 5.9161| = 0.636 \geq \hbar_m (= 0.55)$. $D$ is therefore classified as a job-ads document.

To further demonstrate the validity of $SIM_m(O, D)$, we consider another multiple-record Web document $D$, as shown in Figure 5. Three out of the 100 resumes, submitted by people who looked for jobs, are shown in Figure 5. The three resumes are: resume $a$, entitled "Home-Based Typist & More 4 Hire!" (between lines 1 and 11); resume $b$, entitled "Number Cruncher" (between lines 13 and 21); and resume $c$, entitled "Jack of all Trades" (between lines 23 and 30). The frequencies of appearance of constants and keywords found in resumes $a$, $b$, and $c$ are shown in the 4th column of Tables 2 - 4. Note that there are variations on the appearances of constants and keywords in resumes $a$, $b$, and $c$. Based on the results as shown in the 3rd column of Table 5, the average frequencies of *Skill*, *Degree*, *Email*, *Voice*, *Web*, *Fax*, *KEYWORD(Fax)*, and *KEYWORD(Voice)* of the 100 resumes are 2.39, 0.69, 1.03, 1.21, 0.17, 0.3, 0.5, and 1.07, respectively, which are the coefficients of $DV$. Given the coefficients of $OV$, i.e., 5, 2, 1, 1, 1, 1, 1, and 1 (as shown in the 2nd column of Table 6), $SIM_d(O, D) = 0.931$ $(= 17.61 / (5.9161 \times 3.1979))$ (see the 7th column of

1: Home-Based Typist & More 4 Hire!
2: …
3: Ms. Lisa M. Staggs
4: **Phone**: **(909) 931-4841** or **(909) 603-8212**
5: e-Mail: **StaggsTranscribing@Yahoo.com**
6: …
7: Experienced in the business and clerical fields. Qualifications include effective use of office
8: equipment, accessibility to **Windows 98**, **Microsoft Word**, Word Perfect, **Excel**, the **Internet**, faxing
9: capability and a modem all from my residence. High quality communication and organizational skills,
10: proficient typist, dependable, motivated, highly productive combined with a positive and professional
11: personality. Please contact me!
12: …
13: Number Cruncher
14: …
15: Mrs. Chelle A. Perry
16: **Phone**: **(817)284-0445**
17: e-Mail: **chelleperry@yahoo.com**
18: …
19: I am proficient in **Access**, **Excel**, **Word**, and Power Point. In my 5 years of accounting experience I
20: have worked with Quickbooks and Peachtree.My experience has been primarily with A/P, A/R, G/L
21: and bank reconcilliation.
22: …
23: Jack of all Trades
24: …
25: Mr. Frank D. Swarbrick
26: **Phone**: **(203)375-3166**
27: e-Mail: **frankdavid@cshore.com**
28: …
29: I Have a certificate in computer repair.I also know how to type make phone calls.I am looking for
30: some type of telecommuting job. Please email me for more info
31: …

Figure 5: Three of the 100 resumes, advertised on the Richard W. Cummings' Web site (URL: www.2020tech.com/ww4f/cando-jacksjills.html).

Table 6), $D$ is further considered for its similarity to $O$. Since $SIM_m(O, D) = |3.1979 \; / \; 5.9161| = 0.54$ (as shown in the last column of Table 6) $< \hbar_m$ ($= 0.55$), the Web document $D$ of resumes is not classified as an instance of the job-ads application ontology.

# 6 Experimental results

We have categorized four sets of Web documents, $S_1$, $S_2$, $S_3$, and $S_4$, with valid URLs, a total of 541 Web documents based on four application ontologies, i.e., car-sale, job-ads, obituary, and university-course descriptions, and two IR models, eVSM and CM. The graphical representation of the application ontology on *job-ads* has been shown in Figure 1, whereas the others (i.e., car-sale, university-course descriptions, and obituary) are shown graphically in Figures 6(a), 6(c), and 6(b), respectively.

Out of the 541 Web documents, 293 were verified to be multiple-record Web documents

| Names of the coefficients of $DV$ | Constants & keywords in resume $a$ | Corresponding line number in Figure 5 | Frequencies of appearance of constants & keywords |
|---|---|---|---|
| Skill | Windows 98, Microsoft Word, Excel, Internet | 8 | 4 |
| Degree | | | 0 |
| Email | StaggsTranscribing@ Yahoo.com | 5 | 1 |
| Voice | (909)931-4841 (909)603-8212 | 4 | 2 |
| Web | | | 0 |
| Fax | | | 0 |
| KEYWORD(Fax) | | | 0 |
| KEYWORD(Voice) | Phone | 4 | 1 |

Table 2: Frequencies of all the constants and keywords found in resume $a$ (Home-Based Typist & More 4 Hire!) as shown between lines 1 and 11 in Figure 5.

| Names of the coefficients of $DV$ | Constants & keywords in resume $b$ | Corresponding line number in Figure 5 | Frequencies of appearance of constants & keywords |
|---|---|---|---|
| Skill | Acess, Excel, Word | 19 | 3 |
| Degree | | | 0 |
| Email | Chelleperry@yahoo.com | 17 | 1 |
| Voice | (817)284-0445 | 16 | 1 |
| Web | | | 0 |
| Fax | | | 0 |
| KEYWORD(Fax) | | | 0 |
| KEYWORD(Voice) | Phone | 16 | 1 |

Table 3: Frequencies of all the constants and keywords found in resume $b$ (Number Cruncher) as shown between lines 13 and 21 in Figure 5.

| Names of the coefficients of $DV$ | Constants & keywords in resume $c$ | Corresponding line number in Figure 5 | Frequencies of appearance of constants & keywords |
|---|---|---|---|
| Skill | | | 0 |
| Degree | | | 0 |
| Email | Frankdavid@cshore.com | 27 | 1 |
| Voice | (203)375-3166 | 26 | 1 |
| Web | | | 0 |
| Fax | | | 0 |
| KEYWORD(Fax) | | | 0 |
| KEYWORD(Voice) | Phone | 26 | 1 |

Table 4: Frequencies of all the constants and keywords found in resume $c$ (Jack of all Trades) as shown between lines 23 and 30 in Figure 5.

| Names of the coefficients of $DV$ | Frequencies of appearance of constants & keywords in all the resumes | Average frequencies of appearance of constants & & keywords in all the resumes |
|---|---|---|
| Skill | 239 | 2.39 |
| Degree | 69 | 0.69 |
| Email | 103 | 1.03 |
| Voice | 121 | 1.21 |
| Web | 17 | 0.17 |
| Fax | 30 | 0.30 |
| KEYWORD(Fax) | 50 | 0.50 |
| KEYWORD(Voice) | 107 | 1.07 |

Table 5: The average frequencies of all the constants and keywords found in the 100 resumes as partially shown in Figure 5.

| Names of the coefficients of $OV$ | Coefficients of $OV$ | Coefficients of $DV$† | Inner Product of $DV$ & $CV$ | $\|OV\|$ | $\|DV\|$ | Computed $SIM_d(O,D)$ | Computed $SIM_m(O,D)$ |
|---|---|---|---|---|---|---|---|
| Skill | 5 | 2.39 | | | | | |
| Degree | 2 | 0.69 | | | | | |
| Email | 1 | 1.03 | | | | | |
| Voice | 1 | 1.21 | | | | | |
| Web | 1 | 0.17 | | | | | |
| Fax | 1 | 0.30 | | | | | |
| KEYWORD (Fax) | 1 | 0.50 | | | | | |
| KEYWORD (Voice) | 1 | 1.07 | | | | | |
| Result | | | 17.61 | 5.9161 | 3.1979 | 0.931 | 0.54 |

† This column is the 3rd column of Table 5.

Table 6: Computed $SIM_d(O, D)$ and $SIM_m(O, D)$ for the 100 resumes as partially shown in Figure 5.

| URLs ranked in descending order | Total no. of records | $SIM_d(O,D)$ | $SIM_m(O,D)$ | $\geq \hbar_d$ 0.83 | $\geq \hbar_m$ 0.55 | Status |
|---|---|---|---|---|---|---|
| www.carsnet.com/caddy.html | 19 | 0.9547 | 0.882 | Yes | Yes | A |
| www.laneautomall.com/importauto.htm | 68 | 0.9073 | 1.010 | Yes | Yes | A |
| www.resumefactory.com/jobs/ad_car.htm | 8 | 0.9035 | 1.581 | Yes | Yes | A |
| www.brooks-inc.com/ccc/submit/west.htm | 16 | 0.8950 | 1.337 | Yes | Yes | A |
| 206.163.142.104/Private/queries/cpcar_price_1.html | 29 | 0.8777 | 0.904 | Yes | Yes | A |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| www.local-cars.com/dealers/mo19.asp | 5 | 0.8641 | 1.289 | Yes | Yes | A |
| www.dubaicompanies.com/cars/landdisc.html | 8 | 0.7674 | 0.907 | No | - | R |
| www.johnfletcher.com/1z08.htm | 8 | 0.7354 | 1.481 | No | - | R |

Note: "-" = not applicable; "R" = Rejected; "A" = Assigned to the category.

Table 7: A portion of Test set 1 - the car-sale category.

and were included in the four test sets of Web documents $F_1$, $F_2$, $F_3$, and $F_4$ ($F_i \subset S_i, 1 \leq i \leq 4$). Since each test set $F_i$ ($1 \leq i \leq 4$) contains a long list of URLs, we only display several URLs of each test set and rank (according to the similarity metrics $SIM_d(O,D)$ and $SIM_m(O,D)$ of eVSM) in descending order, as shown in the first column of Tables 7 - 10. In these tables, multiple-record Web documents are either rejected or categorized into the car-sale, job-ads, obituary, or university-course descriptions application ontology. Each rejected (assigned, respectively) document is indicated by *R(ejected)* (*A(ssigned)*, respectively) in the last column ("Status") of its corresponding row. The "Total number of records" column in each table shows the number of multiple records in each document (with a valid URL). According to our experimental results (see Table 11), we have achieved 100% precision in retrieving three sets of multiple-record Web documents relevant to the car-sale, obituary, and university-course descriptions ontologies, respectively. The recall ratios of these application ontologies are 85.7% for car sale, 100% for obituary, and 90.3% for the university-course descriptions, respectively. We also obtained 87.2% precision and 84.5% recall in retrieving the set of Web documents that are relevant to job-ads. Among the 243 relevant Web documents, we have achieved an average of 90% recall and 97% precision on the four application ontologies using our binary-categorization approach.

# 7 Concluding remarks

We present an automatic, binary-categorization approach on multiple-record Web documents based on application ontologies and two IR models, the Vector Space Model (VSM) and Clustering Model (CM). In our approach, we (1) collect a set of Web documents $S$

| URLs ranked in descending order | Total no. of records | $SIM_d(O,D)$ | $SIM_m(O,D)$ | $\geq \hbar_d$ 0.83 | $\geq \hbar_m$ 0.55 | Status |
|---|---|---|---|---|---|---|
| www.ext.nodak.edu/ homepages/aedept/jobs.htm | 13 | 0.9721 | 0.615 | Yes | Yes | A |
| www.klaa.com/careers. htm | 4 | 0.9619 | 0.581 | Yes | Yes | A |
| www.tax-jobs.com/ listse.htm | 27 | 0.9616 | 1.132 | Yes | Yes | A |
| www.snr.missouri.edu/ ~fass/fassjobs.htm | 6 | 0.9615 | 1.445 | Yes | Yes | A |
| florida-jobs-online. com/p178.htm | 39 | 0.9568 | 0.898 | Yes | Yes | A |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| www.2020tech.com/ ww4f/cando-jacksjills.html | 100 | 0.9501 | 0.54 | Yes | No | R |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| www.tekjobs.com/ cgibin/hotjobs.cgi | 20 | 0.8552 | 2.9166 | Yes | Yes | A |
| www.peoplesoft.com/ en/career_opportunities/ technical_en.htm | 20 | 0.8496 | 2.9159 | Yes | Yes | A |
| christianet.com/christ ianjobs/miscellaneous.htm | 27 | 0.7759 | 0.4162 | No | - | R |
| www.belhaven.edu/ CampusLife/Life/jobs.htm | 58 | 0.7404 | 0.2275 | No | - | R |
| www.employersusa. com/cgi-bin/folioisa.dll/disx_ topjobs/query=*/doc/{t46221} | 10 | 0.7287 | 0.6166 | No | - | R |

Note: “-” = not applicable; “R” = Rejected; “A” = Assigned to the category.

Table 8: A portion of Test set 2 - the job-ads category.

| URLs ranked in descending order | Total no. of records | $SIM_d(O,D)$ | $SIM_m(O,D)$ | $\geq \hbar_d$ 0.83 | $\geq \hbar_m$ 0.55 | Status |
|---|---|---|---|---|---|---|
| www.dnps.com/dies/ 0418dies.htm | 82 | 0.9817 | 0.7050 | Yes | Yes | A |
| www.dnps.com/dies/ 0330dies.htm | 79 | 0.9786 | 0.7336 | Yes | Yes | A |
| www.geocities.com/ Heartland/Bluffs/ 7748/obits3.htm | 66 | 0.9573 | 0.7900 | Yes | Yes | A |
| www.abqjournal.com/ obits/obits1998/ OB04-25.HTM | 26 | 0.9410 | 0.8900 | Yes | Yes | A |
| www.wico.net/˜zacho/ waveobitH.htm | 44 | 0.9306 | 1.160 | Yes | Yes | A |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| www.geocities.com/ Heartland/Estates/1160/ obituaries.htm | 22 | 0.9161 | 0.4304 | Yes | No | R |
| www.palatkadailynews. com/pages/standing/obits.html | 42 | 0.9056 | 0.862 | Yes | Yes | A |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| www.geopages.com/Heart land/Plains/5012/funeral22.html | 10 | 0.8129 | 0.3691 | No | - | R |
| www.geopages.com/Heart land/Plains/5012/funeral12.html | 25 | 0.7717 | 0.3237 | No | - | R |
| www.geocities.com/Heart land/Plains/5012/funeral2.html | 23 | 0.7594 | 0.3465 | No | - | R |
| www.geopages.com/Heart land/Plains/5012/funeral7.html | 9 | 0.7566 | 0.6351 | No | - | R |
| www.geopages.com/Heart land/Plains/5012/funeral19.html | 4 | 0.7313 | 0.4128 | No | - | R |

Note: "-" = not applicable; "R" = Rejected; "A" = Assigned to the category.

Table 9: A portion of Test set 3 - the obituary category.

(e.g., retrieved by a search engine) and identify the set of multiple-record Web documents $M$ in $S$ using our combined heuristic approach [EJN99], (2) create the ontology vector for an application ontology $O$, (3) create the document vector for each multiple-record Web document $D$ in $M$ according to CM, and (4) determine the relevance of $D$ to $O$ using their corresponding vectors based on an extended VSM, i.e., eVSM.

We have verified the correctness of our approach on four application ontologies: car-ads, job-ads, obituaries, and university-course descriptions. Using the search-engine queries on the four application ontologies, we collected 622 Web documents that were retrieved using Yahoo! and AltaVista, out of which 541 Web documents came with valid URLs. These documents are representative Web documents retrieved from different Web sites at different geographical locations all over the world. Among the 541 documents, 293 Web documents were identified as multiple-record Web documents. Out of the 293 documents, 231 multiple-record Web documents were correctly categorized to the four application ontologies using our categorization approach. Our experimental results show that we have achieved an

| URLs ranked in descending order | Total no. of records | $SIM_d(O,D)$ | $SIM_m(O,D)$ | $\geq \hbar_d$ 0.83 | $\geq \hbar_m$ 0.55 | Status |
|---|---|---|---|---|---|---|
| www.runet.edu/˜psyc-web /undergrad/undercourse.html | 30 | 0.9996 | 1.025 | Yes | Yes | A |
| www.registrar.ualberta. ca/calendar/courses/NURS.html | 93 | 0.9991 | 1.520 | Yes | Yes | A |
| www.plattsburgh.edu/ catalog/phycd.html | 32 | 0.9980 | 1.119 | Yes | Yes | A |
| www.wku.edu/Dept/ Academic/Ogden/CS/course_ descriptions.html | 47 | 0.9978 | 1.231 | Yes | Yes | A |
| plaid.hawk.plattsburgh. edu/catalog/psccd.html | 32 | 0.9977 | 1.057 | Yes | Yes | A |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| canes.gsw.peachnet.edu/ ˜soe/courses/middle_grades.htm | 47 | 0.8859 | 0.3422 | Yes | No | R |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| www.ais-salzburg.com/ data/academic-overview/ course-listings/high.html | 48 | 0.7494 | 1.3204 | No | - | R |
| osu-okmulgee.edu/ catalog/dhe.htm | 29 | 0.7485 | 1.0019 | No | - | R |
| www.franklin.edu/acad prog/coursedesc/hsa-desc.html | 12 | 0.7391 | 1.2402 | No | - | R |
| mgmt.utoronto.ca/mba/ Curriculum/ | 70 | 0.7071 | 0.1111 | No | - | R |
| www.luc.edu/schools/bus iness/academics/descript.htm | 89 | 0.7040 | 0.6583 | No | - | R |

Note: "-" = not applicable; "R" = Rejected; "A" = Assigned to the category.

Table 10: A portion of Test set 4 - the university-course descriptions categry.

average of 90% recall and 97% precision in categorizing the search-engine retrieved multiple-record Web documents. Based on our experimental results, our categorization approach looks very promising. The recall and precision of our approach are significantly high.

Even though we have achieved high recall and precision in our categorization approach, there remains other work to be pursued. One of the limitations of our categorization approach is that we only deal with textual Web documents. Furthermore, we have not considered multiple-record Web documents that can only be retrieved by filling in CGI (Common Gateway Interface) forms, nor have we considered any Web documents that are connected by hyperlinks from a source homepage.

# References

[BB63] H. Borko and M. Bernick. Automatic document classification. *Journal of the ACM*, 10(2):151–162, 1963.

| Application Ontology | $D_{SE}$ | $D_{URL}$ | $D_{MR}$ | $RD_{CAT}$ | $Rel\_D_{MR}$ | $Rel\_RD_{CAT}$ | Recall % ($Rel\_RD_{CAT}$ / $Rel\_D_{MR}$) | Precision % ($Rel\_RD_{CAT}$ / $RD_{CAT}$) |
|---|---|---|---|---|---|---|---|---|
| Car Sale | 220 | 159 | 8 | 6 | 7 | 6 | 85.7 | 100 |
| Job Ads | 160 | 144 | 112 | 94 | 97 | 82 | 84.5 | 87.2 |
| Obituary | 97 | 93 | 43 | 31 | 31 | 31 | 100 | 100 |
| University-Course Descriptions | 145 | 145 | 130 | 112 | 124 | 112 | 90.3 | 100 |
| Total | 622 | 541 | 293 | 243 | 259 | 231 | 90 (average) | 97 (average) |

$D_{SE}$ = Number of Web documents submitted (SE)
$D_{URL}$ = Number of Web documents, out of the $D_{SE}$ documents, that have valid URLs
$D_{MR}$ = Number of Web documents, out of the $D_{URL}$ documents, that are multiple-record documents
$RD_{CAT}$ = Total number of multiple-record Web documents, out of the $D_{MR}$ documents, identified by our categorization approach
$Rel\_D_{MR}$ = Actual number of relevant multiple-record Web documents among all the $D_{MR}$ documents
$Rel\_RD_{CAT}$ = Number of relevant multiple-record Web documents retrieved by using our categorization approach

Table 11: Recall and precision ratios on categorizating multiple-record Web documents relevant to the car-sale, job-ads, obituary, and university-course descriptions application ontologies, respectively.


[BGG+99] D. Boley, M. Gini, R. Gross, E.-H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. More. Document categorization and query generation on the world wide web using webace. *Journal of Artificial Intelligence Review*, 13(5-6):365–391, 1999.

[Bun79] M.A. Bunge. *Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems.* Reidel, Boston, 1979.

[BYRN99] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, Menlo Park, California, 1999.

[CGRU97] C. Chekuri, M. Goldwasser, P. Raghavan, and E. Upfal. Web Search Using Automatic Classification. In *Proceedings of the Sixth International WWW Conference*, 1997.

[CLvR98] F. Crestani, M. Lalmas, and C.J. van Rijsbergen. *Information Retrieval: Uncertainty and Logics.* Kluwer Academic Publishers, Massachusetts, 1998.

[ECJ+99] D. W. Embley, D. M. Campbell, Y.S. Jiang, S. W. Liddle, D. W. Lonsdale, Y.-K. Ng, and R. D. Smith. Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages. *Journal of Data and Knowledge Engineering*, 31(3):227–251, November 1999.

[EJN99] D.W. Embley, Y. Jiang, and Y.-K. Ng. Record-Boundary Discovery in Web Documents. In *Proceedings of the SIGMOD'99 Conference*, pages 467–478, 1999.

[ENX01] D.W. Embley, Y.-K. Ng, and L. Xu. Recognizing Ontology-Applicable Multiple-Record Web Documents. In *Proceedings of the 20th International Conference on Conceptual Modeling (ER 2001)*, pages 555–570, November 2001.

[KS97] D. Koller and M. Sahami. Hierarchically Classifying Documents Using Very Few Words. In *Proceedings of the 14th International Conference on Machine Learning*, pages 170–178, July 1997.

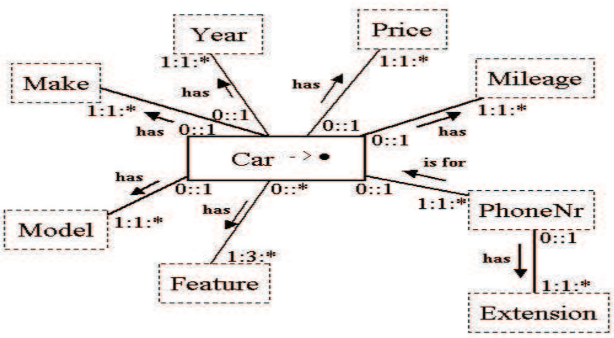
[LSC96] D. Lewis, R. Schapire, and J. Callan. Training Algorithms for Linear Text Classifiers. In *Proceedings of the ACM SIGIR*, pages 298–306, 1996.

[RL94] E. Riloff and W. Lehnert. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12(3):296–333, July 1994.

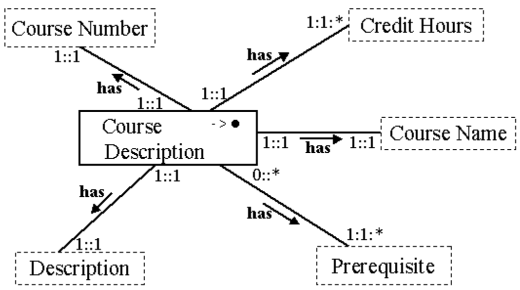
[Sal89] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, New York, 1989.

[SDUS98] V. Storey, D. Dey, H. Ullrich, and S. Sundaresan. An ontology-based expert system for database design. *Data & Knowledge Engineering*, 28(1):31–46, 1998.

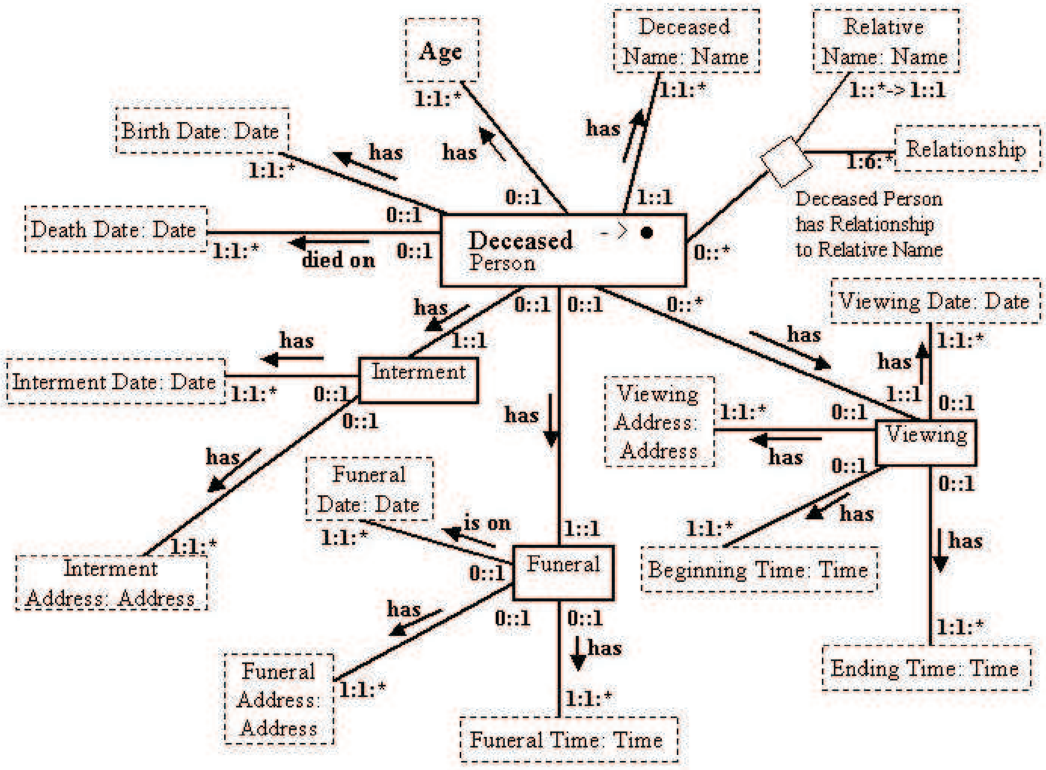
[SM83] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.




(a) Graphical **car-sale** application ontology.




(b) Graphical **university-course descriptions** application ontology.




(c) Graphical **obituary** application ontology.

Figure 6: Graphical representations of the car-sale, university-course descriptions, and obiturary application ontologies.