Ontology-based Focused Crawling*

Hiep Phuc Luong
CSCE Department
University of Arkansas
hluong@uark.edu

Susan Gauch
CSCE Department
University of Arkansas
sgauch@uark.edu

Qiang Wang
CSCE Department
University of Arkansas
qxw002@uark.edu

Abstract

Ontology learning has become a major area of research whose goal is to facilitate the construction of ontologies by decreasing the amount of effort required to produce an ontology for a new domain. However, there are few studies that attempt to automate the entire ontology learning process from the collection of domain-specific literature, to text mining to build new ontologies or enrich existing ones. In this paper, we present a framework of ontology learning that enables us to retrieve documents from the Web using focused crawling in a biological domain, amphibian morphology. We use a SVM (Support Vector Machine) classifier to identify domain-specific documents and perform text mining in order to extract useful information for the ontology enrichment process. This paper reports on the overall system architecture and our initial experiments on the focused crawler and document classification.

1. Introduction

The next generation of the Semantic Web focuses on supporting a better cooperation between humans and machines [2]. In this approach, ontologies play an important role as a backbone for providing and accessing knowledge sources. Since manual building of ontology is costly, time-consuming, error-prone and inflexible to change, it is hoped that an automated process will result in a better ontology construction and create ontologies that better match a specific application [13]. These ontology learning approaches can be distinguished by the type of input used for learning, e.g., they can learn from text, from a dictionary, from a knowledge base, from a semi-structured schemata, or from relational schemata [9] [16]. Currently, few projects attempt to support the entire ontology learning process including automated support for tasks such as retrieving documents, classifying, filtering and extracting relevant information for the ontology enrichment.

Most existing approaches for ontology learning require a large number of input documents for accurate results [15]. With the enormous growth of the Web, it is important to develop document discovery mechanisms based on intelligent techniques such as focused crawling [4] to make this process easier for a new domain. Focused crawlers go a step further than classic crawlers in order to be able to quickly collect Web pages about a particular topic or domain of the Web [7]. In our work, we use focused crawling to retrieve documents and information in a biological domain, i.e., amphibian, anatomy and morphology, by using a combination of general search engines, scholarly search engines, and online digital libraries. Due to the huge number of retrieved documents, we require by an automatic mechanism rather than domain experts in order to separate out the documents that are truly relevant to the biological domain of interest. Since SVM has been recognized as one of the most successful current classification methods, we have adopted it for the classification task [17].

In this paper, we report on our initial work on the ontology learning process through web focused crawling and information extraction applied to the domain of amphibian anatomy and morphology. The potential documents in this domain are gathered, classified to identify the best candidates for analysis, and then mined to extract the relevant information for the ontology enrichment process. In section 2, we present a survey of current research on ontology learning, focused crawlers, document classification,

* This research is partially supported by the NSF grant DBI-0445752: Semi-Automated Construction of an Ontology for Amphibian Morphology.
and information extraction. In section 3, we present our ontology learning framework and its main architectural components. We also underline the process of document classifying and filtering by using SVM technique. Section 4 presents some initial experimental results for our approach and discusses on the usability of our work. The final section presents conclusions and discusses our ongoing and future work in this area.

2. Related Work

An ontology is an explicit, formal specification of a shared conceptualization of a domain of interest [10], where formal implies that the ontology should be machine-readable and the domain can be any that is shared by a group or community. In our view, there are two main approaches to ontology building: (i) manual construction of an ontology from scratch, and (ii) semi-automatic construction using tools or software with human intervention. It is hoped that semi-automatic generation of ontologies will substantially decrease the amount of human effort required in the process [13][15].

Ontology learning has recently been studied as an effective approach to facilitate the semi-automatic construction of ontologies by ontology engineers or domain experts. Gómez-Pérez et al. [9] present a good summary of several ontology learning projects that are concerned with knowledge acquisition from a variety of sources such as text documents, dictionaries, knowledge bases, relation schemas, semi-structured data, etc. Many of these existing approaches employ ontology learning from text documents [3], although only a few deal with ontology enrichment from documents collected from the Web. Omelayenko [15] has discusses the applicability of machine learning algorithms to learning of ontologies from Web documents and also surveys the current ontology learning and other closely related approaches. Similar to our approach, authors in [13] introduces an ontology learning framework for the Semantic Web which proceeds through ontology import, extraction, pruning, refinement, and evaluation giving the ontology engineers a wealth of coordinated tools for ontology modeling. However, they do not mention any automated support to collect the domain documents from the Web or how to automatically identify domain-relevant documents needed by the ontology learning process. In another approach similar to ours, [1] has presents an automatic method to enrich very large ontologies, e.g., WordNet, that uses documents retrieved from the Web. However, in their approach, the query strategy is not entirely satisfactory in retrieving relevant documents which affects the quality and performance of the topic signatures and clusters. Moreover, they do not apply any filtering techniques to verify that the retrieved documents are truly on-topic.

Many ontology learning approaches require a large collection of input documents in order to enrich the existing ontology [15]. A common way to get these documents from the Web is to use general purpose crawlers and search engines, but this approach faces problems with scalability due to the rapid growth of the Web. In contrast, focused crawlers overcome this drawback, i.e., they yield good recall as well as good precision, by restricting themselves to a limited domain [7]. Authors in [4] describe a new hypertext resource discovery system with the purpose of selectively seeking out pages that are relevant to a pre-defined set of topics. Ester et al. [7] also introduce a generic framework for focused crawling consisting of two major components: (i) specification of the user interest and measuring the resulting relevance of a given web page; and (ii) a crawling strategy. In order to improve accuracy of the learned ontologies, the documents retrieved by focused crawlers may need to be automatically filtered by using some text classification technique such as Support Vector Machines (SVM), k-Nearest Neighbors, Linear Least-Squares Fit, TF-IDF, etc. A thorough survey and comparison of such techniques such as Support Vector Machines (SVM), k-Nearest Neighbors, Linear Least-Squares Fit, TF-IDF, etc. A thorough survey and comparison of such.

3. Framework

3.1. Architecture

In this section, we present architecture of our ontology learning process framework that incorporates crawling, classifying and extracting relevant information in the amphibian and morphology domain from Internet documents. The main processes are as following (see Figure 1):

- We begin with an existing small, manually-created amphibian morphology ontology [12]. From this, we automatically generate queries for each concept in the hierarchically-structured ontology.

- We use a topic-specific spider (focused crawler) to submit these queries to a variety of Web search engines and digital libraries.

- Next, we apply SVM classification to filter out documents in the search results that match the query well but which are less relevant to the domain of our ontology.
After the above process, we have created a collection of documents relevant to amphibian morphology. These are input to an information extraction (IE) system to mine information from documents that can be used to enrich the ontology. We plan to use a combination of pattern-based extraction methods, e.g., GATE [6] and statistical NLP algorithms to identify attributes to enrich the ontology. We have completed initial work with vocabulary enrichment but this work will not be further discussed in this paper.

![Figure 1. Architecture of ontology learning framework](image)

### 3.2. Amphibian Morphology Ontology

The need for terminological standardization of anatomy is particularly pressing in amphibian morphological research [12]. By standardizing the lexicon used for diverse biological studies related to anatomy, an amphibian ontology will facilitate the integration of anatomical data representing all orders of amphibians, thus enhancing knowledge representation of amphibian biology and diversity.

A long-term AmphibAnat\(^2\) NSF-sponsored project aims at integrating the amphibian anatomical ontology knowledge base with systematic, biodiversity, embryological and genomic resources. However, another important goal of this project is to semi-automatically construct and enrich the amphibian anatomical ontology. From a manually constructed seed ontology, we use focused crawler and data-mining software in order to mine electronic resources for instances of concepts and properties to be added to the existing ontologies. The current amphibian ontology created by this project is available in two main formats: (i) OWL and (ii) OBO - Open Biomedical Ontology.

The amphibian anatomy semantic network consists of 212 semantic concepts and 58 relationships [12].

### 3.3. Searching and Crawling Documents

In order to collect a corpus of documents from which ontological enrichments can be mined, we use the seed ontology as input to our topic specific spider. For each concept in a selected subset of ontology, we generate a query that is then submitted to two main sources, i.e., search engines and digital libraries. To aid in query generation strategies, we created an interactive system that enables us to create queries from existing concepts in the ontology and allows us to change parameters such as the website address, the number of returned results, the format of returned documents, etc.

From our exploration, we found that if we use the concept name, e.g., “anatomical system,” alone as a query, we retrieve very few relevant results. However, by expanding the query containing the concept name with keywords describing the ontology domain overall, e.g., “amphibian” and/or “morphology” and also query for type of result we want, e.g., “.pdf”, we get a larger number of relevant results. Based on these explorations, we created an automated module that, given a concept in the ontology, currently generates 3 queries with the expansion added, e.g., “amphibian” “morphology” “.pdf”.

We next automatically submit the ontology-generated queries to multiple search engines and digital libraries related to the domain (e.g., Google, Yahoo, Google Scholar, http://www.amphibanat.org). For each query, we process the top 10 results from each search site using an HTML parser\(^1\) to extract the hyperlinks.

The current amphibian ontology is large, containing more than 380 concepts\(^4\), and our goal is to develop techniques that can minimize manual effort by growing the ontology from a small, seed ontology. Thus, rather than using the whole ontology as input to the system, we expect to use a subset of approximately 5 concepts. Ultimately, we hope to compare the larger ontology we build to the full ontology built by domain expert. From each of the 5 concepts, we generate 3 queries, for a total of 15 automatically generated queries. Each query is then submitted to each of the 4 search sites from which the top 10 results are requested. This results in a maximum of 600 documents to process.

### 3.4. Classifying and Filtering Documents

Although documents are retrieved selectively through restricted queries and by focused crawling, we

---

still need a mechanism to evaluate and verify the relevance of these documents to the predefined domain of amphibian morphology. We use LIBSVM 5 to separate the remaining documents into two main categories: (i) relevant and (ii) non-relevant to the domain of amphibian morphology. Only documents that are deemed truly relevant are input to the pattern extraction process.

The SVM classification algorithm must first be trained, based on labeled examples, so that it can accurately predict unknown data (i.e., testing data). The training phase consists of finding a hyperplane that separates the elements belonging to two different classes. According to [5], for median-sized problems, cross-validation might be the most reliable way to select SVM parameters so that the classifier is as accurate as possible. First, the training data is separated into several folds. Sequentially, one fold is considered as the validation set and the rest are used for training. The average of accuracy on predicting the validation sets is the cross-validation accuracy.

In our situation there are not enough examples to accurately train the classifier on all features. Thus, we may need to choose a subset of features before submitting the data to SVM [5][19]. To identify the most important features, we calculate the weights of tokens in documents using the KeyConcept package [8]. Each document is represented by a vector of values \( wt_i \), where \( wt_i \) is calculated by the term frequency \( tf \) / size of document (i.e., normalized by document size), and the inverse document frequency \( idf_i \) is calculated from dictionary over all documents.

### 4.1. Training the Classifier

In this section, we present our experiments on training the SVM classifier to filter out the non-relevant search result. However, since all documents are top results retrieved from domain-relevant queries, the vocabulary overlap between the relevant and irrelevant documents is high, making this a challenging task for an automatic classifier, even one as good as SVM. Thus, the training phase is of particular importance in our work. Using the interactive ontology-based query system described in section 3.3, we manually created a corpus of 60 relevant and 60 irrelevant Web documents retrieved by our concept-generated queries in HTML, pdf and text formats.

4 documents each. For each run, two subsets are held back for testing, i.e., 12 relevant and 12 non-relevant documents, and the classifier is trained on the remaining 96 documents, 48 from each category. Thus, using five-fold cross-validation, each instance in the test collection is predicted once and the cross-validation accuracy is the percentage of documents that are correctly classified. We carry out training the classifier with and without feature selection and evaluate a variety of feature selection algorithms. For each approach, the selected features are weighted using \( tf*idf \) normalized by document size.

To identify important features for classification, we try to select those features which are much more important in the relevant set or irrelevant set. Tokens that appear equally frequently in both subsets are not good features for distinguishing between them. Thus, we calculate the frequency of each token in the relevant training set and also its frequency in the irrelevant training set. Finally, we calculate the frequency difference (FD) as the absolute difference between those two values to identify those features more strongly associated with one subset or the other. Another set of tokens that we consider as potentially important for classification is those tokens that appear only in one subset or the other. These are the one-subset tokens. We also experimented with using features that are important content descriptors for the documents, i.e., those tokens that appear in many, but not all, documents and those which have high normalized \( tf*idf \) weights, meaning that they are important representations of the document contents. We call this high distribution tokens selection.

### 4.2. Experiments

We compared four main feature selection methods as follows:

1. **No feature selection (Baseline):** We use all tokens from all documents in the training collection as features. This is our baseline against which other approaches are compared.

2. **Feature selection with frequency difference (FD) only:** In this approach, we select only those tokens whose FD value is above a given threshold. We vary the FD values from 1 (all features) to 1181, at which point only 1 feature remains.

3. **Feature selection with frequency difference (FD) and one-subset selection:** Features are selected as the same way and FD variation as in the above case; however we augment the feature with those tokens that appeared in only one subset.

4. **Feature selection with high distribution tokens (HDT):** We change some parameters \( m, n \) and

---

5 http://www.csie.ntu.edu.tw/~cjlin/libsvm/
TopN, where \( m \) and \( n \) are the maximum and minimum number of documents containing the feature respectively, and \( TopN \) is the number of features selected from each document, chosen selecting the highest weighted tokens. Experiments in this case cover all training documents distribution ranges corresponding with four values pairs \((m, n)\) = (36, 12), (60, 36), (84, 60) and (96, 0), with \( TopN \) varies from 1 to 110.

4.3. Evaluation of Classification Results

Classification effectiveness is usually measured in terms of the classic IR notions of precision, recall and F-measure. They can also be adapted to the case of text categorization. Denote \( TP, FP, TN, FN \) the number of true/false positives/negatives of returned results. These measures are calculated as following:

- **Precision** \((P) = \frac{TP}{TP+FP}\)
- **Recall** \((R) = \frac{TP}{TP + FN}\)
- **F-measure** \(\beta = \frac{(\beta^* \beta + 1)P * R}{\beta^* P + R}\) where \(\beta\) allowing differential weighting of \(P\) and \(R\).

The best accuracy achieved with the FD only method is \(P=79.5\%\) and \(R=50.7\%\) with \(FD = 145\).
greater than 70%, with appropriate thresholds. We compared a variety of methods, and the FD method based on tokens that appear more frequently in the in either the relevant or non-relevant training sets performed the best. Adding in words that appeared in only one subset degraded performance as did a method based on the number of documents that contained the word (HDT) rather than the word frequency in each subset. When we only took tokens that occurred in many training documents, we got better accuracy than the baseline that considered all tokens from all documents, but this method’s maximum accuracy was only 68.85% when tokens with the highest document counts were used. Overall, the best-performing method was FD only that achieved an accuracy of 77.5%. With a bias towards high precision, this method worked best with tokens that appeared at least 159 times more frequently in one training subset versus the other, with a high threshold of 0.8 for inclusion in the relevant class. In this case, there are 162 features used for classification which is far fewer than that total set of 40,265 features used with no feature selection.

5. Conclusion and Future Work
In this paper, we have presented a general ontology learning framework including automated support for tasks of retrieving documents, classifying, filtering and extracting relevant information for the ontology enrichment. We have studied and implemented a focused crawler enabling us to retrieve documents in the domain of amphibian and morphology from some digital library websites or search engines. The core of our presented work is the evaluation of our SVM-based filtering technique that automatically filters out the non-relevant documents collected by the crawler so that only those most likely to be relevant are passed along for information extraction. Although the automatic collection is quite accurate, over 77.5%, this classifier could be used semi-automatically in future to allow experts to do further filtering.

Our main tasks in the future are to validate the focused crawler on a wider range of documents, implement and disseminate an interactive tool for corpus creation that would work with any ontology, and implement and evaluate a variety of ontology learning methods based on the domain-specific corpus.

6. References