

Combining Classifiers to Identify Online Databases

Luciano Barbosa
School of Computing
University of Utah

lbarbosa@cs.utah.edu

Juliana Freire
School of Computing
University of Utah

juliana@cs.utah.edu

ABSTRACT

We address the problem of identifying the domain of online databases. More precisely, given a set F of Web forms automatically gathered by a focused crawler and an online database domain D , our goal is to select from F only the forms that are entry points to databases in D . Having a set of Web forms that serve as entry points to similar online databases is a requirement for many applications and techniques that aim to extract and integrate hidden-Web information, such as meta-searchers, online database directories, hidden-Web crawlers, and form-schema matching and merging.

We propose a new strategy that automatically and accurately classifies online databases based on features that can be easily extracted from Web forms. By judiciously partitioning the space of form features, this strategy allows the use of simpler classifiers that can be constructed using learning techniques that are better suited for the features of each partition. Experiments using real Web data in a representative set of domains show that the use of different classifiers leads to high accuracy, precision and recall. This indicates that our modular classifier composition provides an effective and scalable solution for classifying online databases.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Selection process.

General Terms

Algorithms, Design, Experimentation.

Keywords

Hidden Web, learning classifiers, hierarchical classifiers, online database directories, Web crawlers.

1. INTRODUCTION

Due to the explosion in the number of online databases, there has been increased interest in leveraging the high-quality information present in these databases [2, 3, 11, 23, 33]. However, finding the right databases can be very challenging. For example, if a biologist needs to locate databases related to molecular biology and searches on Google for the keywords “molecular biology database” over 27 million documents are returned. Among these, she will find pages that contain databases, but the results also include a very large

number of pages from journals, scientific articles, personal Web pages, etc.

Recognizing the need for better mechanisms to locate online databases, people have started to create online database collections such as the Molecular Biology Database Collection [15], which lists databases of value to biologists. This collection, has been manually created and is maintained by the National Library of Medicine. Since there are several million online databases [23], manual approaches to this problem are not practical. Besides, since new databases are constantly being added, the freshness of a manually maintained collection is greatly compromised.

In this paper, we describe a new approach to the problem of identifying online databases that belong to a given domain. There are a number of issues that make this problem particularly challenging. Since online databases are sparsely distributed on the Web, an efficient strategy is needed to locate the forms that serve as entry points to these databases. In addition, online databases do not publish their schemas and their contents are hard to retrieve. Thus, a scalable solution must determine the relevance of a form to a given database domain by examining information that can be automatically extracted from the forms and in their vicinity.

Web crawlers can be used to locate online databases [3, 9, 10, 13, 26, 29]. However, even a focused crawler invariably retrieves a diverse set of forms. Consider for example, the Form-Focused Crawler (FFC) [3] which is optimized for locating searchable Web forms. For a set of representative database domains, on average, only 16% of the forms retrieved by the FFC are actually relevant—for some domains this percentage can be as low as 6.5%. These numbers are even lower for less focused crawlers, e.g., crawlers that focus only on a topic [9, 10, 13]. The problem is that a focus topic (or concept) may encompass pages that contain many different database domains. For example, while crawling to find airfare search interfaces the FFC also retrieves a large number of forms for rental car and hotel reservation, since these are often co-located with airfare search interfaces in travel sites. The set of retrieved forms also includes many non-searchable forms that do not represent database queries such as forms for login, mailing list subscriptions, and Web-based email forms.

Having a homogeneous set of forms that lead to databases in the same domain is useful, and sometimes required, for a number of applications. For example, whereas for constructing online database directories, such as BrightPlanet [8] and the Molecular Biology Database Collection [15], it is desirable that only relevant databases are listed, the effectiveness

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

(a) Job search forms

(b) Hotel and Airfare search forms

Figure 1: Variability in Web Forms. (a) shows forms in the Job domain that use different attribute names to represent the same concepts. (b) shows forms in two distinct domains, Hotel and Airfare, which contain attributes with similar labels.

statistical schema matching across Web form interfaces [18] can be greatly diminished if the set of input forms is noisy and contains forms from multiple domains.

Identifying relevant forms as belonging to a given database domain, however, is a hard problem that has started to receive attention in recent literature. Previous works on form classification can be broadly characterized as *pre-query* and *post-query* [27]. Post-query techniques issue probe queries and the retrieved results (i.e., the database contents) are used for classification purposes. Probing-based techniques [7, 17] are effective and required for simple, keyword-based interfaces, which are easy to fill out automatically and have little or no information pertinent to the underlying database (e.g., a form with a single attribute labeled “search”). However, these techniques cannot be easily adapted to (structured) multi-attribute interfaces. The difficulties in automatically filling out structured Web forms have been documented in the literature [7, 11, 25, 32]. To help automatically fill out multi-attribute forms, paradoxically, it is often necessary to first discover and organize forms in the domain, so that attribute correspondences can be found and possible values for attributes collected (e.g., through matching a large number of forms in a domain [18]).

Pre-query techniques, on the other hand, rely only on visible features of forms. They are suitable for forms whose contents are indicative of the database domain. An important requirement for these techniques is the ability to deal with the wide variation in content and structure of automatically gathered forms. This variability is present even in well-defined and narrow domains. As Figure 1 illustrates, different attribute names can be used to represent the same concepts in forms that belong to a domain, and forms may be similar even when they belong to different domains. There can also be high similarity between searchable and non-searchable forms (see Figure 2). Previously proposed techniques, however, have limitations with respect to scalability [22], and the ability to deal with highly heterogeneous

Figure 2: Searchable and a non-searchable form with similar contents. The form on the left is used for searching over a database of used cars, whereas the one on the right is used to request quotes.

form sets [12]. The technique proposed by Hess and Kushmerick [22] classifies forms based on form attribute labels. Thus, the effectiveness of their technique depends on the ability to extract descriptive labels for form attributes—a task that is hard to automate [19, 25]. Cope et al. [12], on the other hand, use a classifier that considers only a subset of the form contents: the form structure and the content inside tags in the form context (e.g., the text inside an `input` element). Consequently, their classifier is unable to correctly identify forms whose content are indicative of the database domain, but occur outside these tags.

Contributions and Outline. In this paper, we describe a new pre-query method for automatically classifying forms with respect to a database domain that is both *scalable and accurate*. As discussed in Section 2, we model the problem of identifying the domain of an online database as a search problem over features of Web forms that serve as entry points to these databases. We propose **H**ierarchical **F**orm **I**dentification (HIFI), a new strategy that obtains high accuracy by partitioning the space of form features and using learning classifiers that are best suited for the features in each partition. The modular classifiers are presented in Sections 3 and 4, and in Section 5, we discuss how they are combined. Because our approach relies only of features that are automatically extracted from forms, it is scalable. An extensive experimental evaluation of HIFI, using real Web data consisting of over 27,000 forms in eight distinct database domains, is discussed in Section 6. The results confirm that our choice of partitioning the form features and combining different classifiers leads to higher accuracy than if a single classifier is used for all features. Most importantly, the high precision and recall obtained indicate that it is possible to automatically and accurately classify online databases using visible information readily available in form interfaces. This makes our strategy especially suitable for large-scale Web information integration tasks.

2. SOLUTION OVERVIEW

Our goal is to select only the *relevant* forms in set of heterogeneous forms retrieved by a focused crawler. The problem we are trying to solve can be stated as follows: *Given a set F of Web forms automatically gathered by a focused crawler, and an online database domain D , our goal*

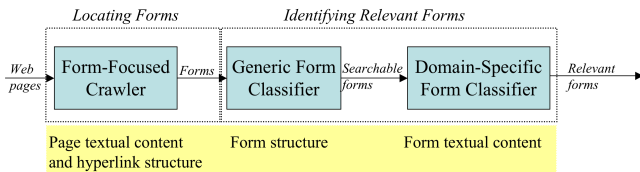


Figure 3: HIFI Architecture.

is to select from F only the forms that are entry points to databases in D .

In other words, we would like to filter out all *irrelevant forms*—non-searchable forms, and searchable forms that do not belong to the domain D . *Searchable forms* are forms which serve as entry points to online databases—once submitted, a query is issued against the database and its results are returned. *Non-searchable forms*, in contrast, are often used to just to submit information.

Our approach to locate and identify online databases consists of three components: a focused crawler; the generic form classifier (GFC); and the domain-specific form classifier (DSFC). Figure 3 shows how the different components of our solution explore and prune the space of Web forms. The focused crawler uses features of Web pages to focus on a topic. It helps prune the search space by avoiding a large number of irrelevant forms that reside in off-topic pages. Nonetheless, as we discussed above, within a focus topic there can be both non-searchable forms as well as searchable forms from multiple database domains. The two classifiers are used in a sequence to identify relevant forms in the input set: the first classifier eliminates non-searchable forms based on structural characteristics; and the second uses the textual contents of forms to identify, among searchable forms, the ones that belong to the target database domain.

Learning to Classify Forms. Since our goal is to devise a general solution to this problem, that works across different domains, we formalize the problem of identifying relevant forms in a particular database domain in terms of inductive learning concepts [24]. One of the basic ideas in inductive inference is that there exists a target concept C that can be modeled by an unknown function f . Given an instance x , $f(x)$ indicates whether x belongs to C . Thus, the task of induction is, given a collection of examples of f , to determine a function h , called hypothesis or classifier, that approximates f .

There can be several different functions that approximate f whose hypotheses are consistent with the examples. An important question is then how to choose from among these hypotheses. For the problem defined above, one possible solution would be to gather examples of forms in the domain D and construct a classifier based on these examples (the hypothesis h). However, given the wide variation in form structure and content within and across different domains, the resulting hypotheses can be very complex.

A simpler hypothesis set can be obtained by hierarchically decomposing the feature space. In HIFI, this decomposition follows the hierarchy of form types. The most general concept (GC), which sits at the top of the hierarchy, corresponds to searchable forms which may belong to multiple database domains. The more specific concept (SC) corresponds to forms in a specific database domain. The original hypothesis h is replaced by h_{GC} and h_{SC} , where h_{GC} identifies the searchable forms over the set of all Web forms and

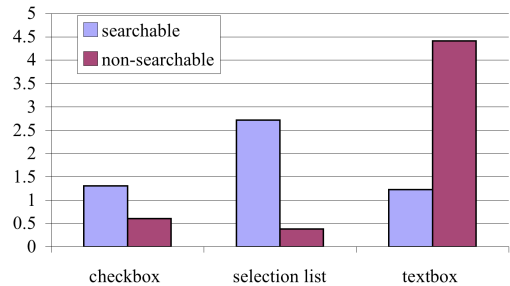


Figure 4: The differences between the average number of checkboxes, selection lists and textboxes of searchable and non-searchable forms illustrate some the structural differences between these two kinds of forms.

h_{SC} identifies, among searchable forms, forms that are relevant to a given domain.

In essence, by creating these different levels of abstraction, instead of using a single complex classifier, we can construct two classifiers that make simpler decisions: h_{GC} is performed by the *Generic Form Classifier (GFC)* and h_{SC} by the *Domain-Specific Form Classifier (DSFC)*. In addition, the decomposition of the feature space allows the use of learning techniques that are more appropriate for each feature subset. As we discuss in Sections 3 and 4, an evaluation of different learning techniques for these classifiers shows that decision trees [24] present the lowest error rates for determining whether a form is searchable based on structural patterns, whereas SVM [24] is the most effective technique to identify forms that belong to the given database domain based on their textual content. More details about these classifiers, how they are constructed and combined are given in Sections 3, 4 and 5.

3. USING STRUCTURE TO IDENTIFY SEARCHABLE FORMS

As a first step in the form-filtering process we apply the GFC to identify searchable forms. Empirically, we have observed that some structural characteristics of a form can be a good indicator as to whether the form is searchable or not [3]. See for example, Figure 4, which shows the average number of selection lists, textboxes and checkboxes in a sample set that contains both searchable and non-searchable forms (details about this sample are given below). This figure suggests that searchable forms have a higher number of selection lists and checkboxes, whereas non-searchable forms have a higher number of textboxes. The forms in Figure 2 are concrete examples of this trend. The form on the left is searchable and contains several selection lists, whereas the form on the right is non-searchable and contains a large number of textboxes.

Other structural features of forms that are useful in differentiating searchable and non-searchable forms include: number of hidden tags; number of radio tags; number of file inputs; number of submit tags; number of image inputs; number of buttons; number of resets; number of password tags; number of textboxes; number of items in selection lists; sum of text sizes in textboxes; submission method (post or get). Another useful feature is the presence of the string “search” within the form and submit tags. In fact, the presence of the “search” string within the form and submit tags

Algorithm	Error test
Naïve Bayes	24%
MultiLayer Perceptron	12.8%
C4.5	9.05%
SVM (degree=1)	14.7%
SVM (degree=2)	16.2%
SVM (degree=3)	14.9%
SVM (degree=4)	14.9%
SVM (degree=5)	15.1%

Table 1: Error test rates for GFC.

was the feature which obtained the highest entropy in our feature set. Note that all of these features can be automatically extracted from Web forms—they require no manual pre-processing.

We built classifiers for these features using different machine learning techniques: Naïve Bayes, Decision tree (C4.5 algorithm), MultiLayer Perceptron and Support Vector Machine with distinct polynomial kernel degrees (SMO algorithm).¹ For positive examples we extracted 216 searchable forms from the UIUC repository [30], and we manually gathered 259 non-searchable forms for the negative examples. We set up the training phase with 10-fold cross validation and split the form set in two thirds to the training set and one third to the testing set. The error test rates for the different techniques are shown in Table 1. These low error rates indicate that structural features are very effective to differentiate between searchable and non-searchable forms. For HIFI, we selected the C4.5 classifier because it had the lowest error rate.

It is worthy of note that Cope et al. [12] also proposed the use of decision trees to classify searchable forms. Similar to the GFC, their classifier uses features that can be automatically extracted from forms. However, because their strategy also takes the textual contents inside the form tags into account, it is domain-specific. As a result, a different classifier needs to be constructed for each domain. In contrast, as we discuss in Section 6, the GFC is very effective in identifying searchable forms in different domains. Because the GFC is domain independent, it can be re-used, as is, in many applications. For example, it can be used to pre-process input forms for form clustering algorithms [4, 19], or to improve the quality of automatically constructed online database directories such as Complete Planet [8].

4. FORM DOMAIN IDENTIFICATION AS TEXT CLASSIFICATION

The GFC is effective for identifying searchable forms, regardless of their domains. However, as we discussed in Section 1, even when a focused crawler is used, the set of forms retrieved may include searchable forms from many different domains. Consider, for instance the forms in Figure 5. These two forms were retrieved by the Form-Focused Crawler (FFC) during a crawl to find airfare search forms, but neither belongs to the target Airfare database domain—one is a hotel search form and the other is a rental car search form.

To identify searchable forms that belong to a given domain, as the second step of our form-filtering process, we use a more specialized classifier, the DSFC. The DSFC uses the textual content of a form to determine its domain. The form content is often a good indicator of the database domain—it

¹We used WEKA [31] to construct these classifiers as well as for constructing the classifiers described in Section 4.

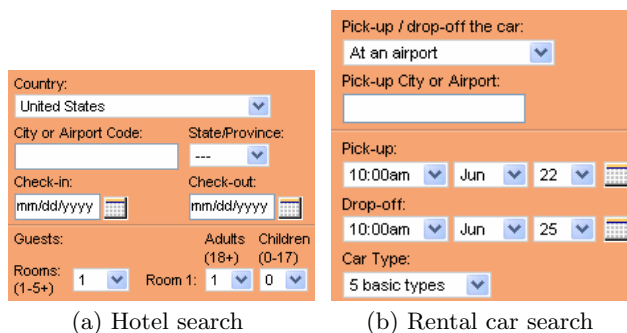


Figure 5: Searchable forms obtained among Airfare-related Web pages.

contains metadata and data that pertain to the database. For example, form attribute names often match the names of fields in the database, and selection lists often contain values that are present in the database.

Other works have used the form contents for both classification and clustering purposes [19, 22]. He et al. [19] used form contents, more precisely the attribute labels, to cluster forms. They note that forms in a given domain contain a well-defined and restricted vocabulary. Hess and Kushmerick use both the attribute labels and their values (if present) to classify forms. However, the effectiveness of approaches that rely on attribute labels is highly dependent on the ability to extract descriptive labels for form attributes, and this task is hard to automate [19, 25]. It is worthy of note that the experiments reported in both [19] and [22] relied on a manual pre-processing step for extracting the form attribute labels.

To construct a scalable solution that is able to automatically classify thousands of forms, instead of attempting to extract attribute labels, the DSFC uses the textual content within a form—the text enclosed by the `form` tags after the HTML markup is removed. Since, in essence, the DSFC needs to perform text classification, we experimented with two learning techniques that have been found to be effective for this task: decision trees (C4.5 algorithm) and SVMs (the SMO algorithm).

We evaluated the effectiveness of these techniques over eight distinct domains.² For each domain we built a DSFC instance as follows. First, we performed a focused crawl (using the FFC [3]) and collected all distinct forms in the crawled pages. Forms with the same structure and located in the same host are considered the same—even if they appear in different pages. Then, we used the GFC to filter out the non-searchable forms. From this set, we extracted the structured forms and manually selected positive (on average 220 per domain) and negative examples (on average 220 per domain). After the HTML markup is removed, each form is represented as a vector where each cell represents a stemmed word [1], and the cell value is the frequency of the corresponding term in the form.

These vectors are then used as input to the DSFC instance. We set up the training phase with 10-fold cross validation and split the form set in two thirds to the training set and one third to the testing set.

Table 2 shows, for the eight domains, the error rates obtained by decision trees and by SVMs using distinct poly-

²These domains are described in Section 6.

Domain	Decision tree	SVM(d=1)	SVM(d=2)	SVM(d=3)	SVM(d=4)	SVM(d=5)
Airfare	11.03%	7.1%	9.7%	12.3%	16.2%	18.1%
Auto	4.3%	3.5%	10%	10.7%	13.6%	14.3%
Book	11.5%	7.8%	7.2%	6.6%	22.4%	35.1%
Hotel	12.6%	13.4%	10%	11.7%	13.4%	28.5%
Job	8.9%	8%	9.8%	10.7%	14.2%	18.75%
Movie	12.6%	8.2%	9.7%	15.6%	17.9%	25.4%
Music	12.2%	11.4%	10.6%	14.7%	22.1%	25.4%
Rental	10.9%	5%	6.7%	6.7%	8.4%	8.4%

Table 2: Error test rates of DSFC in different domains.

mial kernel degrees. The low error rates reinforce our choice to use the form contents as the basis for classification. Note that SVMs perform better than decision trees. However, no single polynomial kernel degree is uniformly better for all domains. For the experimental evaluation described in Section 6, for each domain, we selected the classifier with the lowest error rate.

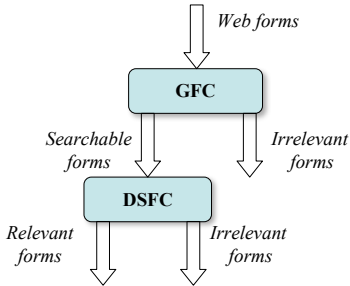


Figure 6: Hierarchical Composition of Modular Classifiers

5. COMBINING CLASSIFIERS

A popular technique to combine classifiers is to compose them in an ensemble. A set of base classifiers is constructed, and classification of new examples is decided by combining individual decisions from the base classifiers (see e.g., [5]). In contrast, HIFI partitions the feature space and applies different classifiers to the different partitions in a sequence. As illustrated in Figure 6, the domain-independent GFC first performs a coarse classification and prunes a large number of irrelevant (non-searchable) forms. The DSFC then works on the smaller set of searchable forms and identifies among them the relevant forms.

The hierarchical composition of classifiers leads to modularity: a complex problem is decomposed into simpler sub-components and a monolithic classifier is replaced by a hierarchy of classifiers, each dedicated to a subset of the hypothesis [16, 21]. This has several benefits. First and foremost, because the learning task of the DSFC is simplified, the overall classification process is more accurate and robust. The DSFC need not consider hypotheses that deal with the high variability present in non-searchable forms, and this simpler hypothesis set leads to improved classification accuracy. As an example, consider Figure 2, which shows two forms with similar content. Whereas the form on the left is a (relevant) searchable form in the Auto domain, the one on the right is a non-searchable form (irrelevant) for requesting car quotes.

There are also instances of badly designed Web pages in which the form content contains extraneous terms. Since our form extraction process is automated and all non-HTML terms within the form are considered, a non-searchable form may be incorrectly classified if these extraneous terms are representative of the database domain. Because the GFC

takes structural features into account, it is able to accurately prune many irrelevant forms which might otherwise be misclassified as relevant by the DSFC.

Another important benefit of the hierarchical composition of classifiers is that we can apply to each partition a learning technique that is best suited for the feature set of the partition: decision trees lead to the lowest error rates when applied to structural features of forms, whereas for the textual contents, SVMs are the most effective.

6. EXPERIMENTAL EVALUATION

In this section, we first assess the effectiveness of the hierarchical composition of the GFC and DSFC using forms retrieved by the FFC in a set of representative database domains. Then, to evaluate the sensitivity of our approach to the quality of the input set of forms, we study the performance of HIFI using forms retrieved by two crawlers that use distinct focus strategies: the FFC and the best-first focused crawler [10].

6.1 Experimental Setup

Database Domains. In order to evaluate our solution and assess its generality, we selected domains with different characteristics. The database domains used in our experiments are shown in Table 3. These domains present high variability in the size, structure, and vocabulary of their forms. As Table 4 shows, there are striking differences in form structure across the domains. While Airfare, Hotel and Rental have a relatively large number of hidden inputs and selection lists, Music and Movie have very few of these types of fields. In fact, Music and Movie have a much lower total number of attributes than the other domains.

The textual content of the forms in the different domains also have different characteristics. As Table 3 shows, there is a wide variation in the average number of terms for forms in the different database domains. Whereas domains such as Auto and Music have relatively small forms (with an average of 52 and 82 terms, respectively), others such as Airfare and Job have fairly large forms (with an average of 172 and 165 terms, respectively).

These domains also present a wide variation in their vocabulary. Figure 7 shows the Simpson Index [28] for the different domains. The value of this index represents the probability that two words selected at random from distinct forms in a domain are the same word—thus, the higher the value, the more homogeneous the vocabulary is.³ The figure clearly shows that there is a significant variation in vocabulary homogeneity for the different domains. For example, the vocabulary of Auto is substantially more homogeneous than that of Movie.

³The Simpson Index provides a better measure for vocabulary diversity than other measures such as vocabulary size.

Domain	Description	# of Forms	Avg Form Size
Airfare	airfare search	4729	172
Auto	new and used cars	1930	52
Book	books for sale	1672	96
Hotel	hotel availability	10228	139
Job	job search	1674	165
Movie	movie titles and DVDs	2313	115
Music	music titles and CDs	1129	82
Rental	car rental availability	2702	137

Table 3: Database domains used in experiments. The table shows for each domain, the total number of structured forms and the average number of terms in the relevant forms.

Domain	Hidden	Checkbox	Selection list	Textbox
Airfare	8.3	0.4	9.9	2.5
Auto	2.2	0.4	3.4	1.3
Book	1.7	0.9	2.0	3.3
Hotel	6.7	1.2	5.8	0.9
Job	1.7	1.2	3.2	1.7
Movie	1.4	0.2	0.8	1.1
Music	1.4	0.2	1.1	1.3
Rental	4.1	0.2	6.3	0.7

Table 4: Variability in form structure. The table shows the average number of different form attribute types in each database domain.

Performance Metrics. To evaluate the performance of the classifiers, we use a confusion matrix which represents the relationship between actual classification and predicted classification:

	Predicted positive	Predicted negative
Actual positive	True Positive (TP)	False Negative(FN)
Actual negative	False Positive(FP)	True Negative(TN)

A commonly used measure of the performance of a classifier is accuracy:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Accuracy is a suitable measure when the input to the classifier contains similar proportions of positive and negatives examples. Consider, for example, an input set consisting of 1 positive example and 9 negative examples. If a classifier labels all items as negative, its accuracy is 90%. Since during a Web crawl, the large majority of forms retrieved are irrelevant—i.e., the forms match the negative examples—we use three other measures of performance which better reflect the effectiveness of a classifier over the set of relevant forms:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \quad (5)$$

Recall shows the number of relevant items retrieved as fraction of all relevant items; *precision* represents the number of

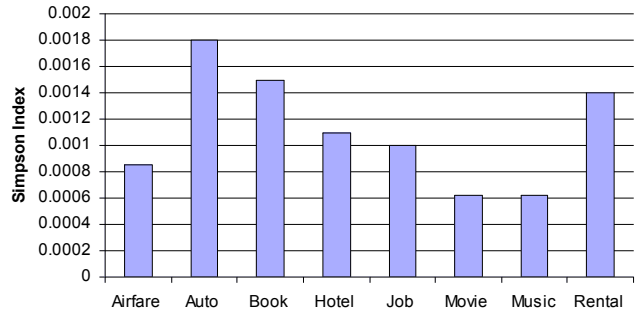


Figure 7: Variability in the homogeneity of form vocabulary across domains.

relevant items as a fraction all the items predicted as positive by the classifier; and *specificity* is the proportion of actual irrelevant items predicted as irrelevant. The F-measure is the harmonic mean between precision and recall. A high F-measure means that both recall and precision have high values—a perfect classification would result in an F-measure with value equal 1.

6.2 Effectiveness of HIFI

The effectiveness of our approach depends on how the classifiers used for each partition of the search space interact with each other. To measure the performance of the combination of GFC+DSFC, we performed the following experiment. First, we ran the FFC for the eight database domains. Among all the structured and distinct forms gathered by the crawler, we executed the form-filtering process.

As described in Section 3, the GFC splits the form space into searchable forms (SF) and non-searchable forms (NSF); and the DSFC classifier further splits the space of searchable forms into predicted relevant forms (PRF) and predicted irrelevant forms (PNRF). Since for each domain, the FFC retrieves from hundreds to thousands of forms (see Table 3), we randomly selected 20% of the forms in each one of these sets and manually inspected them to verify whether they were correctly classified.

As the GFC prunes the search space of forms that serve as input to the DSFC, we are interested in (1) evaluating its effectiveness in removing the subset of irrelevant forms that are non-searchable (specificity, Equation 4); and (2) verifying whether it misclassifies relevant forms (recall, Equation 2). The specificity and recall values of GFC in the eight domains are given in Table 5. Note that these values were measured taking into account only the relevant forms (a subset of searchable forms) and not all searchable forms, which are the target of the GFC. These numbers show that *GFC effectively partitions the search space*: it removes a significant percentage of irrelevant forms, which are non-searchable (high specificity), and misclassifies only a few relevant forms (high recall). The domains in which the GFC obtained the lowest specificity values were Airfare, Hotel and Rental. Since the GFC is domain independent, and forms in these domains are often co-located in the same sites (e.g., Orbitz and Travelocity), a large percentage of the forms it classifies as searchable belong to a domain other than the target database domain. As we discuss below, because the DSFC takes the textual content of forms into account, when applied to the set of forms returned by the GFC, the DSFC is able to filter out forms that do not belong to the target database domain with high accuracy.

Domain	Recall	Specificity
Airfare	0.97	0.18
Auto	0.90	0.58
Book	0.92	0.78
Hotel	0.98	0.30
Job	0.97	0.53
Movie	0.95	0.57
Music	0.98	0.41
Rental	0.96	0.34

Table 5: Effectiveness of GFC in identifying relevant forms.

Domain	Recall	Precision	Accuracy
Airfare	0.91	0.91	0.98
Auto	0.87	0.87	0.93
Book	0.90	0.92	0.96
Hotel	0.96	0.97	0.95
Job	0.87	0.95	0.95
Movie	0.75	0.80	0.99
Music	0.73	0.88	0.89
Rental	0.94	0.91	0.97

Table 6: Effectiveness of classifier composition.

	Recall	Precision
Configuration 1	0.94	0.72
Configuration 2	0.49	0.94
HIFI	0.87	0.95

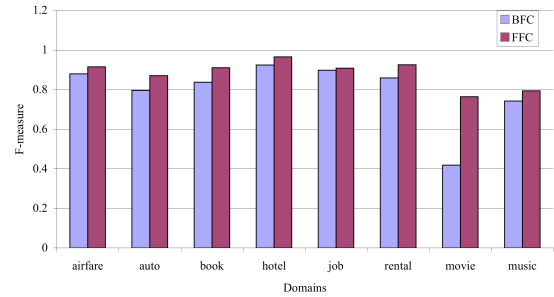
Table 7: Effectiveness of two configurations that use monolithic classifiers.

As expected, the specificity of GFC for searchable forms is much higher than for relevant forms. For all domains, specificity values were above 90%. This confirms the ability of the GFC to accurately identify searchable forms.

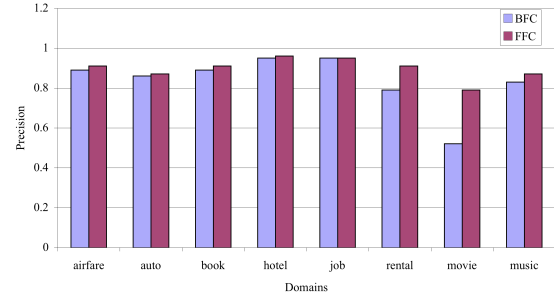
HIFI = GFC + DSFC. Table 6 shows the recall, precision and accuracy of the form-filtering process which combines the GFC and the DSFC. The overall accuracy obtained by HIFI is high in all domains. As discussed above, since a large percentage of the input forms is irrelevant, it is important to also examine the recall and precision. For most domains both recall and precision are high—with precision varying from 0.8 to 0.97 and recall varying from 0.73 to 0.96. Note that these results were obtained for domains with very different characteristics. For example, while the Auto domain has small forms and a relatively homogeneous vocabulary, Airfare has big and very heterogeneous forms (see Table 3 and Figure 7). This indicates that our approach provides an effective and general mechanism to automatically classify online databases based on their corresponding forms.

Classifier Composition vs. Monolithic Classifier. To verify whether the combination of classifiers is more effective than a monolithic classifier, we measured the recall and precision of two configurations of monolithic classifiers. Configuration 1 uses only the DSFC which is trained as described in Section 4, but executed over the entire input set provided by the focused crawler (not just over searchable forms). For Configuration 2, we built a new classifier which combines the structural features of the GFC and the form contents. This new classifier was trained using both searchable and non-searchable forms, and it was executed over all the forms returned by the crawler. We note that, for both configurations, SVM with polynomial degree 1 obtained the highest accuracy. Table 7 shows the recall and precision for these two configurations for the Job domain—a similar behavior was observed in other domains.

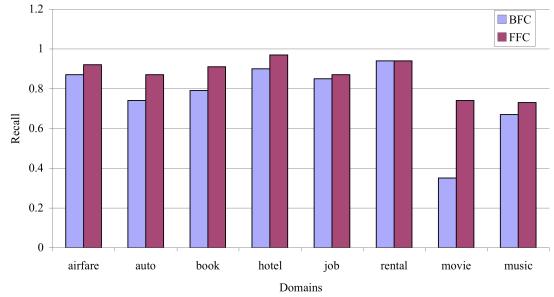
For Configuration 1, the classifier precision (0.72) is lower than the combination of classifiers (0.95). Since this classi-



(a) F-measure



(b) Precision



(c) Recall

Figure 8: Performance of the HIFI classification strategy using forms retrieved by two different crawlers: the best-first crawler (BFC) and the form-focused crawler (FFC).

fier is trained only with searchable forms, it performs poorly over the non-searchable forms. Because the classifier for Configuration 2 learns features of both searchable and non-searchable forms, its model is more specific. Although this more specific model leads to a higher precision (0.94) over the entire input set, it also misclassifies a large number of relevant forms, as one can be seen from the low recall (0.49). These results reinforce our decision to decompose the search space, and to use classifiers trained with different sets of features.

6.3 Sensitivity to Input Quality

The results presented in the previous sections were obtained using forms retrieved by the FFC. To assess the sensitivity of HIFI to the quality of the input forms, we evaluate the classifier composition using forms gathered by a different focused crawler: the best-first crawler proposed by Chakrabarti et al. [10]. The best-first crawler (BFC) uses a classifier that learns to classify pages as belonging to topics in a taxonomy. During a search, the crawler only follows links from pages classified as being on-topic. The FFC’s search is more focused than that of the BFC. Like the BFC,

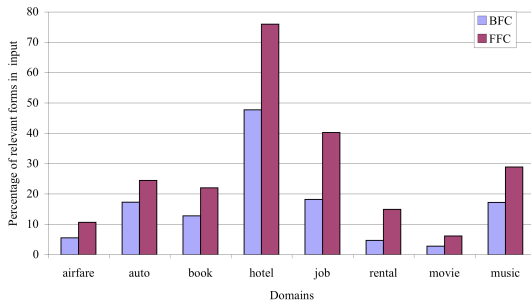


Figure 9: Percentage of relevant forms among the forms retrieved by the BFC and the FFC.

it uses pages’ contents to focus the crawl on a topic, but instead of following all links in a page classified as being on-topic, the FFC prioritizes links that are more likely to lead to pages that contain searchable forms.

Figure 8 shows the F-measure, recall and precision obtained by HIFI using forms gathered by the two crawlers. HIFI performed consistently better using the higher-quality inputs from the FFC. To give some insight about these results, Figure 9 shows the percentage of relevant forms in the input provided by the two crawlers. In all domains, the percentage of relevant forms retrieved by the FFC is larger than that of the BFC. Furthermore, the FFC retrieves between 40% and 220% more relevant forms than the BFC. Since F-measure, precision and recall are directly proportional to the number of true positives (Equations 3, 2, and 5), it is not surprising that better performance is obtained using the higher-quality input produced by the FFC, that contains both a larger number and a larger percentage of relevant forms.

Note that high values for F-measure, recall and precision were also obtained using the forms obtained by the BFC. The only exception is the Movie domain. A possible explanation is that the vocabulary of Movie is very heterogeneous, as one can be seen from its Simpson index (Figure 7). As a result, the DSFC for Movies is less accurate than for the other domains. This is reflected in Figure 8, where the lowest F-measure values are for the Movie domain. The problem is compounded due to the sparseness of this domain: only 2.7% of the forms retrieved by the BFC were relevant—a total of 102 relevant forms.

The Music domain, similar to Movie, has a low Simpson index and its DSFC also has lower accuracy. However, Music is much less sparse than Movie. Around 17% of the forms retrieved by the BFC were relevant—a total of 284 relevant forms. This explains the relatively higher F-measure for HIFI using BFC in the Music domain. Rental, on the other hand, is very sparse. BFC retrieves only 96 relevant forms, 4.6% of all forms it locates. But the vocabulary for Rental is much more homogeneous than the vocabulary for Movie—its Simpson index is over twice that of Movie.

7. RELATED WORK

Our work is closely related to pre-query approaches to form classification. Hess and Kushmerick [22] use a stochastic generative model of a Web service designer creating a form to host a service (e.g., a query interface to an online database), and learn to identify forms in a domain. Although their results are promising, their solution requires that forms be manually pre-processed to remove irrelevant

attributes. In contrast, in our solution, the extraction of form features is completely automated.

Similar to the GFC, Cope et al. [12] try to identify searchable forms [24]. Although they reported high precision and recall for two testbeds (academic and random Web sites), because they construct decision trees based both on structural features and the textual content inside form tags, their approach is domain-dependent, requiring the construction of specialized classifiers for different form sets. In contrast, the GFC is domain-independent—it is able to accurately classify forms in different domains. Although Cope’s classifier has a domain-specific component, unlike the DSFC it considers only a subset of the form contents: the form structure and the contents inside the HTML tags. Thus, its effectiveness is compromised for forms where the content representative of the database domain lies outside the HTML tags (e.g., descriptive labels for text fields).

The problem of classifying online databases based on the contents of the forms that serve as entry points to these databases can be seen as an instance of the problem of text classification. Although there is a rich literature on combining text classifiers to obtain higher classification accuracy, much of this work has centered around policies for selecting the most appropriate classifier or on strategies to combine the output of the individual classifiers (see [6] for an overview). In contrast to prior research in combining text classifiers, to obtain higher accuracy, we partition the feature space and combine classifiers in a hierarchical fashion—a distinct classifier is applied to each partition. In this respect, our approach is similar to the approach proposed by Heiseler et al. [21] to classify images. While our decision to combine classifiers in a hierarchy was motivated by the need to obtain higher accuracy, for Heiseler et al., performance was a key consideration. Their application requires the classification of a large number of images considering a very large number of features, some of which can be expensive to identify. The classifier hierarchy they proposed contains a coarse classifier at the root which is used to remove large portions of an image’s background (which is easily and cheaply identifiable). The other classifiers that use features that are less common or more expensive to identify are placed in the bottom of the hierarchy. Our strategy for composing classifiers is also related to the sequential classifier model proposed by Even-Zohar and Roth [14], who applied this idea to part-of-speech tagging.

8. CONCLUSION

This paper presents a solution to the problem of identifying online databases among a heterogeneous set of Web forms automatically gathered by a focused crawler. Our approach composes two classifiers in a hierarchical fashion by partitioning the space into structural features and content. This composition not only allows the construction of simpler classifiers, but it also enables the use of learning techniques that are more effective for each feature subset. In addition, since all the features used in the classification process can be automatically extracted, our solution is scalable. Last, but not least, because the proposed form-filtering process uses learning techniques, it is general and can be applied to many different domains. The high precision and recall obtained in our experimental evaluation indicate that our approach is a scalable alternative to the problem of online database classification. HIFI can be used as a basic building

block for large-scale information integration tasks. For example, it can be used to generate sets of homogeneous forms required in form-schema matching and merging [18, 20, 33]; and it can help automate the process of constructing online database collections such as the one described in [15] as well as improve the quality of directories like BrightPlanet [8].

Because HIFI relies on the form content for classification purposes, it is not able to reliably identify the domain of simple search forms, whose contents have little or no information related to the database schema and contents. An interesting direction we plan to pursue in future work is to try and combine our pre-query classification strategy with probing-based post-query methods.

Acknowledgments. The authors thank Eun Yong Kang for his help in collecting and labeling the data used in the experiments. This work is partially supported by the National Science Foundation (under grants IIS-0513692, CNS-0524096, IIS-0534628) and a University of Utah Seed Grant.

9. REFERENCES

- [1] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press/Addison-Wesley, 1999.
- [2] L. Barbosa and J. Freire. Siphoning Hidden-Web Data through Keyword-Based Interfaces. In *Proc. of SBBD*, pages 309–321, 2004.
- [3] L. Barbosa and J. Freire. Searching for Hidden-Web Databases. In *Proceedings of WebDB*, pages 1–6, 2005.
- [4] L. Barbosa, J. Freire, and A. Silva. Organizing hidden-web databases by clustering visible web documents. In *Proceedings of ICDE*, 2007. To appear.
- [5] P. Bennett, S. Dumais, and E. Horvitz. Probabilistic combination of text classifiers using reliability indicators: Models and results. In *Proceedings of SIGIR*, 2002.
- [6] P. N. Bennett, S. T. Dumais, and E. Horvitz. The combination of text classifiers using reliability indicators. *Information Retrieval*, 8(1):67–100, 2005.
- [7] A. Bergholz and B. Chidlovskii. Crawling for Domain-Specific Hidden Web Resources. In *Proceedings of WISE*, pages 125–133, 2003.
- [8] Brightplanet’s searchable databases directory. <http://www.completeplanet.com>.
- [9] S. Chakrabarti, K. Punera, and M. Subramanyam. Accelerated focused crawling through online relevance feedback. In *Proceedings of WWW*, pages 148–159, 2002.
- [10] S. Chakrabarti, M. van den Berg, and B. Dom. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. *Computer Networks*, 31(11-16):1623–1640, 1999.
- [11] K. C.-C. Chang, B. He, and Z. Zhang. Toward Large-Scale Integration: Building a MetaQuerier over Databases on the Web. In *Proc. of CIDR*, pages 44–55, 2005.
- [12] J. Cope, N. Craswell, and D. Hawking. Automated Discovery of Search Interfaces on the Web. In *Proceedings of ADC*, pages 181–189, 2003.
- [13] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori. Focused Crawling Using Context Graphs. In *Proceedings of VLDB*, pages 527–534, 2000.
- [14] Y. Even-Zohar and D. Roth. A sequential model for multi-class classification. In *Empirical Methods in Natural Language Processing*, 2001.
- [15] M. Galperin. The molecular biology database collection: 2005 update. *Nucleic Acids Res*, 33, 2005.
- [16] S. Gangaputra and D. Geman. A design principle for coarse-to-fine classification. In *Proceedings of CVPR*, pages 1877–1884, 2006.
- [17] L. Gravano, P. G. Ipeirotis, and M. Sahami. Qprober: A system for automatic classification of hidden-web databases. *ACM TOIS*, 21(1):1–41, 2003.
- [18] B. He and K. C.-C. Chang. Statistical Schema Matching across Web Query Interfaces. In *Proceedings of ACM SIGMOD*, pages 217–228, 2003.
- [19] B. He, T. Tao, and K. C.-C. Chang. Organizing structured web sources by query schemas: a clustering approach. In *Proc. of CIKM*, pages 22–31, 2004.
- [20] H. He, W. Meng, C. Yu, and Z. Wu. Wise-integrator: An automatic integrator of web search interfaces for e-commerce. In *Proceedings of VLDB*, pages 357–368, 2003.
- [21] B. Heisele, T. Serreb, S. Prenticeb, and T. Poggiob. Hierarchical Classification and Feature Reduction for Fast face Detection with Support Vector Machines. *Pattern Recognition*, 36(9), 2003.
- [22] A. Hess and N. Kushmerick. Automatically attaching semantic metadata to web services. In *Proceedings of IWeb*, pages 111–116, 2003.
- [23] W. Hsieh, J. Madhavan, and R. Pike. Data management projects at Google. In *Proceedings of ACM SIGMOD*, pages 725–726, 2006.
- [24] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [25] S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. In *Proceedings of VLDB*, pages 129–138, 2001.
- [26] J. Rennie and A. McCallum. Using Reinforcement Learning to Spider the Web Efficiently. In *Proceedings of ICML*, pages 335–343, 1999.
- [27] Y. Ru and E. Horowitz. Indexing the invisible Web: a survey. *Online Information Review*, 29(3):249–265, 2005.
- [28] E. H. Simpson. Measurement of Diversity. *Nature*, 163:688, 1949.
- [29] S. Sizov, M. Biber, J. Graupmann, S. Siersdorfer, M. Theobald, G. Weikum, and P. Zimmer. The BINGO! System for Information Portal Generation and Expert Web Search. In *Proc. of CIDR*, 2003.
- [30] The UIUC Web integration repository. <http://metaquerier.cs.uiuc.edu/repository>.
- [31] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition, 2005.
- [32] P. Wu, J.-R. Wen, H. Liu, and W.-Y. Ma. Query selection techniques for efficient crawling of structured web sources. In *Proceedings of ICDE*, page 47, 2006.
- [33] W. Wu, C. Yu, A. Doan, and W. Meng. An Interactive Clustering-based Approach to Integrating Source Query interfaces on the Deep Web. In *Proceedings of ACM SIGMOD*, pages 95–106, 2004.