# Filtering Multiple-Record Web Documents Based on Application Ontologies*

D.W. Embley (Contact Author)
Y.-K. Ng
L. Xu
Department of Computer Science
Brigham Young University
Provo, Utah 84602, U.S.A.
{embley,ng,lx}@cs.byu.edu

**Abstract**

Hand filtering of a large set of documents, such as a list of search-engine-returned Web pages, is extremely tedious. As a step toward solving this problem, we propose an automated technique for filtering multiple-record Web documents. Our approach is database-centric, being based on conceptual-model schemes called application ontologies, and it uses information-retrieval techniques and machine-learned decision rules. Given potential attribute-value pairs recognized in an unstructured Web document, we apply three heuristics: (1) a density heuristic that measures the percent of the document that appears to apply to an application ontology, (2) an expected-value heuristic that compares the number of values found for each attribute in a document to the number expected by the application ontology, and (3) a grouping heuristic that considers whether the potential attribute-value pairs of the document appear to be grouped as application-ontology records. Then, based on an application-dependent, machine-learned rule over the heuristic measurements, we filter Web documents. Our experimental results show that we have been able to achieve about 95% accuracy, with over 90% for both recall and precision.

**Keywords**: Document filtering, domain-specific filtering, filtering heuristics, machine-learned filtering rules, application ontologies, information extraction, World-Wide Web.

## 1 Introduction

The World Wide Web contains abundant repositories of information in Web documents. Given a set of these documents (e.g. a set returned by a search-engine query), how can we automatically select just those documents that have the information a user wants? When we select from a set those documents relevant to a user's information needs and discard irrelevant documents, we appropriately *filter* the set for the user. If a filtering process discards too many relevant documents (i.e. has too many *false drops*), it has poor *recall*—ratio of the number of relevant documents accepted to the total number of relevant documents in the set. If a filtering process accepts too many irrelevant documents (i.e., has too many *false positives*), it has poor *precision*—ratio of the

*Reference Number: 334, Category: Research number of relevant documents accepted to the sum of the number of documents accepted and the number of irrelevant documents accepted. As a combined measure, which also takes into account the total number of documents, *accuracy* measures how well a filtering process classifies a set of documents into sets of relevant and irrelevant documents by giving the ratio of total number of documents minus the false drops and false positives to the total number of documents. We wish to have a filtering process that is highly accurate and has both high recall and high precision.

In this paper we propose a new approach to filtering (large) sets of Web documents that are possibly applicable to a chosen application of interest. We base our approach on application ontologies [EJN99], which are conceptual-model snippets [Wan89, SDUS98] of standard ontologies [Bun77, Bun79], and we use techniques from information retrieval [SM83, BYRN99] and machine learning [Qui93] to filter a set of potentially suitable Web documents to a subset suitable for the application. Our filtering objective is related to text categorization (e.g. [LH98, WZL99, YL99])—each application ontology can be a category—but our approach is vastly different because we have rich predefined category descriptions (application ontologies), and we use heuristics specifically applicable to multiple-record Web documents.

Our experience in teaching others to use our system suggests that an application ontology of the kind we use can be created in two to three dozen person-hours. Since our approach does have this associated cost, it is not appropriate for ad-hoc, one-time Web searches. It is appropriate, however, for inquiries that are either long-lasting or are common-interest applications for which the cost of application-ontology development can be amortized over large user groups. We believe that many serious inquiries fall into these two categories.

Our work reported here is also partly motivated by our success in using application ontologies to extract information from multiple-record Web documents that are data-rich and narrow in ontological breadth [EJN99]. For several applications we have tried—automobile want-ads, obituaries, jobs, real estate, stocks, musical instruments, precious gems, games, personals, and computer monitors—we have achieved fact-extraction recall rates mostly around 90% and fact-extraction precision rates mostly better than 90% [EJN99]. If we can locate documents applicable to an ontology and filter them successfully (as proposed in this paper), we can apply techniques we have already developed, to extract, structure, and archive in databases, information found in data-rich, application-specific Web documents. Thus, the contributions of this work have the potential to be more far-reaching than just the salient contribution of increasing accuracy, recall, and precision in filtering application-specific Web documents.

We outline our approach and provide an overview for the rest of our paper as follows. In our approach we first specify a conceptual-model ontology for our application of interest. Section 2 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. Basically, an application ontology is much like a database scheme; it specifies (1) objects of interest and their relationship to one another, (2) expected cardinalities among objects in relationships, and (3) lexical appearance of objects of interest, including context and keyword information to help match identified lexical objects in a document with object sets specified in the application ontology.

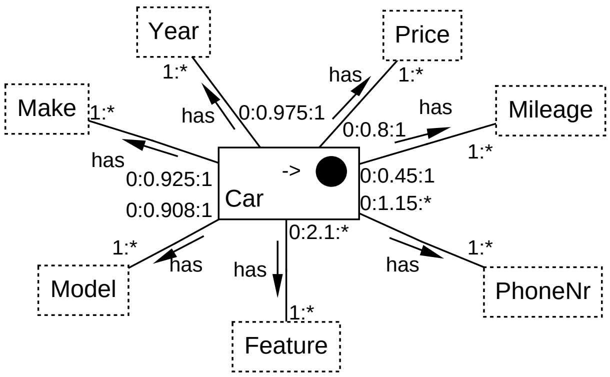
Then, given a set of Web documents we wish to filter, we obtain statistics for three heuristics for each document: (1) a density heuristic, (2) an expected-values heuristic, and (3) a grouping heuristic. Our density heuristic is concerned with the ratio of the number of plain-text characters recognized as being of possible interest to the total number of plain-text characters in a document. Section 3.1 describes these details. Our expected-values heuristic is concerned with the comparison between the number of values actually recognized in a document for each attribute and the number of values expected for each attribute in an application ontology. We use the vector-space model [SM83, BYRN99] for measuring how well the expectations are met. Section 3.2 explains these details. Our grouping heuristic is concerned with whether the values recognized are in record groups. We measure proper grouping by looking for unexpected repetitions of value types. As Section 3.3 explains, not many unexpected repetitions should be found if records are properly grouped. We combine the results returned from these three heuristics in machine-learned decision-tree rules. Section 3.4 explains our procedure for obtaining a filtering rule.

In Section 4 we discuss our experimental results—which, for the application ontologies we tried, are over 95% accurate and are in the 90% range for both recall and precision. In Section 5, we give concluding remarks.

## 2 Application Ontology

An application ontology is a conceptual-model instance that describes a real-world application of interest [EJN99]. An application ontology consists of two components: (1) an *object/relationship-model instance*, which describes sets of objects, sets of relationships among objects, and constraints over object and relationship sets, and (2) for each object set, a *data frame* that defines the contents of the object set. A data frame for an object set defines the lexical appearance of constant objects within the object set and establishes appropriate keywords that are likely to appear in a document when objects in the object set are mentioned [Emb80].

Figure 1(a) shows a graphical representation of the object sets and relationship sets of our car-ads application ontology. (Appendix A contains a graphical representation of our obituaries application ontology—a much more complex application, as can easily be seen by the increased complexity of the diagram.) Figure 1(b) shows part of the textual representation for our car-ads ontology, including all object and relationship sets and all cardinality constraints (lines 1-8) and a few lines of the data frames (lines 9-42). (The full textual ontology is about 600 lines in length.) The system we have built accepts application ontologies as depicted in Figure 1(b), but




(a) Cars application ontology—graphical.

```
Car [-> object];
Car [0:0.975:1] has Year [1:*];
Car [0:0.925:1] has Make [1:*];
Car [0:0.908:1] has Model [1:*];
Car [0:0.45:1] has Mileage [1:*];
Car [0:2.1:*] has Feature [1:*];
Car [0:0.8:1] has Price [1:*];
PhoneNr [1:*] is for Car [1:1.15:*];
Year matches [4]
  constant { extract "\d{2}";
          context "([^\$\d]|^)[4-9]\d[^,\dkK]";
          substitute "^" -> "19";
           },
         { extract "\d{2}";
          context "([^\$\d]|^)[4-9]\d,[^\d]";
          substitute "^" -> "19";
          },
         { extract "\d{2}";
          context "\b'[4-9]\d\b";
          substitute "^" -> "19";
          }, 22.          { extract "\d{4}";
          context "([^\$\d]|^)(19[4-9]\d|200\d)[^,\dkK]";
           },
         { extract "\d{4}";
          context "([^\$\d]|^)(19[4-9]\d|200\d),[^\d]";
          };
end;
Make matches [10]
  constant { extract "\bacura\b"; },
         { extract "\balfa(\s*romeo)?\b"; },
         { extract "\bamc\b"; },
         { extract "\bam\s*general\b"; },
         { extract "\baudi\b"; },
         ...
end;
...
Mileage matches [8]
...
  keyword  "\bmiles\b", "\bmi\.", "\bmi\b", "\bmileage\b";
end;
...}
```

(b) Cars application ontology—textual.

Figure 1: Car-ads applicaiton ontology.

we often use the graphical abstraction of an ontology as depicted in Figure 1(a) in presentations and discussions.

An object set in an application ontology represents a set of objects which may either be lexical or nonlexical. Dashed boxes (in the graphical representation in Figure 1(a)) or data frames with constant declarations (in the textual representation in Figure 1(b)) represent lexical object sets, and solid boxes (in the graphical representation) or data frames without constant declarations (in the textual representation) represent nonlexical object sets. We describe the constant lexical objects and the keywords for an object set by regular expressions using Perl syntax. When applied to a textual document, the **extract** clause in a data frame causes a string matching a regular expression to be extracted, but only if the **context** clause also matches the string and its surrounding characters. A **substitute** clause lets us alter the extracted string before we store it in an intermediate file, in which we also store the string's position in the document and its associated object set name (i.e. its data frame name). The **extract/context/substitute** triple in Lines 18–20 of Figure 1(b), for example, matches dates such as *'92*, *'75*, and *'69* and (respectively) stores *1992*, *1975*, and *1969* along with the object set name *Year* and the character start and end positions of the identified years in the text.

One of the nonlexical object sets is the *object set of interest*—*Car* for the car-ads ontology and *Deceased Person* for the obituary ontology. The graphical notation "-> •" and the textual notation "[-> object]" denote the object set of interest. Only one object set of interest may appear in an application ontology. In general, the symbol "->" specializes the application ontology for a single record. A "•" designates a single, stand-alone object in the graphical notation.

A line connecting rectangles in Figure 1(a) represents a relationship set among the connected object sets. A labeled reading-direction arrow next to a relationship-set line along with the names of the connected object sets provides a name for a relationship set. *Car has Year*, for example, names the relationship set between *Car* and *Year* in Figure 1(a). In the textual representation we use this name to represent both the relationship set and the connected object sets. Line 2 in Figure 1(b), for example, defines the *Car has Year* relationship set.

The *min:max* pairs and *min:ave:max* triples near the connections between object sets and relationship sets in Figure 1(a) are *participation constraints*: *min* designates the minimum number of times an object in the object set can participate in the relationship set; *ave* 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, with * designating an unknown but finite maximum number of times. The participation constraint on *Car* for *Car has Feature*, for example, specifies that a car need not have any listed features, that a car has 2.10 features on the average, and that there is no specified maximum for the number of features listed for a car.

For our car-ads and obituary application ontologies, we obtained participation constraints as follows. We selected 10 different regions covering the United States and found one car-ads page and one obituary page from each of these regions. From each of these pages we selected 12 individual car-ads/obituaries by taking every $n/12$-th car-ad/obituary, where $n$ was the total number of car-ads/obituaries on the page. We then simply counted by hand and obtained minimum, average, and maximum values for each object set in each relationship set.

## 3 Filtering Heuristics

We are interested in determining whether a given document $D$ is suitable for an application ontology $O$. In our filtering approach, we consider three different heuristics: ($H_1$) *density*, ($H_2$) *expected values*, and ($H_3$) *grouping.* $H_1$ measures the density of constants and keywords defined in $O$ that appear in $D$. $H_2$ uses the Vector Space Model (VSM) [SM83], a common information-retrieval measure of document relevance, to compare the number of constants expected for each object set, as declared in $O$, to the number of constants found in $D$ for each object set. $H_3$ measures the occurrence of groups of lexical values found in $D$ with respect to expected groupings of lexical values implicitly specified in $O$.

The next three subsections define these three heuristics, explain the details about how we provide a measure for each heuristic, and give examples to show how they work. The fourth subsection explains how we use machine learning to combine these heuristics into a single filtering rule. When reading these subsections, bear in mind that in creating these heuristics, we favored simplicity. More sophisticated measures can be obtained. For example, for $H_1$ we could account for uncertainty in constant and keyword matches [EFKR99]. For $H_2$, we could more accurately match object sets with recognized values by using our more sophisticated downstream heuristics [EJN99]. For $H_3$, we could first compute record boundaries [EJN99]. However, more sophisticated measures are more costly. We have chosen to experiment with less costly filtering heuristics, and, as will be shown, our results bear out the seeming correctness of this choice.

## 3.1 $H_1$: Density Heuristic

A Web document $D$ that is relevant to a particular application ontology $A$ should include many constants and keywords defined in the ontology. Based on this observation, we define a *density heuristics*. We compute the density of $D$ with respect to $O$ as follows:

$$Density(D, O) = \text{total number of matched characters / total number of characters}$$

where *total number of matched characters* is the number of characters of the constants and keywords recognized by $O$ in $D$, and *total number of characters* is the total number of characters in $D$.

We must be careful, of course, not to count characters more than once. For example, in the phrase "asking only 18K" a car-ads application ontology might recognize "18K" as potentially both a price and a mileage. Nevertheless, we should only count the number of characters as three, not six. Document position information for recognized strings tells us which strings overlap.

Consider the Web document $D_a$ in Figure 2(a). Recall that the lexical object sets of the car-ads application ontology are $Year$, $Make$, $Model$, $Mileage$, $Price$, $Feature$, and $PhoneNr$. Some of the lexical values found in $D_a$ include 1989 (Year), \$1900 (Price), 100K (Mileage), Auto (Feature), Cruise (Feature), (336)835-8579 (PhoneNr), Subaru (Make), and SW (Model). Only the keywords, "miles" and "mileage" appear in $D_a$. The total Number of characters in $D_a$ is 2048, whereas the number of matched characters is 626. Hence, the density of $D_a$ is $0.3056 = 626/2048$.

When we apply the density heuristic for the car-ads application ontology to the Web document $D_b$ in Figure 2(b), the density is much lower. Although no makes, models, or car features appear,

Last Updated
Monday, January 24, 2000 12:19pm Cars for Sale

---

DEPENDABLE CAR
1989 Subaru SW. Auto, AC, $1900 OBO. Call (336)835-8579.

---

FACTORY WARRANTY
1998 Elantra. Black 4 door w/ tinted windows. Auto, pb, ps, cruise, am/fm cassette stereo, a/c. Excellent condition pay off OBO. Call (336)526-5444 anytime & leave message.

---

1994 HONDA ACCORD EX
Auto, power everything, jade green w/gold package. Under 100K miles. Call (336)526-1081 after 7pm.

---

1999 GRAND AM
27,000 miles, silver, auto, still under warranty. $14,000 OBO. Call (336)366-4996 anytime.

---

`53 Chevy Bel Aire. All original, looks like new. Serious inquiries only. $8500. Call (336)468-8924 after 4 pm.

---

TWO GREAT CARS
1973 MGB convertible. British racing green. Mags, new tires, 4-speed, 1 owner, excellent running condition. $4500.
1977 Olds Cutlass Supreme. New white paint job w/ 1/2 red Landau top, original mags & new tires. Auto., 1 owner, low mileage, loaded. Call (336)984-2843.

---

95 FORD CONTOUR
5-speed, great condition, one owner, $5300. Call (336)526-8853 & leave message if no answer.

---

SEIZED CARS FROM $500
Sports, luxury & economy cars, trucks, 4x4's utility and more. For current listings, call 1-800-311-5048 ext. 10012.

---

1996 VW JETTA GL
26,000 miles. 4 door, 5-speed, AC, sunroof, 1 owner. $11,000. Call (336)874-7317 anytime.

---

`85 Buick Park Avenue. $500. Head may be cracked. Will run. Body good condition. Call (336)526-2768.

---

`95 Ford Thunderbird. Loaded, V-8, 45K, $6995. Call S&J Motors at (336)874-3403.

---

`96 Mercury Tracer. 4 door, 5 speed, 34K, $4995. Call S&J Motors at (336)874-3403.

---

`88 Firebird. V8, 5.0, fuel injected, T-tops, 109,000 miles, red, runs great. $1880. Call (336)526-1164 anytime.

---

1990 CONVERSION VAN
350 motor, auto, new tires, TV, VCR, captain chairs, front & rear AC. $4,995. Call (336)320-2658 anytime.

---

COMMERCIAL WORK VAN
`95 Chevy Astro, V6, w/ac & fully equipped utility shelves. $9400. Call (336)526-2675 & leave message.

---

(a) Car advertisements retrieved from http://www.elkintribune.com/.

Apartment For Rent  For Sale or Rent  Lost or Found
For Rent  Help Wanted
For Sale  House For Rent

---

Apartment For Rent
ONE EFFICIENCY, 2 & 3 bdrm, all utilities paid. Call 281-2051 -

---

For Rent
HOUSING SOLUTIONS - Free TV cable furn. $60/wk - $210/mo. 281-4060. -

---

For Sale
1998 JD 455 mower, 60' deck. Call for price. Also, homemade Go-Cart. Call after 5:30 pm 218-281-1128. -

---

For Sale or Rent
10,000 SQ. FT. office building. Handicap accessible. Call 281-3631. -

---

Help Wanted
NOW HIRING full time and part time customer service representatives. Advancement possible and weekly pay. Must be able to work weekends and holidays. Apply at Superamerica, 411 N Main St., Crookston, MN EOE -

---

PART-TIME AND weekend help working with developmentally disabled adults. Call Melissa or Karen at 281-3872. -

---

REM-NORTHWEST Services, Inc. has a full time Program Coordinator/Coordinator position open in Crookston working with four developmentally disabled adults. Duties include hiring, staff supervision, scheduling, oversight of most areas of the home's operation. Applicant must be 18 years of age or older. Must have a high school diploma or equivalent. One year experience serving people with developmental disabilities preferred. Must have a valid driver's license and driving record that meets REM's insurability requirements. Insurance and benefits available. If interested call for application at 218-281-5113. E.O.E. -

---

REM-NORTHWEST Services, Inc. has full and part time Coordinator positions available in Crookston, MN, working with citizens who are developmentally disabled. Excellent benefits are offered including health, dental, life, 401K and profit sharing for full and part time employees working 20 hours a week or more. Exceptional training is provided Applicants must be 18, have a valid driver's license and high school diploma or GED. Apply by calling for application at 218-281-5113 or 1-800-532-7655. E.O.E. -

---

House For Rent
3 BDRM HOUSE $450/mo. 281-1970. 22 STEEL BUILDINGS, NEW, must sell. 40x60x14 was $17,500 now $10,971; 50x100x16 was $27,850 now $19,990; 80x135x16 was $79,850 now $42,990; 100x175x20 was $129,650 now $78,850. 1-800-406-5126. -

---

Lost or Found
FOUND: Golden retriever about 4 months old. Found 7 miles south of Crookston. Call after 5:30 pm 281-1128. -

---

(b) Items for sale advertisements retrieved from http://www.crookstontimes.com.

Figure 2: A car advertisement Web document and a non-car advertisement Web document.

there are years, prices, and phone numbers and the ontology (mistakenly) recognizes "10,000" (in "10,000 SQ. FT.") and "401K" (the retirement plan) as potential mileages. Altogether 196 characters of 2671 are recognized by the car-ads ontology. Thus, the density is 0.0734.

### 3.2 $H_2$: Expected-Values Heuristic

We apply the VSM model to measure the relevance of a multiple-record Web document $D$ to an application ontology $O$. Based on the lexical object sets and the participation constraints in $O$, we construct an ontology vector $OV$. Based on the same lexical object sets and the constants recognized for these object sets by $O$ in $D$, we construct a document vector $DV$. We measure the relevance of $D$ and $O$ by comparing $DV$ and $OV$.

#### 3.2.1 Ontology Vector (OV) Construction

To construct the ontology vector $OV$, we (1) identify the lexical object-set names—these become the names of the coefficients of $OV$, and (2) determine the average participation for each lexical object set with respect to the object set of interest specified in $O$—these become the values of the coefficients of $OV$.

For example, consider the car-ads application ontology. The lexical object sets and the average values of these object sets in the ontology are given in Figure 1. Thus, the ontology vector for the car-ads application ontology is $<$ Year:0.975, Make:0.925, Model:0.908, Mileage:0.45, Price:0.8, Feature:2.1, PhoneNr:1.15 $>$.

#### 3.2.2 Document Vector ($DV$) Construction

The names of the coefficients of $DV$ are the same as the names of the coefficients of $OV$. We obtain the value of each coefficient of $DV$ by automatically counting the number of appearances of constant values in $D$ that belong to each lexical object set.

Consider the multiple-record Web document $D_a$ in Figure 2(a). The names of the coefficients of $DV$ for the car-ads application ontology are the lexical object sets $Year$, $Make$, $Model$, $Mileage$, $Price$, $Feature$, and $PhoneNr$, which are the same as in $OV$. As Table 1 shows, the values of the coefficients of $DV$ for $D_a$ are the number of lexical values for each lexical object set found in $D_a$.

Further consider multiple-record Web document $D_b$, which does not contain car ads, in Figure 2(b). We include this Web document to demonstrate the difference in the measures for a car-ads document and a non-car-ads document. The coefficients of $DV$ for $D_b$ are in Table 2.

We have discussed the creation of the document vector as if correctly detecting the appearance of lexical values in a document is easy—but it is not easy. Identifying *potential* lexical values for an object set is easy—when a regular expression in a data frame matches as explained in Section 2, we associate the recognized string with the data frame's object set. What's hard is sorting out which of these potential object-set/constant pairs to accept. In our downstream processes, we use sophisticated heuristic based on keyword proximity, application-ontology cardinalities, record boundaries, and missing-value defaults to best match object sets with potential constants. For filtering we use techniques that are far less sophisticated and thus also far less costly.

In our simple extraction procedure for filtering, there are two cases: (1) a recognized string has no overlap either partially or completely with any other recognized string, and (2) a recognized string does overlap in some way with at least one other recognized string. For Case 1, we accept it even if the sophisticated downstream processes would reject it. As an example, there might be two years mentioned in a car ad—one for the year of the car in a summary title and one for the year of the car in the detailed car ad. We would recognize and count both in our filtering process even though a downstream cardinality-constraint check would force us to have at most one year for a car.

For Case 2, we resolve the overlap. There are three subcases: (1) exact match, (2) subsumption, and (3) partial overlap.

1. *Exact Match.* If a lexical value $v$ is recognized as potentially belonging to more than one lexical object set, we use keywords that appear before or after $v$ to determine which object set to choose. For example, according to our obituary ontology the lexical value "March 20, 1999" is potentially a *Birth Date*, *Death Date*, *Viewing Date*, *Interment Date*, or *Funeral Date.* If the closest keyword phrase to "March 20, 1999" is one of the death-date phrases *died, passed on, was released from*, etc., we make "March 20, 1999" a *Death Date.* If no applicable keywords are found either before or after, we choose one of the object sets arbitrarily.

2. *Subsumption.* If a lexical value $v$ is a proper substring of of lexical value $w$, and $v$ belongs to the lexical object set $v\_set$ whereas $w$ belongs to the lexical object set $w\_set$, then we give preference to $w$ and classify $w$ as an element of $w\_set$.

3. *Partial Overlap.* If lexical value $v$ of the lexical object set $v\_set$ and lexical value $w$ of the lexical object set $w\_set$ appear in a Web document, such that a suffix of $v$ is a prefix of $w$, we retain $v$ as a lexical value of $v\_set$ and discard $w$, i.e., we give preference to $v$ since it appears before $w$ in the document.

#### 3.2.3 Similarity Cosine

We measure the similarity between an ontology vector $OV$ and a document vector $DV$ by measuring the cosine of the angle between them. In particular, use use the *Similarity Cosine Function* defined in [SM83], which calculates the acute angle $SIM(D,O) = \cos\theta = P/N$, where $P$ is the inner product of the two vectors[1], and $N$ is the product of the lengths of the two vectors. When

[1]The inner product of two vectors is the sum of the products of their corresponding coefficients.

| Name of lexical object set | Corresponding lexical values in the document | Frequencies of appearance of anticipated lexical values |
|---|---|---|
| Year | 1989, 1998, 1994, 1999, '53, 1973, 1977, 95, 1996, ... | 16 |
| Make | Subaru, HONDA, Chevy, Olds, FORD, VW, Buick, Murcury, ... | 10 |
| Model | SW, Elantra, ACCORD, GRAND AM, Cutlass, CONTOUR, JETTA, ... | 12 |
| Mileage | 100K, 27000, 26000, 45K, 34K, 109000 | 6 |
| Price | \$1900, \$14,000, \$8500, \$4500, \$5300, \$11,000, \$6995, \$4995, \$1880, ... | 11 |
| Feature | Auto, Black, 4 door, pb, ps, cruise, am/fm, cassette, stereo, green, ... | 29 |
| PhoneNr | (336)835-8579, (336)526-5444, (336)526-1081, (336)366-4996, ... | 15 |

Table 1: The frequencies of appearance of lexical values that are found in the multiple-reocrd car advertisements as shown in Figure 2(a).

| Name of lexical object set | Corresponding lexical values in the document | Frequencies of appearance of anticipated lexical values |
|---|---|---|
| Year | 1999, 1998, 60, 401K, 50, 80 | 6 |
| Make | | 0 |
| Model | | 0 |
| Mileage | 10,000, 401K | 2 |
| Price | \$17,500, \$10,971, \$27,850, \$19,990, \$79,850, \$42,990, \$129,650, \$78,850 | 8 |
| Feature | | 0 |
| PhoneNr | 281-2051, 281-4060, 218-281-1128, 281-3631, 281-3872, 218-281-5113, 800-532-7655, 281-1970, 800-406-5126, 281-1128 | 11 |

Table 2: The frequencies of appearance of lexical values that are found in the multiple-reocrd *Items for Sale* as shown in Figure 2(b).

the distribution of values among the object sets in $DV$ closely matches the expected distribution, the angle $\theta$ will be close to zero degrees, and cos $\theta$ will be close to one.

Consider the car-ads application ontology $O$ as shown in Figure 1 and the Web document $D_a$ as shown in Figure 2(a). The coefficients of $OV$ for $O$ are 0.975, 0.925, 0.908, 0.45, 0.8, 2.1, and 1.15, which are the average values of lexical object sets $Year$, $Make$, $Model$, $Mileage$, $Price$, $Feature$, and $PhoneNr$, respectively in the car-ads application ontology. The coefficients of $DV$ for $D_a$ are 16, 10, 12, 6, 11, 29, and 15 (see the last column of Table 1), which are the frequencies of appearance of the anticipated lexical values in $D_a$. We calculate the inner product of the coefficients of $OV$ and $DV$, which is $0.975 \times 16 + 0.925 \times 10 + 0.908 \times 12 + 0.45 \times 6 + 0.8 \times 11 + 2.1 \times 29 + 1.15 \times 15 = 125.396$. The length of $OV$ is $\sqrt{(0.975^2 + 0.925^2 + 0.908^2 + 0.45^2 + 0.8^2 + 2.1^2 + 1.15^2)}$

= 3.034, whereas the length of $DV$ is $\sqrt{(16^2 + 10^2 + 12^2 + 6^2 + 11^2 + 29^2 + 15^2)} = 41.51$. Hence, $SIM(D, O)$ is 0.9956 = 125.396 / (3.034 × 41.51).

Consider the car-ads application ontology $O$ again and the Web document $D_b$ as shown in Figure 2(b). The coefficients of $OV$ are always the same (see previous paragraph). The coefficients of $DV$ for $D_b$ are 6, 0, 0, 2, 8, 0, and 11 (see the last column of Table 2), which are the frequencies of appearance of the anticipated lexical values in $D_b$. We calculate the inner product of the coefficients of $OV$ and $DV$, which is 0.975 × 6 + 0.925 × 0 + 0.908 × 0 + 0.45 × 2 + 0.8 × 8 + 2.1 × 0 + 1.15 × 11 = 25.8. The length of $OV$, as computed earlier, is 3.034, whereas the length of $DV$ is $\sqrt{(6^2 + 0^2 + 0^2 + 2^2 + 8^2 + 0^2 + 11^2)} = 15$. Hence, $SIM(D, O)$ is 0.5669 = 25.8 / (3.034 × 15).

### 3.3 $H_3$: Grouping Heuristic

A document $D$ may have a high density measure for an ontology $O$, may also have a high expected-values measure for $O$, and still not be considered as a multiple-record document for $O$. This is because the values must also form groups that can be recognized as records for $O$. As a simple heuristic to determine whether the recognized values are interleaved in a way that could be considered consistent with potential records of $O$, we consider the sequence of values in a document that should appear at most once in each record and measure how well they are grouped.

We refer to an object set whose values should appear at most once in a record as a *1-max lexical object set*. Maximum participation constraints in an ontology constrain the values of the 1-max object sets to appear at most once in a record. For example, in the car-ads application ontology, the 1-maximum on *Car* in the relationship set *Car [0:0.975:1] has Year [1:*]* specifies that $Year$ is a 1-max object set. Other 1-max lexical objects in the car-ads ontology are $Make$, $Model$, $Mileage$, and $Price$.

Instead of counting the number of 1-max lexical objects in an application ontology $O$, a more adequate counting approach is to sum the average values expected for the 1-max objects in $O$. Since the average values expected for $Year$, $Make$, $Model$, $Mileage$, and $Price$ in the car-ads ontology are 0.975, 0.925, 0.908, 0.45, and 0.8, respectively, the anticipated number of lexical values from these object sets in a car advertisement is 4.058. To obtain an expected group size, we truncate the decimal value of the sum.

The expected group size $n$ is an estimate of the number of 1-max object-set values we should encounter in a document within a single record. On the average, each record should have $n$ 1-max object sets. Thus, if we list all recognized 1-max object-set values in the order they occur in a document $D$ and divide this sequence into groups of $n$, each group should have $n$ values from $n$ different object sets. The closer a document comes to this expectation, the better the grouping measure should be. For the multiple-record car-ads Web document in Figure 2(a),

Year: 2000
Year: 1989
Make: Subaru
Model: SW
-- Nr of Distinct "One Max" Object Sets: 3

Make: HONDA
Model: ACCORD
Mileage: 100000
Year: 1999
-- Nr of Distinct "One Max" Object Sets: 4

Price: 1900
Year: 1998
Model: Elantra
Year: 1994
-- Nr of Distinct "One Max" Object Sets: 3

Model: GRAND AM
Mileage: 27000
Price: 14000
Year: 1953
-- Nr of Distinct "One Max" Object Sets: 4

(a) First four groups of 1-max lexical values extracted from Figure 2(a).

Year: 1999
Year: 1998
Year: 1960
Mileage: 10000
-- Nr of Distinct "One Max" Object Sets: 2

Year: 1950
Price: 27850
Price: 19990
Year: 1980
-- Nr of Distinct "One Max" Object Sets: 2

Mileage: 401000
Year: 1940
Price: 17500
Price: 10971
-- Nr of Distinct "One Max" Object Sets: 3

Price: 79850
Price: 42990
Price: 129650
Price: 78850
-- Nr of Distinct "One Max" Object Sets: 1

(b) First four groups of 1-max lexical values extracted from Figure 2(b).

Figure 3: Groups of 1-max lexical values extracted from advertisement Web documents.

Figure 3(a) shows the first four groups of 1-max lexical object-set values extracted from the document. Similarly, Figure 3(b) shows the first four groups of 1-max lexical object-set values extracted from the document in Figure 2(b).

We measure how well the groups match the expectations with a grouping factor (denoted $G_{factor}$), which is calculated as follows:

$$G_{factor}(D, O) = \frac{\text{Sum of Distinct Lexical Values in Each Group}}{\text{Number of Groups} \times \text{Expected Number of Values in a Group}}$$

For example, the number of extracted groups from the Web document $D_a$ in Figure 2(a) is 13 (1 group of 2, 5 groups of 3, and 7 groups of 4). Since the number of anticipated lexical values in each group is four, $G_{factor}$ of $D_a$ is

$$G_{factor}(D_a, O) = \frac{(2 \times 1) + (3 \times 5) + (4 \times 7)}{(1 + 5 + 7) \times 4} = 0.8653$$

By way of comparison, the number of extracted groups from the Web document $D_b$ in Figure 2(b) is 4 (1 group of 1, 2 groups of 2, and 1 group of 3). Since the number of anticipated lexical values in each group is four, $G_{factor}$ of $D_b$ is

$$G_{factor}(D_b, O) = \frac{(1 \times 1) + (2 \times 2) + (3 \times 1)}{(1 + 2 + 1) \times 4} = 0.5$$

### 3.4 Combining Heuristics

The result we obtain when we run the heuristics on a Web document for an application ontology is a triple of heuristic measures: ($H_1$, $H_2$, $H_3$). For example, when $O$ is the car-ads application ontology and the Web document is $D_a$ in Figure 2(a), the heuristic-measure triple is (0.3056, 0.9956, 0.8653). For the Web document $D_b$ in Figure 2(b), the triple is (0.0734, 0.5669, 0.5).

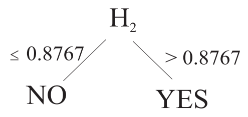
To combine the three heuristics into a single decision rule, we used C4.5 [Qui93], a popular machine learning algorithm. C4.5 is a rule post-pruning decision tree algorithm. The learning task is to judge the suitability of a Web document for a given application ontology, (i.e. to return "YES" when it is suitable, and to return "NO" otherwise). The performance measure is the percent of documents correctly classified when using a generated rule (i.e. the accuracy). The bias favors the shortest rule, so that if several rules are equally accurate, a decision tree with the fewest branches is chosen. The training experience is a set of Web documents classified by a human expert in the application domain.

We represent every instance of a Web-document/application-ontology pair in both training and test sets as a triple ($H_1$, $H_2$, $H_3$) composed of the measures returned by the three heuristics. We trained C4.5 with 20 positive examples and 30 negative examples for each of our two application ontologies—car ads and obituaries. (Appendix B lists the URLs for these sites.) The positive examples included in the training sets came from 20 different sites, two each selected arbitrarily from 10 different geographical regions in the United States. To select 20 of the 30 negative examples, we first chose 20 document subjects (see Table 3) and then found a Web page for each subject. Because we wanted to be able to make fine distinctions when filtering documents, we chose most of the subjects based on a perceived similarity between the subject and either car-ads or obituaries. To make sure that gross distinctions were also filtered properly, we also chose a few documents "arbitrarily." In addition to the negative examples in Table 3, we also used 10 car-ads documents (one from each region) to play the role of 10 negative obituary examples and 10 obituary documents (one from each region) play the role of 10 negative car-ad examples.

Based on the 50 training examples for car-ads application ontology, C4.5 generated the following rule:

<u>Rule 1</u>

$H_2 \leq 0.8767$: NO
$H_2 > 0.8767$: YES




Thus, our filtering technique selects a document as a car ad if its expected-values measure is

| *Document* | *Document Subject* | *Document* | *Document Subject* |
|---|---|---|---|
| 1 | Any Advertisement List | 11 | Announcement |
| 2 | Employment Classified | 12 | Personal Classified |
| 3 | Real Estate Classified | 13 | Student List |
| 4 | Bike and Cycles Classified | 14 | People Features |
| 5 | Service Classified | 15 | Sport Reports |
| 6 | Pets Classified | 16 | School News |
| 7 | Computer Classified | 17 | Event and Festival |
| 8 | Vehicle Part Classified | 18 | Generational Personality Types |
| 9 | Rental Classified | 19 | Wedding List |
| 10 | Employment Agencies | 20 | Birth List |

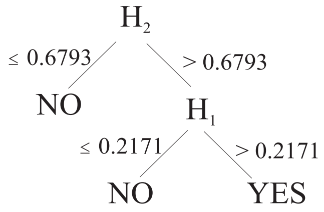
Table 3: Negative examples in training sets.

greater than 0.8767 (i.e. if the cosine between the car-ads ontology vector and the document vector is greater than 0.8767).

Based on the 50 training examples for the obituary application ontology, C4.5 generated the following rule:

<u>Rule 2</u>

$H_2 \leq 0.6793$: NO
$H_2 > 0.6793$
| $H_1 \leq$ 0.2171: NO
| $H_1 > 0.2171$: YES

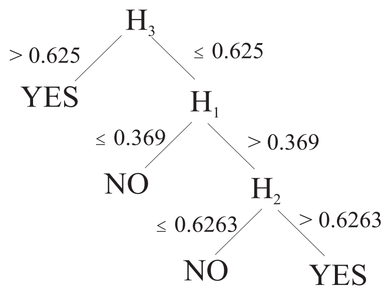



Thus, our filtering technique selects a document as an obituaries document only if its expected-values measure is greater then 0.6793 and its density measure is greater than 0.2171.

Searching for a potential universal rule over all ontologies, we combined the 50 training triples for car ads and the 50 training triples for obituaries, and applied the C4.5 algorithm to produce the following decision rule:

<u>Rule 3</u>

$H_3 \leq 0.625$
| $H_1 \leq$ 0.369: NO
| $H_1 > 0.369$
|| $H_2 \leq$ 0.6263: NO
|| $H_2 > 0.6263$: YES
$H_3 > 0.625$: YES




To use Rule 3 for an application ontology $A$ for a Web document $W$, we would obtain the heuristic triple ($H_1$, $H_2$, $H_3$) for $W$ with respect to $A$ and apply Rule 3. Then, our filtering technique would classify $W$ as suitable for $A$ if the grouping measure ($H_3$) is greater than 0.625 or if the grouping measure ($H_3$) is less than 0.625, the density measure ($H_1$) is greater than 0.369,

| *Document* | *Document type* | *Document* | *Document type* |
|---|---|---|---|
| 1 | Person | 6 | Items for Sale or Rent |
| 2 | Search Engine | 7 | Personals |
| 3 | Purchase Form | 8 | Motocycles |
| 4 | Missing people | 9 | Boats |
| 5 | Jobs | 10 | Antique Car |

Table 4: Negative examples in test sets.

and the expected-values measure ($H_2$) is greater than 0.6263.

# 4 Results and Discussion

To test the machine-learned rules, we chose 30 test documents—10 positive documents for car ads, 10 positive documents for obituaries, and 10 negative examples. (Appendix C lists the URLs for these sites.) We chose the 10 positive examples for car ads and the 10 positive examples for obituaries from sites located in the ten US geographical regions we had previously designated for training sets. The test sites, of course, were different from the training sites even though they were located in the same geographical regions. For the negative test documents, we selected documents based on the subjects listed in Table 4. We purposely chose some of these subjects to be fairly close to either car ads or obituaries. Indeed, the page selected for antique cars turned out to be "too close" to car ads (even a human expert could not tell the difference), and we later classified it as a positive example for car ads. We also used the 10 car-ads positive documents in the test set as 10 negative obituary documents and vice versa.

## 4.1 Experimental Results

Generated Rules 1 and 2 successfully filtered the test set for both the car-ads application ontology and the obituary application ontology at the same level of accuracy, 96.7%. The precision for the car-ads application ontology was 100%, and the recall was 91%. The precision for the obituary ontology application was 91%, and the recall was 100%. We also applied Rule 3, the generated universal rule, to the test set. The accuracy for the Rule 3 was 93.4%, the precision was 84%, and the recall was 100%.

Tables 5 and 6 show the test results for the car-ads application ontology, and Tables 7 and 8 show the test results for the obituaries ontology. For the tables describing the positive examples, the first column gives the document location, while the first column of the tables for the negative examples gives the document subject. The second, third, and forth columns of all these tables show the values of the three heuristic measures for a Web document. The last column of each table shows the results computed by the C4.5 generated rules—Rule 1 for Tables 5 and 6 and Rule 2 for Tables 7 and 8. Observe that there is one positive example for car ads that is judged

| *Region* | *Density Heuristic* | *Expected-Values Heuristic* | *Grouping Heuristic* | *Generated Filtering Rule* |
|---|---|---|---|---|
| East Coast | 0.3287 | 0.9345 | 0.8929 | Yes |
| North East Central | 0.4614 | 0.7958 | 0.8029 | No |
| North East Coast | 0.2079 | 0.8855 | 0.8043 | Yes |
| North West Central | 0.2153 | 0.9837 | 0.9167 | Yes |
| Pacific North West | 0.3967 | 0.8881 | 0.8718 | Yes |
| Pacific South West | 0.3032 | 0.9309 | 0.7824 | Yes |
| South East Central | 0.1445 | 0.945 | 0.90625 | Yes |
| South East Coast | 0.3659 | 0.9658 | 0.85 | Yes |
| South West Central | 0.2271 | 0.8867 | 0.7125 | Yes |
| West Coast | 0.2382 | 0.9909 | 0.9231 | Yes |
| Antique Car | 0.1058 | 0.8782 | 0.7813 | Yes |

Table 5: Car ontology test results: car positive examples.

incorrectly (a false drop) and one negative example for obituaries that is judged incorrectly (a false positive).

### 4.2 Discussion

We discuss the two documents judged incorrectly in Sections 4.2.1 and 4.2.2 and provide reasons for discrepancies and insight into how these exceptional cases could be handled. In Section 4.2.3 we discuss our assumption about multiple records being in the document. Finally, in Section 4.2.4 we discuss our views on a universal rule versus application-ontology-dependent rules.

#### 4.2.1 False Drop Example

Figure 4 displays the the car-ads document for which Rule 1 gives a false drop. Observe that the "last" car ad is "quite different." It is not a single car ad; instead it is a dealer ad for several dozen cars. This, by itself, is not a problem, but there are three complications that do cause problems. (1) The "Year" and the "Price" in this dealer ad in Figure 4 are concatenated. Our *Year* data frame in Figure 1(b) did not anticipate this concatenation, and thus the years were not recognized. (2) These ads contain neither mileage information nor feature information about the cars. Missing mileage and feature information is generally not a problem, but since there are more cars mentioned in the "last" ad than in all the rest of the car ads together, missing mileage and feature information adds up. (3) The phone number is factored out of each individual ad within the "last" ad—it is the same for all dealership cars. As a result of these three problems, most of the cars in these three documents have only a *Make*, *Model*, and *Price*. Even so, the 0.7958 measure for the expected-values heuristic is almost high enough to be acceptable (see Rule 1).

Based on this discussion, we can see that the Web page in Figure 4 violates some assumptions

| *Document Subject* | *Density Heuristic* | *Expected-Values Heuristic* | *Grouping Heuristic* | *Generated Filtering Rule* |
|---|---|---|---|---|
| Person | 0.0642 | 0.4682 | 0.5 | No |
| Search Engine | 0.0115 | 0.7582 | 0 | No |
| Purchase Form | 0.0053 | 0.3738 | 0.4167 | No |
| Missing People | 0.053 | 0.8319 | 0.4286 | No |
| Jobs | 0.0242 | 0.5055 | 0.447 | No |
| Items for Sale or Rent | 0.0733 | 0.5669 | 0.5 | No |
| Personals | 0.1566 | 0.7785 | 0.4167 | No |
| Motocycles | 0.2061 | 0.6266 | 0.61 | No |
| Boats | 0.0917 | 0.6605 | 0.5484 | No |
| Obituary | 0.0237 | 0.5428 | 0.5 | No |
| Obituary | 0.0460 | 0.4281 | 0.4074 | No |
| Obituary | 0.0288 | 0.5513 | 0.4375 | No |
| Obituary | 0.0326 | 0.4534 | 0.5 | No |
| Obituary | 0.0263 | 0.4668 | 0.4388 | No |
| Obituary | 0.0220 | 0.4597 | 0.4479 | No |
| Obituary | 0.0465 | 0.5267 | 0.5 | No |
| Obituary | 0.0308 | 0.5307 | 0.5 | No |
| Obituary | 0.0343 | 0.3213 | 0.25 | No |
| Obituary | 0.0156 | 0.4613 | 0.5 | No |

Table 6: Car ontology test results: car negative examples.

we have made in our filtering process. To filter such Web documents as car ads, we must make some adjustments. One adjustment would be to alter the regular expressions for year to be more forgiving of unexpected concatenations. Another adjustment would be to allow some documents to be classified as "maybe" when they are "close" to the threshold values specified in the generated judging rules. Indeed, if we use Rule 3, which has more tolerance (but therefore lower precision), the document in Figure 4 is judged as a car-ad document. Finally, another adjustment would be to identify that the phone number is factored and distribute it to each individual car ad within the dealer ad—not a simple task to do robustly and automatically. (We leave all these adjustments for future work.)

### 4.2.2 False Positive Example

Table 8 shows that one negative example, "Missing People," is misjudged. The document consists of a list of descriptions for missing persons. Each description typically contains several lexical objects that are defined in our obituary ontology—name, birth date, and age. Although other special lexical objects exist only in obituaries (e.g. interment, funeral, relative-name list), the precision for these lexical objects is much lower than the precision of the lexical objects such as date and age. Hence, "thinking" that it is working on an obituary, the obituaries application ontology extracts places and times in the missing-people document that it "thinks" are lexical

| *Region* | *Density Heuristic* | *Expected-Values Heuristic* | *Grouping Heuristic* | *Generated Filtering Rule* |
|---|---|---|---|---|
| East Coast | 0.3622 | 0.7983 | 0.8571 | Yes |
| North East Central | 0.4108 | 0.7542 | 0.56 | Yes |
| North East Coast | 0.2435 | 0.8106 | 0.7143 | Yes |
| North West Central | 0.2458 | 0.7761 | 0.75 | Yes |
| Pacific North West | 0.278 | 0.7926 | 0.7626 | Yes |
| Pacific South West | 0.3864 | 0.8297 | 0.7589 | Yes |
| South East Central | 0.3207 | 0.7084 | 0.8438 | Yes |
| South East Coast | 0.3112 | 0.7459 | 0.9531 | Yes |
| South West Central | 0.5043 | 0.711 | 0.8125 | Yes |
| West Coast | 0.3863 | 0.7466 | 0.8235 | Yes |

Table 7: Obituary rule test results: positive examples.

objects for interment and funeral places and times, and it extracts names it "thinks" are relative names. Thus, all the heuristic measurements are artificially inflated and the document is judged incorrectly. To adjust for this problem, we could consider using extraction confidence factors based on precision and recall to (1) ignore low-confidence attributes, (2) adjust the artificially inflated values downward, or (3) weight high-confidence attributes higher. (As for the earlier adjustments we have recognized, we leave adjusting for extraction confidence to future work.)

If someone considers antique car sales a false positive for car ads, we have no good suggestions on how to modify our filtering process to solve this problem. Our downstream operations would extract the information from antique car ads and make it available to query by SQL. Any SQL query for late model cars would certainly exclude all antique cars for someone not looking for old cars.

### 4.2.3 Singleton Document

When we define an application ontology for use in our system, we assume that the Web documents are multiple-record Web documents. Nevertheless, we wondered what would happen if we were to apply our filtering process to Web documents containing information for only a single record. Since singleton car ads are rare (if they exists at all), we only tested for obituaries. We selected nine "arbitrary" singleton obituaries plus an obituary for Princess Diana. (Appendix D lists the URLs for these sites.) Besides Princess Diana, three of the other nine can also be considered to be famous: John Atanasoff who invented the first electronic computer, Jennifer Paterson who was a TV chef known to millions as one half of the celebrated culinary duo the "Two Fat Ladies," and Vincent O. ("Vinny") Marino who was a former heroin addict and founder of one of California's most successful rehabilitation centers.

Table 9 shows the results of applying our obituary ontology to these singleton obituaries. Using Rule 2, we see that 50% of the singleton obituaries are judged as obituaries while 50% are not.

| *Document Subject* | *Density Heuristic* | *Expected-Values Heuristic* | *Grouping Heuristic* | *Generated Filtering Rule* |
|---|---|---|---|---|
| Person | 0.0713 | 0.6511 | 0 | No |
| Search Engine | 0.0636 | 0.6531 | 0 | No |
| Purchase Form | 0.0416 | 0.6651 | 0.5 | No |
| Missing People | 0.2283 | 0.7769 | 0.5893 | Yes |
| Jobs | 0.1551 | 0.6307 | 0.6711 | No |
| Items for Sale or Rent | 0.0839 | 0.6528 | 0.75 | No |
| Personals | 0.0784 | 0.6397 | 0.3929 | No |
| Motocycles | 0.3556 | 0.343 | 0.8825 | No |
| Boats | 0.1218 | 0.5323 | 0.5541 | No |
| Antique Car | 0.3774 | 0.6437 | 0.5357 | No |
| Car | 0.1485 | 0.5197 | 0.4166 | No |
| Car | 0.5528 | 0.3520 | 0.5 | No |
| Car | 0.1669 | 0.2991 | 0.5 | No |
| Car | 0.3465 | 0.4192 | 0.5385 | No |
| Car | 0.4445 | 0.3433 | 0.4401 | No |
| Car | 0.39 | 0.1995 | 0.3728 | No |
| Car | 0.1681 | 0.5266 | 0.4375 | No |
| Car | 0.4419 | 0.4719 | 0.5046 | No |
| Car | 0.2294 | 0.3984 | 0.5333 | No |
| Car | 0.2170 | 0.4690 | 0.4167 | No |

Table 8: Obituary rule test results: negative examples.

The first column of the tables gives the name of the deceased person. The "(F)" means that the obituary is for a famous person. From the table, we can see that all the obituaries about famous people are dropped in the filtering process. For the obituary of Princess Diana, the expected-values heuristic is below the threshold selected in Rule 2, but only by 0.0036. For the rest, all the density heuristics for the single famous-person obituaries are lower than the threshold. We observe (as might be expected) that obituaries for famous people are considerably longer than obituaries for ordinary people, which directly affects the density as the verbiage increases and contains correspondingly less of the kind of text expected in and recognized by the obituary application ontology. We were curious about Clara Griffin, who is not famous. Upon closer investigation, we discovered that this particular obituary is embedded in a page with about as much additional text as is in the obituary itself. (In all other cases, the singleton obituaries were in a frame by themselves.) Thus Clara Griffin's obituary has the same characteristics as famous people—it includes considerable extra verbiage not recognized as being text typically found in a "standard" obituary.

### 4.2.4 Universal Rule

Test results for Rule 3 show that the accuracy and recall of this "universal rule" remain high, above 90%, but that the precision drops to 84%. Since this rule spans application ontologies, it

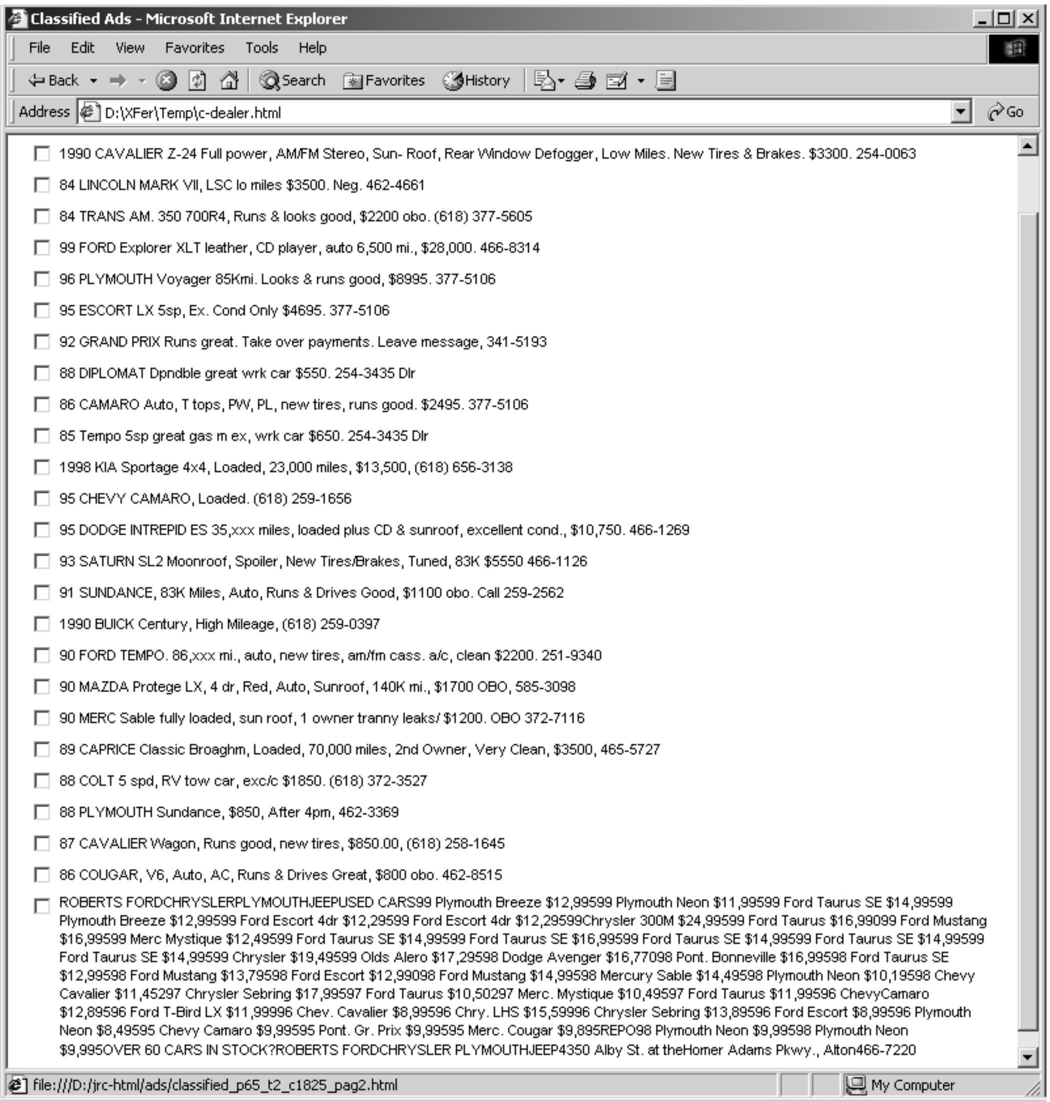



Figure 4: Car-ads document with false drop.

may be useful to apply Rule 3 for a new application ontology. However, since both the extraction precision and the three heuristic measures have some differences for different ontology applications, we suggest using application-dependent rules, such as Rule 1 for car ads and Rule 2 for obituaries, to filter suitable documents.

# 5 Concluding Remarks

We presented an approach to filtering multiple-record Web documents. The approach requires an ontological description of the application, which typically requires a few dozen hours to create. Once created, we can train a machine-learning algorithm over a triple of heuristics (density, expected-values, grouping) to produce a decision tree that accurately filters multiple-record documents for the ontology. Results for the tests we conducted showed that the filtering accuracy was above 95% with recall and precision above 90% for both of our applications.

| *Person Name* | *Density Heuristic* | *Schema Heuristic* | *Grouping Heuristic* | *Generated Rule* |
|---|---|---|---|---|
| Sam J. Humpheiwa | 0.3058 | 0.7917 | 0.6250 | Yes |
| Clara Griffin | 0.1842 | 0.7693 | 1.00 | No |
| John Atanasoff (F) | 0.0923 | 0.7555 | 0.55 | No |
| Jennifer Paterson (F) | 0.1109 | 0.7040 | 0.75 | No |
| Kavin A. Gobrech | 0.3076 | 0.7294 | 0.75 | Yes |
| John Ayles | 0.2637 | 0.7698 | 0.6 | Yes |
| Gaines Mr. O.D. | 0.3422 | 0.7193 | 0.75 | Yes |
| Robert M. Harle | 0.3628 | 0.7528 | 0.875 | Yes |
| Vincent O. Marino (F) | 0.1326 | 0.7788 | 0.75 | No |
| Diana, Princess of Wales (F) | 0.1075 | 0.6757 | 0.75 | No |

Table 9: Single obituary test results, where $F$ denotes a famous person

## References


[Bun77] M.A. Bunge. *Treatise on Basic Philosophy: Vol. 3: Ontology I: The Furniture of the World*. Reidel, Boston, 1977.

[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.

[EFKR99] D.W. Embley, N. Fuhr, C.-P. Klas, and T. Roelleke. Ontology suitability for uncertain extraction of information from multi-record web documents. In *Proceedings of the Workshop on Agenten, Datenbanken und Information Retrieval (ADI'99)*, Rostock-Warnemuende, Germany, 1999.

[EJN99] D.W. Embley, Y.S. Jiang, and Y.-K. Ng. Record-boundary discovery in Web documents. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, pages 467–478, Philadelphia, Pennsylvania, 31 May - 3 June 1999.

[Emb80] D. Embley. Programming With Data Frames for Everyday Data Items. In *Proceedings of the 1980 National Computer Conference*, pages 301–305, 1980.

[LH98] W. Lam and C.Y. Ho. Using a generalized instance set for automatic text categorization. In *Proceedings of the Twenty-First International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 81–89, Melbourne, Australia, 1998.

[Qui93] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, California, 1993.

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

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

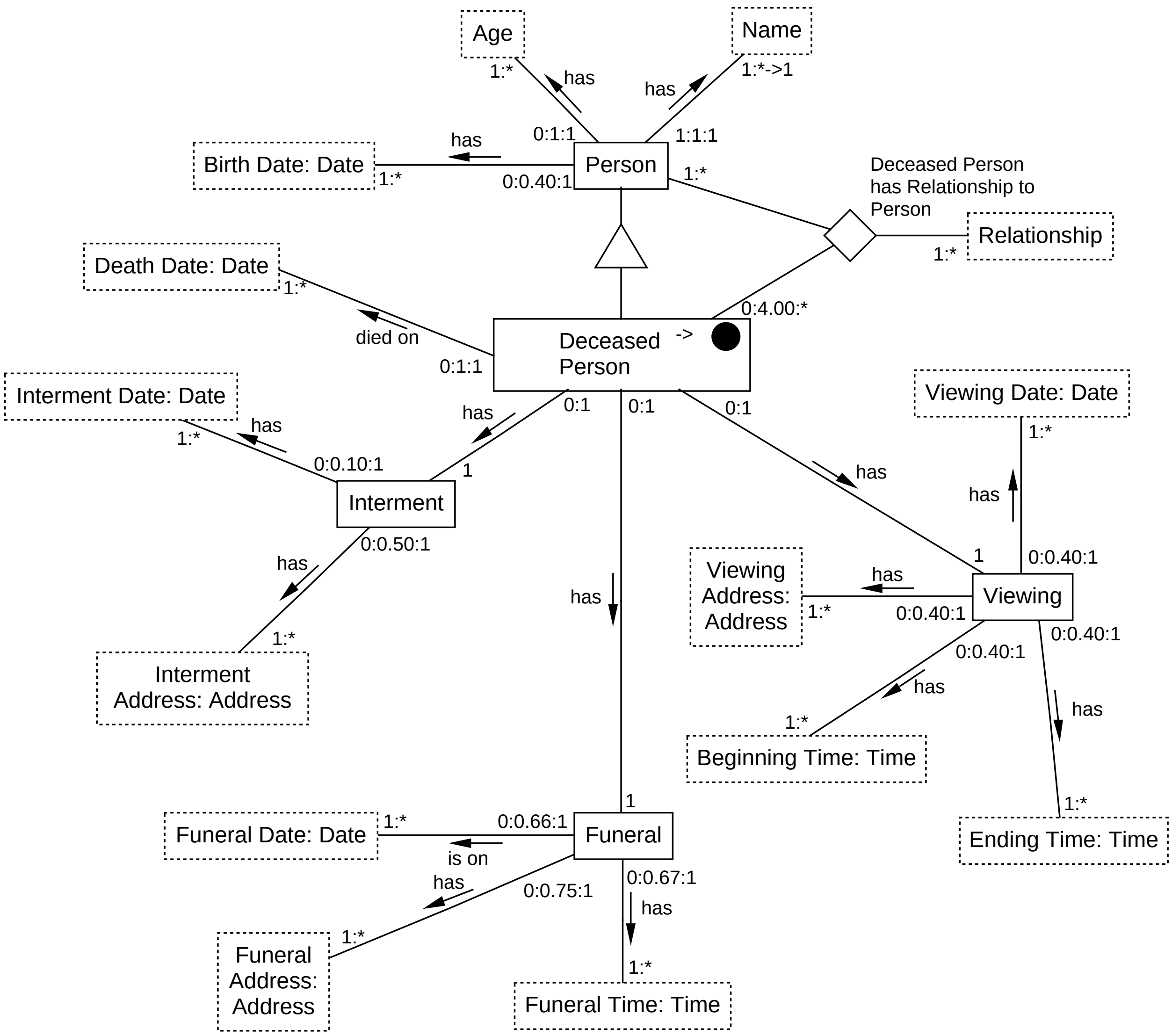
[Wan89] Y. Wand. A proposal for a formal model of objects. In W. Kim and F.H. Lochovsky, editors, *Object-Oriented Concepts, Databases, and Applications*, pages 537–559. ACM Press, New York, 1989.

[WZL99] K. Wang, S. Zhou, and S.C. Liew. Building hierarchical classifiers using class proximity. In M.P. Atkinson, M.E. Orlowska, P. Valduriez, S.B. Zdonik, and M.L. Brodie, editors, *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB'99)*, pages 363–374. Morgan Kaufmann, 1999.

[YL99] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the Twenty-Second International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, Berkley, California, 1999.

# Appendix

## A Obituary Application Ontology—Graphical



## B URLs of Web Documents Used in the Training Set

| Negative Training Documents | |
|---|---|
| *Document Subject* | *URL* |
| Any Advertisement List | http://www.ridenbaugh.com/mv/class.htm |
| Employment Classified | http://www.brewtonstandard.com/ |
| Real Estate Classified | http://www.adn.com/classads/ads/today/110.html |
| Bike and Cycles Classified | http://www.thnews.com/205.html |
| Service Classified | http://www.aspendailynews.com/ |
| Pets Classified | http://198.243.33.129:591/ |
| Computer Classified | http://www.zip2.com/gazette/ |
| Vehicle Part Classified | http://www.elktruth.com/ |
| Rental Classified | http://www.bouldernews.com/ |
| Employ Agency List | http://www.zip2.com/ |
| Announcement | http://www.mountainhomenews.com |
| Personal Classified | http://www.idahoclassified.com/cv3/idaho |
| Student List | http://www.craigdailypress.com/section/people/story/ 10001438 |
| People Features | http://www.browncountyindiana.com/update/features.html |
| Sport Reports | http://www.news-banner.com/sports.html |
| School News | http://www.news-banner.com/school.html |
| Event and Festival | http://www.bestofneworleans.com/frames/capMain.html |
| Generational Personality Types | http://www.seanet.com/Users/pamur/types.html |
| Wedding List | http://www.benewsjournal.com/weddings/weddings.html |
| Birth List | http://www.herald-review.com/0/birthdec-9.html |

| Positive Training Documents—Cars | |
|---|---|
| *Document Region* | *URL* |
| East Coast | http://www.delmarvaclassifieds.com<br>http://www.capitalonline.com/ |
| North East Central | http://www.thetelegraph.com/<br>http://classifieds.cnpapers.com/classsta.html |
| North East Coast | http://www.vermontclassifieds.com<br>http://www.journaltribune.com/classifieds/0110.html |
| North West Central | http://www.adquest3d.com/<br>http://www.ndonline.com/TribWebPage/WebClassAds/window.html |
| Pacific North West | http://www.oregonlive.com<br>http://www.heraldandnews.com/ |
| Pacific South West | http://www.sacbee.com/classads/ads/today/1350.html<br>http://www.avpress.com/c/c803a.htm |
| South East Central | http://www.gomemphis.com/<br>http://classifieds.bgdailynews.com/ |
| South East Coast | http://www.elkintribune.com/<br>http://www.n-jcenter.com/class/975.htm |
| South West Central | http://classifieds.amarillonet.com<br>http://www.panews.com/FrameSets/ClssFrmSt.html |
| West Coast | http:/www.azcentral.com<br>http://www.greeleytrib.com/classifieds/webclassads/925.html |

| Positive Training Documents—Obituaries | |
|---|---|
| *Document Region* | *URL* |
| East Coast | http://www.roanoke.com/roatimes/obits.html<br>http://www.bristolnews.com/obituaries/122899.html |
| North East Central | http://147.72.93.138/<br>http://www.CopleyNewspapers.com/BeaconNews/index.html |
| North East Coast | http://www.addisonindependent.com/Obituaries.html<br>http://www.eveningtribune.com/extras/obits.shtml |
| North West Central | http://www.startribune.com<br>http://www.duluthnews.com/today/dnt/local/robi.htm |
| Pacific North West | http://classifieds.bellinghamherald.com/obits.htm<br>http://www.thedailyworld.com/obituaries/ |
| Pacific South West | http://www.sacbee.com/classads/ads/today/105.html<br>http://www.latimes.com/obituary/ |
| South East Central | http://columbiadailyherald.com/obits/docs/<br>http://www.timesdaily.com/obits/stories/991228o.html |
| South East Coast | http://www.tdn-net.com/genealogy/obits2/obits.htm<br>http://www.newscoast.com/obitsstory.cfm?ID=20043 |
| South West Central | http://www.amarillonet.com/obits/<br>http://www.ent-net.com/news/local/deaths.htm |
| West Coast | http://www.boulderplanet.com/current/1999/19990704/<br>essentials/story8.html<br>http://www.azcentral.com/community/obits/ |

# C URLs of Web Documents Used in the Test Set

| Negative Test Documents | |
|---|---|
| *Document Subject* | *URL* |
| Person | http://www.cs.byu.edu/info/dwembley.html |
| Search Engine | http://www.dogpile.com |
| Purchase Form | https://www.ecampus.com |
| Missing People | http://www.cyberpages.com |
| Jobs | http://www.ohio.com |
| Items for Sale or Rent | http://www.crookstontimes.com |
| Personals | http://www.date-net.com/index.CFM |
| Motocycles | http://www.netbikes.yks.com/general.html |
| Boats | http://www.sunspot.net/ |
| Antique cars | http://www.internetclassiccars.com/files/twenties0.html |

| Positive Car-Ads Documents | |
|---|---|
| *Document Region* | *URL* |
| East Coast | http://www.delmarvaclassifieds.com |
| North East Central | http://www.thetelegraph.com/ |
| North East Coast | http://www.vermontclassifieds.com |
| North West Central | http://www.ndweb.com/mdnonline/mdnonline.html |
| Pacific North West | http://www.adn.com/ |
| Pacific South West | http://www.hawaiisnews.com/cars/ |
| South East Central | http://www.brewtonstandard.com |
| South East Coast | http://www.aikenstandard.com/ |
| South West Central | http://adaeveningnews.com/ |
| West Coast | http://www.tahoe.com/classifieds/tdt/9100.html |

| Positive Obituary Documents | |
|---|---|
| *Document Region* | *URL* |
| East Coast | http://blandnews.com/obit.htm |
| North East Central | http://www.altonweb.com/bb/obituaries/index.html |
| North East Coast | http://rutlandherald.nybor.com/Obits/3327.html |
| North West Central | http://www.bismarcktribune.com/tribwebpage/news/jan2000/1-1obit.html |
| Pacific North West | http://www.thesunlink.com/local/obits/index.html |
| Pacific South West | http://www.honoluluadvertiser.com/obituaries.html |
| South East Central | http://www.andalusiastarnews.com/obituaries/2000/01/0112obituaries.html |
| South East Coast | http://www.aikenstandard.com/obits.html |
| South West Central | http://www.eurekaherald.com/eureka/obits.html |
| West Coast | http://www.lasvegassun.com/sunbin/stories/archives/1999/oct/28/509480286.html?obituary |

# D Singleton Obituaries

| Singleton Obituaries | |
|---|---|
| *Person* | *URL* |
| Sam J. Humpheiwa | http://www.obits.com/ |
| Clara Griffin | http://www.townonline.com/south/abington/obits/obiNOgriffin-0113AR_10.html |
| John Atanasoff | http://www.cs.iastate.edu/jva/jva-obit.shtml |
| Jennifer Paterson | http://news2.thls.bbc.co.uk/hi/english/uk/newsid |
| Kavin A. Gobrech | http://www.arrangements.com/pa/17350/funeral/feiser/gobrech.htm |
| John Ayles | http://www.memorialwebpark.com/cgi-bin/showmem.pl?id=000000016A |
| Gaines Mr.O.D | http://www.cribbs.net/obitsearch/ |
| Robert M. Harle | http://examiner.net/stories/020200/obi_020200011.shtml |
| Vincent O | http://starbulletin.com/2000/02/01/news/story6.html |
| Diana, Princess of Wales | http://www.royal.gov.uk/family/diana.htm |