Recognizing Ontology-Applicable Multiple-Record Web Documents

D.W. Embley, 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

Automatically recognizing which Web documents are "of interest" for some specified application is non-trivial. As a step toward solving this problem, we propose a technique for recognizing which multiple-record Web documents apply to an ontologically specified application. Given the values and kinds of values recognized by an ontological specification 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 and kind of values found in a document to the number and kind expected by the application ontology, and (3) a grouping heuristic that considers whether the values of the document appear to be grouped as application-ontology records. Then, based on machine-learned rules over these heuristic measurements, we determine whether a Web document is applicable for a given ontology. Our experimental results show that we have been able to achieve over 90% for both recall and precision, with an F-measure of about 95%.

1 Introduction

The World Wide Web contains abundant repositories of information in Web documents—indeed, it contains so much, that locating information "of interest" for an application becomes a huge challenge. Even sorting through a tiny subset of Web documents is overwhelming. How can we automatically select just those documents that have the needed information for an application?

When we construct automated processes to recognize which documents apply to a user's information needs, we must be careful not to discard relevant documents and not to accept irrelevant documents. A process that discards too many relevant documents has poor *recall*—ratio of the number of relevant documents accepted to the total number of relevant documents. A process that accepts too many irrelevant documents has poor *precision*—ratio of the number of relevant documents accepted to the total number of documents accepted. The harmonic mean[1] of the precision and recall, which is called the *F-measure* [BYRN99], is a standard way to combine precision

[1] $F = \frac{2}{\frac{1}{recall} + \frac{1}{precision}}$

and recall. We wish to have an automated recognition process that has a good F-measure so that it has both high recall and high precision.

In this paper we propose an approach for recognizing whether a Web document is relevant for a chosen application of interest. We base our approach on application ontologies [ECJ+99], which are conceptual-model snippets [Wan89, SDUS98] of standard ontologies [Bun77, Bun79], and we apply techniques from information retrieval [SM83, BYRN99] and machine learning [Qui93].

Our work reported here is also partly motivated by our success in using application ontologies to extract information from unstructured multiple-record Web documents and structure the information so that it can be queried using a standard query language [ECJ+99]. 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%, and we have achieved robustness over a wide range of pages and pages that change in format, content, and style [ECJ+99]. In these experiments, however, we have assumed (and have made sure by human inspection) that the Web pages were multiple-record documents appropriate for the application we were using. Thus, in the context of our larger project, the purpose of this work is to automate applicability checking. If we can locate documents applicable to an ontology, we can apply techniques we have already developed, to extract, structure, query, and archive in databases, information found in data-rich, application-specific Web documents. Hence, the contributions of this work have the potential to be more far-reaching than just the salient contribution of increasing recall and precision in recognizing application-specific Web documents.

Our approach to document recognition is related to text classification [BB63]—each application ontology can be a class—but our work fundamentally differs from other text-classification work. 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 contain ads of interest for a predefined 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.

Despite this basic differences, we nevertheless compare our work with the work in text classification (e.g. [MLW92, TPL95, WPW95, HPS96, Joa96, LSC96, KS97, BM98, MG98, MRM98, BGG+99]) in order to highlight some advantages and disadvantages of the approaches that have been taken. Most text classification systems are based on machine learning. In typical machine-learning approaches, each document is transformed into a feature vector. Usually, each element of a feature vector represents a word from a corpus. The feature values may be binary, indicating presence or absence of the word in the document, or they may be integers or real numbers indicating some measure of frequency of the word's appearance in the text. This text representation is referred to as a "bag of words," which is used in most text-classifiers. A major difficulty for this bag-of-words approach is the high dimensionality of the feature space. Thus, it is highly desirable to reduce dimensionality of the space without sacrificing classification accuracy. Since our approach uses a predefined application ontology whose object sets constitute the features of interest, we immediately identify a space with comparatively small dimensionality and thus avoid this high-dimensionality problem. Further, our predefined application ontology also overcomes many of the limitations imposed by word-based techniques. There is no need to find object relevancy with respect to a corpus because the application ontology already defines the relationships among the conceptual objects. Moreover, our approach is sensitive to context and domain knowledge and can thus more effectively retrieve the relevant information from a document and use it to classify a document with respect to an application. For example, the basic idea of McCallum's Naive Bayes classifier [BM98], which is one of the most successful systems used in text classification applications and which is implemented in Rainbow [McC96], is to use the joint probabilities of words and categories, which are computed based on labeled training documents, to estimate the probability of categories given a document. However, the naive part of the approach is the assumption of word independence, which makes the classifier less appropriate for "high-precision" classifiers like ours. In compensation for these disadvantages, typical machine-learning approaches may take less user effort to produce—the effort being mainly the work to provide and label a set of training documents. Our experience in teaching others to use our system suggests that an application ontology of the kind we use can be created in a few dozen person-hours, which is roughly comparable to the time and effort it may take to label a set of training documents. Furthermore, the application ontology produced can also serve as an information extractor (see [ECJ+99]); and hence, little, if any, additional work is required to also create a classifier.

Although our work differs fundamentally from most text classifiers, as just discussed, the work reported in [RL94] takes an approach similar to ours in that it also attempts to do "high-precision" classification for information extraction. Like most text classifiers, [RL94] uses machine learning, but to obtain the desired high precision, considerably more effort must be expended to establish the basis for machine learning. Not only must documents be marked as relevant and non-relevant, but each individual relevant element plus the context for each individual relevant element must also be marked. In addition, an application-domain-specific dictionary must be created. The basic trade-off in human effort between our approach and the approach in [RL94] is the effort to tag the elements in the document and create the domain-specific dictionary versus the effort to create the application ontology.

Some recent work has been reported that uses machine learning with less human effort for doing "high-precision" classification for domain-specific search engines [MN99, MNRS99, MNRS00] and focused crawling [CvdBD99, Cha99]. By mostly using unsupervised learning, human effort can be greatly reduced. The challenge, however, is to reach high accuracy, and it may not be possible to achieve the accuracy that can be obtained with an ontology-based approach. Ultimately, some combination of the approaches may be best. In the meantime, we pursue our goal of high-precision binary classification based on ontological specifications.

We outline the rest of our paper as follows. 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. Given an application ontology and a set of Web documents, we automatically obtain statistics for three heuristics for each document: (1) a density heuristic, (2) an expected-values heuristic, and (3) a grouping heuristic. Section 3 describes these heuristics and the statistical measures we obtain for each, as well as the machine-learned decision-tree rules we obtain for judging document applicability. In Section 4 we discuss our experimental results—which, for the two applications we tried (car advertisements and obituaries), are in the 90% range for both recall and precision. In Section 5, we give concluding remarks.

2 Application Ontology

For our work in data extraction, we define an *application ontology* to be a conceptual-model instance that describes a real-world application in a narrow, data-rich domain of interest (e.g. car advertisements, obituaries, job advertisements) [ECJ+99]. Each of our application ontologies consists of two components: (1) an *object/relationship-model instance* that 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 potential contents of the object set. A data frame for an object set defines the lexical appearance of constant objects for the object set and establishes appropriate keywords that are likely to appear in a document when objects in the object set are mentioned. Figure 1 shows part of our car-ads application ontology, including object and relationship sets and cardinality constraints (Lines 1-8) and a few lines of the data frames (Lines 9-18).[2]

An object set in an application ontology represents a set of objects which may either be lexical or nonlexical. Data frames with declarations for constants that can potentially populate the object set represent lexical object sets, and data frames without constant declarations represent nonlexical object sets. *Year* (Line 9) and *Mileage* (Line 14) are lexical object sets whose character representations have a maximum length of 4 characters and 8 characters respectively. *Make*, *Model*,

[2]The full ontology for car ads is about 600 lines in length. Our obituary ontology, which is the other application ontology we discuss in this paper is about 500 lines in length, but it references both a first-name lexicon and a last-name lexicon, which each contain several thousand names.

```
 1. Car [-> object];
 2. Car [0:0.975:1] has Year [1:*];
 3. Car [0:0.925:1] has Make [1:*];
 4. Car [0:0.908:1] has Model [1:*];
 5. Car [0:0.45:1] has Mileage [1:*];
 6. Car [0:2.1:*] has Feature [1:*];
 7. Car [0:0.8:1] has Price [1:*];
 8. PhoneNr [1:*] is for Car [0:1.15:*];
 9. Year matches [4]
10.     constant {extract "\d{2}";
11.         context "\b'[4-9]\d\b";
12.         substitute "^" -> "19"
13.         ...
14. Mileage matches [8]
15.     ...
16.     keyword "\bmiles\b", "\bmi\.", "\bmi\b",
17.         "\bmileage\b";
18. ...
```

Figure 1: Car-Ads Application Ontology (Partial)

Price, *Feature*, and *PhoneNr* are the remaining lexical object sets in our car-ads application; *Car* is the only nonlexical object set.

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 (e.g. Line 10) in a data frame causes a string matching a regular expression to be extracted, but only if the **context** clause (e.g. Line 11) also matches the string and its surrounding characters. A **substitute** clause (e.g. Line 12) 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. One of the nonlexical object sets is designated as the *object set of interest*—*Car* for the car-ads ontology. The notation "[-> object]" in Line 1 designates the object set of interest.

We denote a relationship set by a name that includes its object-set names (e.g. *Car has Year* in Line 2 and *PhoneNr is for Car* in Line 8). The *min:max* pairs and *min:ave:max* triples in the relationship-set name 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 an object is expected to participate in the relationship set; and *max* designates the maximum number of times an object can participate, with * designating an unknown maximum number of times. The participation constraint on *Car* for *Car has Feature* in Line 6, for example, specifies that a car need not have any listed features, that a car has 2.1 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, which we use for illustration in this paper, we obtained participation constraints as follows. To make our constraints broadly representative, we selected ten 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 twelve 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 and normalized the values for a single car ad or obituary.

3 Recognition Heuristics

We are interested in determining whether a given document D is suitable for an application ontology O. In our document-recognition 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 document-recognition 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 [ECJ+99, EX00]. For H_3, we could first compute record boundaries [EJN99] and appropriately rearrange record values [EX00]. However, more sophisticated measures are more costly. We have chosen to experiment with less costly 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, 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 whether a multiple-record Web document D has the number of values expected for each lexical object set of 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 number of constants recognized for these object sets by O in D, we construct a document vector DV. We measure the relevance of D to O with respect to our expected-values heuristic by observing the cosine of the angle between DV and OV.

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 (i.e. the expected frequency of occurrence) 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. Since we do not in general know, indeed do not care, how many records we will find in documents given to us, we normalize these values for a single record. For example, 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 $>$, where these values are obtained as explained in Section 2. Thus, for a typical single car ad we would expect to almost always find a year, make, and model, but we only expect to find the mileage about 45% of the time, the price about 80% of the time. Further, we expect to see a list of features that on the average has a couple of items in it, and we expect to see a phone number and sometimes more than one phone number[3].

[3]It is easy to see that the variance might be useful, as well, but we found that the expected numbers were sufficient to get good results for the examples we tried.

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/.

Last update: Wednesday, December 22, 1999

Select a category

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.

Name of Lexical Object Set	Corresponding Lexical Values Found in the Document	Number of Lexical Values Found
Year	1989, 1998, 1994, 1999, '53, 1973, 1977, 95, 1996, ...	16
Make	Subaru, HONDA, Chevy, Olds, FORD, VW, Buick, Mercury, ...	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: Lexical values found in the multiple-record car advertisements in Figure 2(a).

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. Table 1 shows the values of the coefficients of the document vector for the car-ads document in Figure 2(a), and Table 2 shows the values of the coefficients of the document vector for the non-car-ads document in Figure 2(b). Observe that for document vectors we use the actual number of constants found in a document. To get the average (normalized for a single record), we would have to divide by the number of records—a number we do not know with certainty[4]. Therefore, we do not normalize, but instead merely compare the cosine of the angles between the vectors to get a measure for our expected-values heuristic.

We have discussed the creation of a document vector as if correctly detecting and classifying the lexical values in a document is easy—but it is not easy. We identify potential lexical values for an object set as explained in Section 2; this can be error-prone, but we can adjust the regular expressions to improve this initial identification and achieve good results [ECJ$^+$99]. After initial identification, we must decide 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 upstream ontology/document matching we use techniques that are far less sophisticated and thus also far less costly. In our simple upstream procedures we consider only, two cases: (1) a recognized string has no overlap either partially or completely

[4]We can estimate the number of records by dividing the length of the document vector by the length of the ontology vector. Indeed, we use this information downstream, but here we are still trying to determine whether the given document applies to the ontology. If it does, the length-division estimate for the number of records makes sense; otherwise, the estimate may be nonsense.

Name of Lexical Object Set	Corresponding Lexical Values Found in the Document	Number of Lexical Values Found
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, 218-281-5113, 800-532-7655, 281-1970, 800-406-5126, 281-1128	11

Table 2: Lexical values found in the multiple-record *Items for Sale* document in Figure 2(b).

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 the recognized string for an object set even if the sophisticated downstream processes would reject it. For Case 2, we resolve the overlap simplistically, as follows. There are three subcases: (1) exact match, (2) subsumption, and (3) partial overlap. (1) If a lexical value v is recognized as potentially belonging to more than one lexical object set, we use the closest keyword that appears before or after v to determine which object set to choose; if no applicable keyword is found, we choose one of the object sets arbitrarily. (2) If a lexical value v is a proper substring of lexical value w, we retain w and discard v. (3) If lexical value v and lexical value w appear in a Web document, such that a suffix of v is a prefix of w, we retain v and discard w.

As mentioned, 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, and N is the product of the lengths of the two vectors. When the distribution of values among the object sets in DV closely matches the expected distribution specified in OV, the angle θ will be close to zero, 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 expected frequency values of lexical object sets $Year$, $Make$, $Model$, $Mileage$, $Price$, $Feature$, and $PhoneNr$, respectively for a single ad 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 actual number of appearances of the lexical values in D_a. We thus compute $SIM(D_a, O)$ to be 0.9956. Now 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, but the coefficients of DV for D_b are 6, 0, 0, 2, 8, 0, and 11 (see the last column of Table 2). We thus compute $SIM(D_b, O)$ to be 0.5669.

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. We truncate the decimal value of the anticipated number to obtain the expected group size.

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), 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

Year: 2000 Year: 1989 Make: Subaru Model: SW -- Nr of Distinct "One Max" Object Sets: 3	Year: 1999 Year: 1998 Year: 1960 Mileage: 10000 -- Nr of Distinct "One Max" Object Sets: 2
Price: 1900 Year: 1998 Model: Elantra Year: 1994 -- Nr of Distinct "One Max" Object Sets: 3	Mileage: 401000 Year: 1940 Price: 17500 Price: 10971 -- Nr of Distinct "One Max" Object Sets: 3
Make: HONDA Model: ACCORD Mileage: 100000 Year: 1999 -- Nr of Distinct "One Max" Object Sets: 4	Year: 1950 Price: 27850 Price: 19990 Year: 1980 -- Nr of Distinct "One Max" Object Sets: 2
Model: GRAND AM Mileage: 27000 Price: 14000 Year: 1953 -- Nr of Distinct "One Max" Object Sets: 4	Price: 79850 Price: 42990 Price: 129650 Price: 78850 -- Nr of Distinct "One Max" Object Sets: 1
(a) First four groups of 1-max lexical values extracted from Figure 2(a).	(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.

in each group is four, G_{factor} of D_a is 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 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 the one in Figure 2(a), the heuristic-measure triple, which we derived in the previous three subsections, is (0.3056, 0.9956, 0.8653). For the Web document in Figure 2(b), the triple we derived is (0.0734, 0.5669, 0.5).

Since we did not know exactly how these three heuristics should be combined to best match application ontologies with documents, we decided to use machine learning. We did not know, for example, whether we should use all the heuristics or just one or two of them, and we did not know what threshold values to apply. Since the popular machine-learning algorithm C4.5 [Qui93] answers these questions, we decided to use it to combine the three heuristics into a single decision rule. 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 do binary classification by

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.

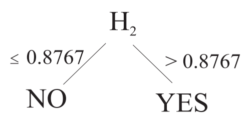
returning "YES" when a document is suitable, and returning "NO" otherwise). The performance measure is the percent of documents correctly classified when using a generated rule (i.e. the accuracy). The bias favors the shortest rule, so that if several rules are equally accurate, a decision tree with the fewest branches is chosen. The training data 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. 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 recognizing 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 recognized 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) to play the role of 10 negative car-ad examples.

Based on the 50 training examples for our 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

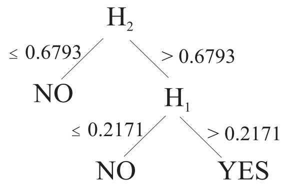


Thus, our document-recognition technique selects a document as a car ad if its expected-values measure is greater than 0.8767 (i.e. if the cosine between the car-ads ontology vector and the document vector is greater than 0.8767).

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

Rule 2

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

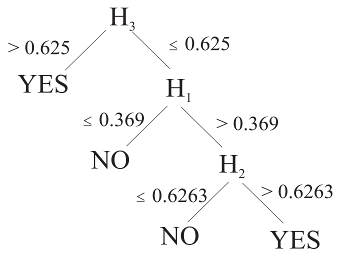


Thus, our document-recognition technique selects a document as an obituaries document only if its expected-values measure is greater then 0.6793 and its density measure is greater than 0.2171.

Searching for a potential universal rule over both 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:

Rule 3

$H_3 \leq 0.625$
| $H_1 \leq 0.369$: 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 document-recognition 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, 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. 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

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

Table 4: Negative examples in test sets.

Document	*Density Heuristic*	*Expected-Values Heuristic*	*Grouping Heuristic*	*Generated Recognition Rule*
1	0.3287	0.9345	0.8929	Yes
2	0.4614	0.7958	0.8029	No
3	0.2079	0.8855	0.8043	Yes
4	0.2153	0.9837	0.9167	Yes
5	0.3967	0.8881	0.8718	Yes
6	0.3032	0.9309	0.7824	Yes
7	0.1445	0.945	0.90625	Yes
8	0.3659	0.9658	0.85	Yes
9	0.2271	0.8867	0.7125	Yes
10	0.2382	0.9909	0.9231	Yes
11	0.1058	0.8782	0.7813	Yes

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

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 recognized the test set for both the car-ads application ontology and the obituary application ontology with the same F-measure, 95.3%. 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 F-measure for the Rule 3 was 91.3%, 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 number, 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 incorrectly (an incorrect negative response) and one negative example for obituaries that is judged

Document Subject	*Density Heuristic*	*Expected-Values Heuristic*	*Grouping Heuristic*	*Generated Recognition 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
Motorcycles	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.

incorrectly (an incorrect positive response).

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. In Section 4.2.4 we discuss our views on a universal rule versus application-ontology-dependent rules.

4.2.1 Incorrect Negative Response

Figure 4 displays the the car-ads document for which Rule 1 gives an incorrect negative response. 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 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

Document	*Density Heuristic*	*Expected-Values Heuristic*	*Grouping Heuristic*	*Generated Recognition Rule*
1	0.3622	0.7983	0.8571	Yes
2	0.4108	0.7542	0.56	Yes
3	0.2435	0.8106	0.7143	Yes
4	0.2458	0.7761	0.75	Yes
5	0.278	0.7926	0.7626	Yes
6	0.3864	0.8297	0.7589	Yes
7	0.3207	0.7084	0.8438	Yes
8	0.3112	0.7459	0.9531	Yes
9	0.5043	0.711	0.8125	Yes
10	0.3863	0.7466	0.8235	Yes

Table 7: Obituary rule test results: positive examples.

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 we have made in our document-recognition process. To recognize 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.[5]

4.2.2 Incorrect Positive Response

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 objects for interment and funeral places and times, and it extracts names it "thinks" are relative

[5]We have addressed this problem with good success in [EX00], but we assumed we knew the document was a multiple-record document applicable to the ontology. Based on some limited evidence, we could guess that a document is applicable and then iterate between adjustments and applicability measurements until we converge to "yes" or "no," but we have not yet tried this iterative approach.

Document Subject	*Density Heuristic*	*Expected-Values Heuristic*	*Grouping Heuristic*	*Generated Recognition 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
Motorcycles	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.

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 ignore low-confidence attributes or to give more weight to high-confidence attributes. These adjustments may be sufficient to properly judge the applicability of missing persons with respect to obituaries, but we have not yet tested these adjustments. As an alternative to these types of adjustments, if we also have an application ontology for missing people, we would see that the document better fits the missing-person ontology—we would thus reject it as a list of obituaries.

If someone considers antique car sales to be an incorrect positive response for car ads, we have no good suggestions on how to modify our document-recognition 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

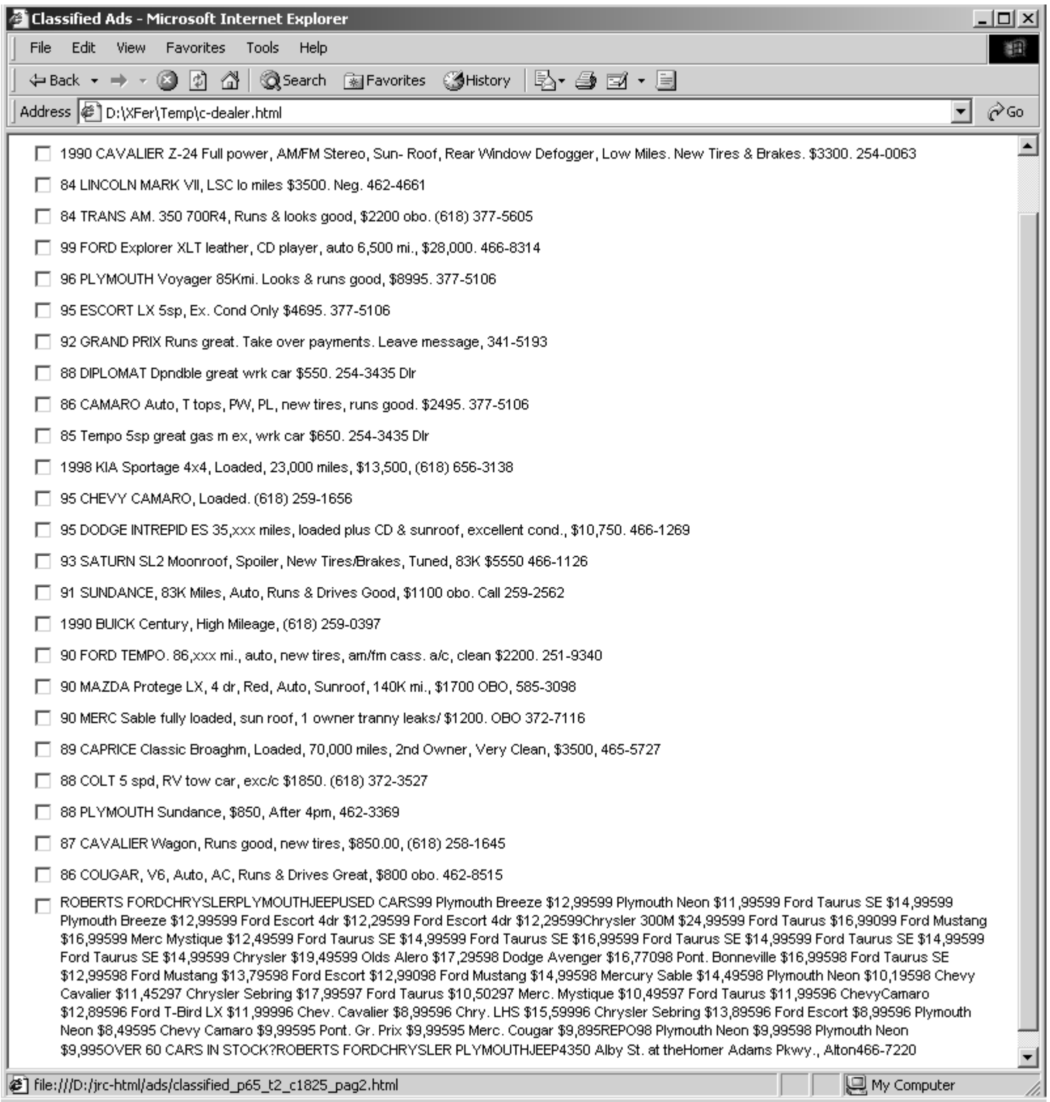


Figure 4: Car-ads document with incorrect negative response.

apply our ontology-applicability test to Web documents containing information for only a single record. Since singleton car ads are rare (if they exist at all), we only tested for obituaries. We selected nine "arbitrary" singleton obituaries plus an obituary for Princess Diana. 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. 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 incorrect negative responses. For the obituary of Princess Diana, the expected-values

Person Name	*Density Heuristic*	*Expected-Values Heuristic*	*Grouping Heuristic*	*Generated Recognition 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

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 F-measure and recall of this "universal rule" remain high, above 90%, but that the precision drops to 84%. Since this rule spans application ontologies, it 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 recognize suitable documents.

5 Concluding Remarks

We presented an approach for recognizing which multiple-record Web documents apply to an ontology. Once an application ontology is created, we can train a machine-learning algorithm over a triple of heuristics (density, expected-values, grouping) to produce a decision tree that accurately recognizes multiple-record documents for the ontology. Results for the tests we conducted showed that the F-measures were above 95% with recall and precision above 90% for both of our applications.

Acknowledgements: This material is based upon work supported by the National Science Foundation under grant No. IIS-0083127.

References

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

[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.

[BM98] L. Douglas Baker and Andrew K. McCallum. Distributional clustering of words for text classification. In *Proceedings of the 21th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 96–103, 1998.

[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.

[Cha99] S. Chakrabarti. Recent results in automatic Web resource discovery. *ACM Computing Surveys*, 31(4es), December 1999.

[CvdBD99] S. Chakrabarti, M. van den Berg, and B.E. Dom. Focused crawling: A new approach for topic-specific resource discovery. *Computer Networks*, 31:1623–1640, 1999.

[ECJ+99] D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y.-K. Ng, and R. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data and Knowledge Engineering*, 31(3):227–251, November 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.

[EX00] D.W. Embley and L. Xu. Record location and reconfiguration in unstructured multiple-record web documents. In *Proceedings of the Third International Workshop on the Web and Databases*, pages 123–128, Dallas, Texas, May 2000.

[HPS96] D.A. Hull, J.O. Pedersen, and H. Schütze. Method combination for document filtering. In *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 279–287, 1996.

[Joa96] T. Joachims. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. Technical Report Computer Science Technical Report CMU-CS-96-118, Carnigie-Mellon University, 1996.

[KS97] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the International Conference on Machine Learning*, pages 170–178, 1997.

[LSC96] D. Lewis, R. Schapire, and J. Callan. Training algorithms for linear text classifiers. In *Proceedings of the ACM SIGIR*, 1996.

[McC96] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[MG98] D. Mladenić and M. Grobelnik. Feature selection for classification based on text hierarchy. In *Proceedings of Automated Learning and Discovery, CONALD-98*, 1998.

[MLW92] B. Masand, G. Linoff, and D. Waltz. Classifying news stories using memory based reasoning. In *Proceedings of 15th Annual ACM/SIGIR Conference*, pages 59–65, 1992.

[MN99] A. McCallum and K. Nigam. Text classification by bootstrapping with keywords, em and shrinkage. In *Proceedings of the ACL'99 Workshop for Unsupervised Learning in Natural Language Processing*, University of Maryland, June 1999.

[MNRS99] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Building domain-specific search engines with machine learning techniques. In *Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace*, Stanford University, March 1999.

[MNRS00] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[MRM98] A. McCallum, R. Rosenfeld, and T. Mitchell. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of 15th International Conference on Machine Learning (ICML98)*, 1998.

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

[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, 1994.

[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.

[TPL95] M. Tresch, N. Palmer, and A. Luniewski. Type classification of semi-structured documents. In *Proceedings of the 21st International Conference on Very Large Data Bases (VLDB'95)*, pages 263–274, Zürich, Switzerland, 1995.

[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.

[WPW95] E. Weiner, J. Pedersen, and A. Weigend. A neural network approach to topic spotting. In *Proceedings of the SDAIR*, 1995.