An Ontology Learning Framework Using Focused Crawler and Text Mining

Hiep Phuc Luong
CSCE Department
University of Arkansas
Fayetteville, AR, USA
hluong@uark.edu

Susan Gauch
CSCE Department
University of Arkansas
Fayetteville, AR, USA
sgauch@uark.edu

Qiang Wang
CSCE Department
University of Arkansas
Fayetteville, AR, USA
qxw002@uark.edu

Anne Maglia
Missouri University of
Science and Technology
Rolla, MO, USA
magliaa@mst.edu

Abstract— **Manual ontology construction is costly, time-consuming, error-prone and inflexible to change. To address these problems, researchers hope that an automated process will result in faster and better ontology construction and enrichment. Ontology learning has become recently 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, most of current approaches are dealing with some specific tasks or a part of the ontology learning process rather than providing complete support to users. There are few studies that attempt to automate the entire ontology learning process from the collection of domain-specific literature, filtering out documents irrelevant to the domain, to text mining to build new ontologies or enrich existing ones.**

In this paper, we present a complete framework for ontology learning that enables us to retrieve documents from the Web using focused crawling and then 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. Our experimental results of this framework in the amphibian morphology domain support our belief that we can use SVM and text mining approaches to improve the identification of documents and relevant words suitable for the ontology enrichment. This paper reports on the overall system architecture and our initial experiments of all phases in our ontology learning framework, i.e., document focused crawling, document classification and information extraction using text mining techniques to enrich the domain ontology.

Keywords – ontology learning; focused crawler; SVM; text mining; amphibian ontology

I. INTRODUCTION

The next generation of the Semantic Web focuses on supporting a better cooperation between humans and machines [3]. In this approach, ontologies play an important role as a backbone for providing and accessing knowledge sources. However, creating ontologies for the many and varied domains on the Web is a time-consuming process and their construction is a major bottleneck to the wider deployment and use of semantic information on the Web. Since manual ontology construction 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 [17]. 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 a relational schemata [10] [21]. 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 [20]. With the enormous growth of the Web, it is important to develop document discovery mechanisms based on intelligent techniques such as focused crawling [5] 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 [8]. 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 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 [23].

We have previously reported our results on collecting potential documents by using web focused crawlers, then filtering and classifying them to identify the best candidates for analysis [1]. To summarize, we found that SVM can be used to improve the identification of documents suitable for the ontology learning process. This paper extends that work in two directions. First, we present results for the information extraction process that allows us to extract the relevant information for ontology enrichment. Second, this paper describes our complete ontology learning approach and continuing work on the progress of enriching relevant vocabularies for the amphibian morphology ontology from the retrieved documents by using text mining techniques. Overall, our classification of relevant documents achieved the good prediction accuracy of 77.5% with the best-performing method of SVM algorithm (i.e., feature selection with frequency difference only). The text mining algorithm also produced good accuracy, over than 81% for all cases and reached the precision is 88% in the best case.

The goal of this research study is to implement and validate an ontology learning framework 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, information extraction and text mining methods. 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 as well as the information extraction using text mining. Section 4 presents some initial experimental results for our approach. Next, we discuss on the results achieved and the usability of our work in the section 5. The final sections present our conclusions and discuss our future work in this area.

II. Related Work

An ontology is an explicit, formal specification of a shared conceptualization of a domain of interest [11], where formal implies that the ontology should be machine-readable and the domain can be any that is shared by a group or community. Much of current research into ontologies focuses on construction and updating issues. 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][20].

Ontology learning has recently been studied as an effective approach to facilitate the semi-automatic construction of ontologies by ontology engineers or domain experts. Ontology learning uses methods from a diverse spectrum of fields such as machine learning, knowledge acquisition, natural language processing, information retrieval, artificial intelligence, reasoning, and database management [21]. Gómez-Pérez et al. [10] 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 [4], although only a few deal with ontology enrichment from documents collected from the Web. Omelayenko [20] 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 [17] 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. In addition to a general framework and architecture, they have implemented Text-To-Onto system supporting ontology learning from free text, from dictionaries, or from legacy ontologies. 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. Maedche et al. have presented in another paper [18] a comprehensive approach for bootstrapping an ontology-based information extraction system with the help of machine learning. They also presented an ontology learning framework which is one important step in their overall bootstrapping approach but it has still been described as a theoretic model and did not deal with the specific techniques used in their learning framework.

In another approach similar to ours, [2] 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. Inspiring the idea of using WordNet to enrich vocabulary for ontology domain, we have presented the lexical expansion from WordNet approach [15] providing a method of accurately extract new vocabulary for an ontology for any domain covered by WordNet.

Many ontology learning approaches require a large collection of input documents in order to enrich the existing ontology [20]. 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 [8]. Authors in [5] 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. [8] 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 methods and their complexity is presented in [27] and the authors in [23] conclude that SVM to be most accurate for text classification and fast training. SVM [24] is a machine learning model that finds an optimal hyperplane to separate two then classifies data into one of two classes based on the side on which they are located [6] [14].

Text mining, also known as text data mining or knowledge discovery from textual databases, refers generally to the process of extracting interesting and non-trivial patterns or knowledge from unstructured text documents [13] [25]. Tan [12] presents a good survey of text mining products/applications and aligns them based on the *text refining* and *knowledge distillation* functions as well as the *intermediate form* that they adopt. One approach similar to ours has presented a supervised ontology learning system using text mining [22]. Speretta et al used WordNet [19] similarity measures to select candidate tokens in a relatively narrow space in order to enrich the ontology. Although we share the same goal, we try to find a general and efficient way to extract a broader collection of accurate candidate tokens for ontology enrichment process that would work with any ontology.

III. Ontology Learning Framework

In this section, we first present the overall architecture of our ontology learning framework. Then, each component in this framework is described in detail in the following sections.

A. *Architecture*

Figure 1 presents the architecture of our ontology learning process framework that incorporates crawling, classifying, filtering 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 [16]. This ontology is created in the project AmphibAnat[1] with the purpose of creating a standardization of anatomy particularly pressing in amphibian morphological domain. From this ontology, 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 (e.g., Google, Scholar Google, Yahoo) and digital libraries. The spider downloads the potentially relevant documents listed on the first page (top-ranked) results. We also provide options to customize the number of returned results, the formats of returned documents, the list of search engines that are used to query documents, etc.
- 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 amphibian 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. In our previous work [1], we planned to use a combination of pattern-based extraction methods, e.g., GATE tool [7] and statistical NLP algorithms to identify attributes to enrich the ontology. This one has been used largely by several existing researches in information extraction field. However, in this paper, we present our new results achieved by using text mining methods in the information extraction phase in order to mine new relevant vocabularies from the collection of amphibian documents. We have completed several experiments with vocabulary enrichment and this work will be further discussed in following sections.

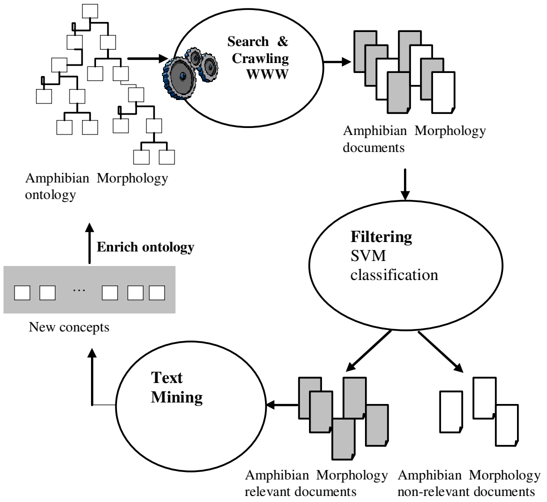

Search & Crawling WWW
Amphibian Morphology documents
Amphibian Morphology ontology
Enrich ontology
Filtering
SVM classification
...
New concepts
Text Mining
Amphibian Morphology relevant documents
Amphibian Morphology non-relevant documents

Figure 1. Architecture of ontology learning framework

B. *Amphibian Morphology Ontology*

Our proposed ontology learning framework can be used for any ontology in general domain. However, in order to validate the feasibility and effectiveness of our ontology learning approach, we have applied this framework into a specific domain, i.e., biology, anatomy and morphology, and do experiments with the Amphibian Anatomical Ontology [16].

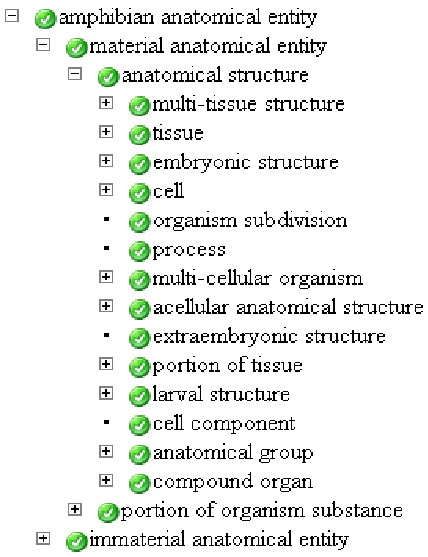
The need for terminological standardization of anatomy is particularly pressing in amphibian morphological research [16]. By standardizing the lexicon used for diverse biological studies related to anatomy, an amphibian ontology will facilitate the integration of anatomical data representing all orders of amphibians, thus enhancing knowledge representation of amphibian biology and diversity.

According to authors in [16], there are several main challenges to developing an ontology for amphibian morphology. First, the separate anatomical lexicons must be reconciled. Second, there are about 6,000 species of

[1] http://amphibanat.org/

amphibians for which the anatomical terminology must be resolved. Although much of the terminology will be similar across species, among-species variation will lead to a much larger ontology than those developed for a single model species. Third, because of anatomical diversity among amphibian orders, homologies of some structures are unknown; therefore, assigning terminological standards to them may be problematic. These challenges can be overcome if we forge a partnership between the amphibian morphological community and the power of information extraction technology. Therefore, one of the main goals of the long-term AmphibAnat (http://amphibanat.org/) NSF-sponsored project is to aim at integrating the amphibian anatomical ontology knowledge base with systematic, biodiversity, embryological and genomic resources.

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 a 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 [1]. The current amphibian ontology created by this project consists of 968 different semantic concepts and 570 relationships (main properties are *is_a* and *part_of*). [16]. Figure 1 presents a part of this ontology which is available in two main formats: (i) OWL and (ii) OBO - Open Biomedical Ontology.



- ⊟ amphibian anatomical entity
  - ⊟ material anatomical entity
    - ⊟ anatomical structure
      - ⊞ multi-tissue structure
      - ⊞ tissue
      - ⊞ embryonic structure
      - ⊞ cell
      - organism subdivision
      - process
      - ⊞ multi-cellular organism
      - ⊞ acellular anatomical structure
      - extraembryonic structure
      - ⊞ portion of tissue
      - ⊞ larval structure
      - cell component
      - ⊞ anatomical group
      - ⊞ compound organ
    - ⊞ portion of organism substance
  - ⊞ immaterial anatomical entity

Figure 2. A part of the amphibian ontology

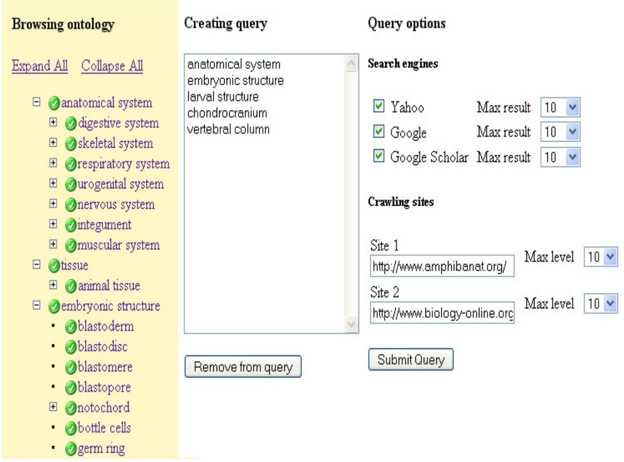
C. Searching and Crawling Documents

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.

Before we could automatically generate queries from an ontology, we explored a variety of query generation strategies. To aid in this exploration, we created an interactive system that allowed us to easily create a queries and evaluate search engines. Figure 3 shows the interface to this 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, etc.). For each query, we process the top 10 results from each search site using an HTML parser[2] to extract the hyperlinks. We have implemented some simple rules in order to automatically filter these hyperlinks to remove obviously irrelevant links, e.g., advertisement links, go-to-section links. The remaining links are then sent to the download module in order to retrieve the full documents. The results pages may contain documents in many formats, but we are interested only in HTML, pdf and text documents.



Browsing ontology

Expand All Collapse All

- ⊟ anatomical system
  - ⊞ digestive system
  - ⊞ skeletal system
  - ⊞ respiratory system
  - ⊞ urogenital system
  - ⊞ nervous system
  - ⊞ integument
  - ⊞ muscular system
- ⊟ tissue
  - ⊞ animal tissue
- ⊟ embryonic structure
  - blastoderm
  - blastodisc
  - blastomere
  - blastopore
  - ⊞ notochord
  - bottle cells
  - germ ring

Creating query

anatomical system
embryonic structure
larval structure
chondrocranium
vertebral column

Remove from query

Query options

Search engines

☑ Yahoo Max result 10
☑ Google Max result 10
☑ Google Scholar Max result 10

Crawling sites

Site 1 http://www.amphibanat.org/ Max level 10
Site 2 http://www.biology-online.org Max level 10

Submit Query

Figure 3. Creating queries from ontology concepts for focused crawling

[2] http://htmlparser.sourceforge.net/

D. 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 classification tool [6] 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 [6], 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 to 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 [6][26]. To identify the most important features, we calculate the weights of words in documents using the KeyConcept package [9]. Each document is represented by a vector of values $wt_i * idf_i$, 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. In section 4, we describe several feature selection methods and compare the classification results.

E. Information Extraction using Text Mining

We have so far a set of relevant documents which are closed to the domain of ontology. Our goal in this step is to extract structured and useful information from the actual text of these filtered documents. As stated in the previous section, we can use a combination of pattern-based extraction methods, e.g., GATE tool [7] and statistical NLP algorithms to identify attributes to enrich the ontology.

However, in our approach, we are aiming at producing a set of words that are most significantly related to the domain ontology by using text mining methods, then validating our algorithm. We have conducted two methods: (i) *Vector space approach* and (ii) *Part-of-speech approach* in order to calculate then rank the weights of words in relevant documents.

In the first approach, i.e., Vector space approach, we implement two algorithms, i.e., *Document-based* and *Corpus-based selection*, based on the vector space model. In order to guarantee words that are more representative of the ontology domain having higher rank values, we calculated *idf* (inverse document frequency) of words across 10,000 documents that were randomly downloaded from ODP[3] category.

*1) Document-based selection: calculates weights of words by using tf*idf*

$$W(i,j) = rtf_{(i,j)} * idf_i$$

$$rtf_{(i,j)} = \frac{tf_{(i,j)}}{N(j)}$$

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

with

$W(i,j)$ is the weight of term i in document j

$rtf_{(i,j)}$ is the relative term frequency of term i in document j

idf_i is the inverse document frequency of term i, which is pre-calculated across 10,000 ODP documents

$tf_{(i,j)}$ is the term frequency of term i in document j

$N(j)$ means the number of words in document j

$|D|$ is the total number of documents in the corpus

$|\{d:t_i \in d\}|$ is number of documents in which t_i appears.

We use a parameter k to control the length of the word list. A ranked word list is generated for each document. Then we take top k words from all lists and merge these words to only one list ranked by theirs weight. This word list created by this document-centric algorithm is called *L1*. We performed some preliminary experiments, not reported here, which varied k from 1 to 110. The results reported here use $k = 30$, a value that was found to perform well.

*2) Corpus-based selection: calculates weights of words by using sum(tf)*idf*

$$W(i) = \sum_{j=1}^{n} rtf_{(i,j)} * idf_i$$

with $W(i)$ is the weight of term i;

Other parameters are calculated as same as in the first algorithm. This word list created by this corpus-centric algorithm is called *L2*.

In the second approach, i.e., *Part-of-speech* approach, we exploit the fact that words describing ontology are usually nouns. Thus, we use only words that are nouns to generate word list. These two word lists, *L1N* and *L2N* corresponding to the subset of words on lists *L1* and *L2* that are tagged as nouns using the WordNet library [19] and JWI[4] (the MIT Java WordNet Interface).

We have totally carried out different experiments for four approaches, i.e., *L1*, *L2*, *L1N*, and *L2N*. In the following sections, we will present experiment results corresponding to each approach and discuss about their performance.

IV. Experimentation

In this section, we present experiments conducted on each component of our ontology learning framework.

[3] http://www.dmoz.org/

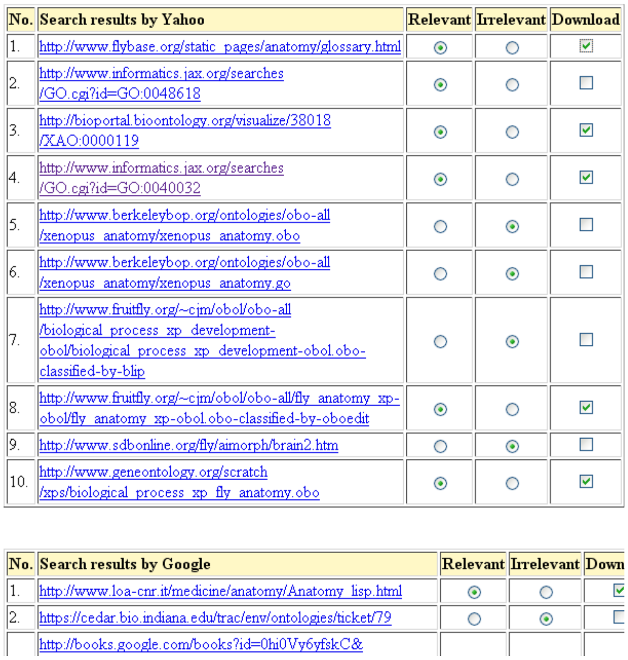
[4] http://projects.csail.mit.edu/jwi/

A. Experimentation of searching and crawling documents

The current amphibian ontology used in our experimentation is very large, containing more than 960 concepts[5]. However, due to a co-edition of this ontology among different specialists and developers in the AmphibAnat project, this current version contains many concept terms which are still not finalized (e.g., *fringe_on_postaxial_edge_of Toe_V*, *ventrolateral_process_of_palatoquadrate*, etc.) and noises data (e.g. *sp*, *aa*, *rr*, *ID_0000223*, etc.) that should be removed in the official version. Thus, the number of meaningful concepts that can be used for searching and crawling documents is decresed in our experiments. In addition, since ontology concepts are organized in hierarchy structure, there are many branches having concept names are very similar, for example the concept *foramen_acusticum_anterius* has two child concepts *foramen_acusticum_minus* and *foramen_acusticum_maius*. For this case, even we use all these concept names as keywords to look for online documents, the search results would not be better due to many duplicated words (e.g., *foramen, acusticum*) in these concepts. Therefore, we have focused on general and meaningful concept names that can be used to retrieval relevant documents in the amphibian morphology domain.

In addition, our goal is to develop techniques that can minimize manual effort by growing the ontology from a small and seed ontology, we have concentrated on experiments using a small set of keywords to search for relevant Web documents from the Internet. Thus, rather than using the ontology as input to the system, we expect to use a subset of concepts to validate our research approach. Ultimately, we hope to compare the larger ontology we build to the full ontology built by domain expert.

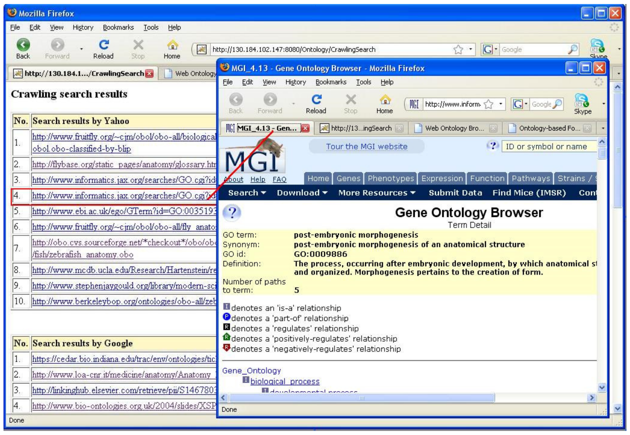
We chose a subset of 5 concepts from the amphibian ontology. From each of these concepts, we generated 3 queries with the expansion added (e.g., *"amphibian" "morphology" "pdf"*), for a total of 15 automatically generated queries. Each query was then submitted to each of the 4 search sites from which the top 10 results were requested. This resulted in a maximum of 600 documents to process. However, due to the fact that some search sites return fewer than 10 results for some queries and others are removed by our syntactic filtering and some returned documents by search engines are the same, in practice this number will be somewhat smaller. This process thus creates a very large number of hyperlinks to be analyzed, not all of which are likely to be truly relevant. Using some simple rules, these hyperlinks are automatically filtered to remove obviously irrelevant links, e.g., advertisement links and go-to-section links. The remaining links are then sent to the download module in order to retrieve the full documents. The results pages may contain documents in many formats, but we are select only HTML, pdf and text documents.



No.	Search results by Yahoo	Relevant	Irrelevant	Download
1.	http://www.flybase.org/static_pages/anatomy/glossary.html	●	○	☑
2.	http://www.informatics.jax.org/searches/GO.cgi?id=GO:0048618	●	○	□
3.	http://bioportal.bioontology.org/visualize/38018/XAO:0000119	●	○	☑
4.	http://www.informatics.jax.org/searches/GO.cgi?id=GO:0040032	●	○	☑
5.	http://www.berkeleybop.org/ontologies/obo-all/xenopus_anatomy/xenopus_anatomy.obo	○	●	□
6.	http://www.berkeleybop.org/ontologies/obo-all/xenopus_anatomy/xenopus_anatomy.go	○	●	□
7.	http://www.fruitfly.org/~cjm/obol/obo-all/biological_process_xp_development-obol/biological_process_xp_development-obol.obo-classified-by-blip	○	●	□
8.	http://www.fruitfly.org/~cjm/obol/obo-all/fly_anatomy_xp-obol/fly_anatomy_xp-obol.obo-classified-by-oboedit	●	○	☑
9.	http://www.sdbonline.org/fly/aimorph/brain2.htm	○	●	□
10.	http://www.geneontology.org/scratch/xps/biological_process_xp_fly_anatomy.obo	●	○	☑

No.	Search results by Google	Relevant	Irrelevant	Down
1.	http://www.loa-cnr.it/medicine/anatomy/Anatomy_lisp.html	●	○	☑
2.	https://cedar.bio.indiana.edu/trac/env/ontologies/ticket/79	○	●	□
	http://books.google.com/books?id=0hi0Vy6yfskC&			

Figure 4. Search results returned by search engines



Mozilla Firefox
File Edit View History Bookmarks Tools Help
Back Forward Reload Stop Home
http://130.184.102.147:8080/Ontology/CrawlingSearch
Google
http://130.184.1.../CrawlingSearch
Web Ontology
MGI_4.13 - Gene Ontology Browser - Mozilla Firefox
http://www.inform
Skype
MGI_4.13 - Gen...
http://13...ingSearch
Web Ontology Bro...
Ontology-based Fo...
Tour the MGI website
ID or symbol or name
MGI
About Help FAQ
Home Genes Phenotypes Expression Function Pathways Strains / S
Search Download More Resources Submit Data Find Mice (IMSR) Con
Done

Crawling search results

No.	Search results by Yahoo
1.	http://www.fruitfly.org/~cjm/obol/obo-all/biological obol.obo-classified-by-blip
2.	http://flybase.org/static_pages/anatomy/glossary.htm
3.	http://www.informatics.jax.org/searches/GO.cgi?id
4.	http://www.informatics.jax.org/searches/GO.cgi?id
5.	http://www.ebi.ac.uk/ego/GTerm?id=GO:0035193
6.	http://www.fruitfly.org/~cjm/obol/obo-all/fly_anato
7.	http://obo.cvs.sourceforge.net/*checkout*/obo/obo/fish/zebrafish_anatomy.obo
8.	http://www.mcdb.ucla.edu/Research/Hartenstein/re
9.	http://www.stephenjaygould.org/library/modern-sci
10.	http://www.berkeleybop.org/ontologies/obo-all/zeb

No.	Search results by Google
1.	https://cedar.bio.indiana.edu/trac/env/ontologies/tic
2.	http://www.loa-cnr.it/medicine/anatomy/Anatomy_
3.	http://linkinghub.elsevier.com/retrieve/pii/S1467803
4.	http://www.bio-ontologies.org.uk/2004/slides/XSP

Gene Ontology Browser
Term Detail

GO term:	post-embryonic morphogenesis
Synonym:	post-embryonic morphogenesis of an anatomical structure
GO id:	GO:0009886
Definition:	The process, occurring after embryonic development, by which anatomical st and organized. Morphogenesis pertains to the creation of form.
Number of paths to term:	5

I denotes an 'is-a' relationship
P denotes a 'part-of' relationship
R denotes a 'regulates' relationship
R denotes a 'positively-regulates' relationship
R denotes a 'negatively-regulates' relationship

Gene_Ontology
I biological_process

Figure 5. Review document content before deciding to download

Figure 4 shows the returned result by each search engine. This result has been already filtered to remove irrelevant links (e.g., advertisement links and go-to-section links…) and containing only useful links that would be considered as relevant to our domain. For each returned result, we can open and see the content of this result (by clicking its URL) before deciding to download that document and classify it into the appropriate document set (c.f. Figure 5). User then can choose which documents will belong to the relevant or irrelevant set. These selected documents will be downloaded to serve the SVM classification task.

B. Experiments on Classifying and Filtering Documents

In this section, we present our experiments on training the SVM classifier to filter out the non-relevant search

[5] http://amphibanat.org/

result. The automatic nature of the corpus creation process generates a large collection of documents, not all of which are likely to be suitable for information extraction. Since extracting information from irrelevant documents would degrade the quality of the resulting ontology, it is crucial to have a filtering stage to remove irrelevant and slightly relevant documents. 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 the section III.C, 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. These documents were converted into text format before using them with the SVM classifier.

1) *Training the Classifier*

The documents in each category, i.e., relevant and non-relevant, were divided into five subsets containing 12 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 evaluated 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 select those features that are most important in either the relevant set or the irrelevant set. Tokens that are appear equally frequently in both subsets are not good features for distinguishing between them. Thus, we calculated 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 considered as potentially important for classification is those tokens that appear only in one subset or the other. These are called the *one-subset* tokens.

We also experimented with using features that are important content descriptors for the documents, i.e., those tokens that are 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* (HDT) selection. To run this experiment, we use parameters *m, n* and *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.

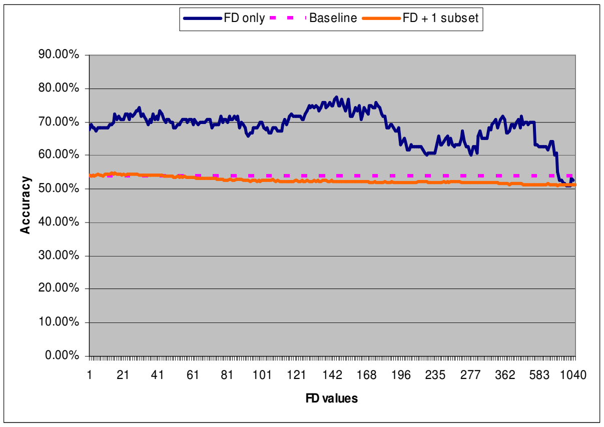


Figure 6. Average accuracy of Baseline and Feature selection with FD methods

2) *Experiments*

In the first experiment, we compared 3 feature selection methods:

- *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.
- *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.
- *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.

Figure 6 shows an overall view of the baseline and the feature selection with FD methods in which we can see their accuracy with different FD values from 1 to 1181. Among these methods, the feature selection with FD only obtains high average accuracy while using just one-subset for feature selection performs worse than the baseline. Based on these experiment results, we found that feature selection with FD only performs best when using features whose frequency difference between the relevant and irrelevant sets is between 130 and 161. The peak in accuracy, 77.5%, occurred at the FD value 145, using a threshold of 0.1. The number of selected features in this case was 162.

Once we had tuned the FD method, we explored the effect of adding terms based on their frequency in the relevant set or irrelevant set. In the second experiment, we select features important representations of the document contents:

- *Feature selection with high distribution tokens (HDT):* We varied parameters values of *m, n* and *TopN* to right parameters giving the best accuracy. 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.

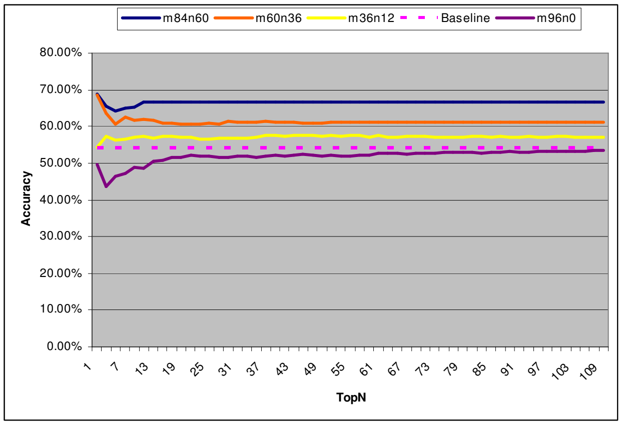


Figure 7. Average accuracy of Baseline and Feature selection with HDT methods

The second comparison (c.f. Figure 7) showed a better average accuracy result of the feature selection with high distribution tokens than the baseline. Among these lines corresponding to different parameters *m, n* and *TopN,* we found that the best result is obtained with the pair *(m, n)* = (84, 60). The results decrease if we take a range of documents having fewer features. If we choose the range covering all documents in training set and the *TopN* varies from 1 to 110, the accuracy is less than the one of the baseline as presented in the Figure 7.

C. Experimentation of Information Extraction using Text Mining

It is crucial to have a filtering stage to remove irrelevant and slightly relevant documents to the amphibian ontology. We have adopted an SVM-based classification technique trained on 60 relevant and 60 irrelevant documents collected from the Web. In earlier experiments, this spider was able to collect new documents and correctly identify those related to the domain with an average accuracy 77.5% [1].

Ultimately, the papers collected and filtered by the topic-specific spider will be automatically fed into the text mining software (with an optional human review in between). However, to evaluate the effectiveness of the text mining independently, without noise introduced by some potentially irrelevant documents, we ran our experiments using 60 documents manually judged as relevant, separated into two groups of 30, i.e., *Group_A* and *Group_B*. All these documents were preprocessed to remove HTML code, stop words and punctuation. First, we run experiment on the *Group_A* to find the case having the best result of extracting vocabulary correctly, and then we use documents in *Group_B* to validate our algorithm and compare results of these experiments.

In order to evaluate the effectiveness of the extracted words from documents, we created two *truth-lists* corresponding to the two approaches in the section 3.4. From the word list *L1* (623 words) and *L2* (623 words), after merging and removing duplicated words from these two lists, we generated the set of 507 unique words found by these two techniques. Similarly, a list of 253 unique words was generated from the lists *L1N* and *L2N*. These word lists then were judged by a human expert to classify words that are relevant or non-relevant to the amphibian morphology domain.

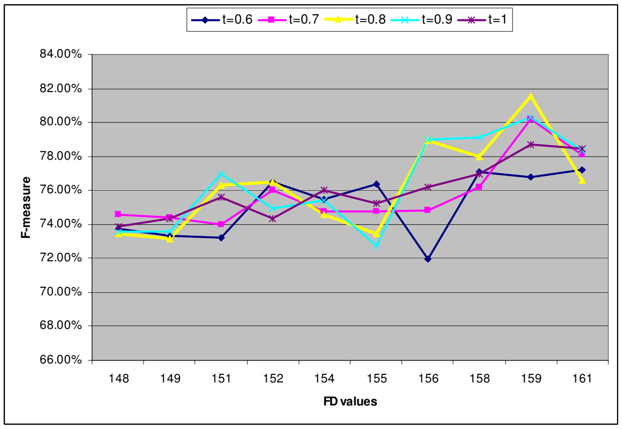


Figure 8. F-measure biased towards higher P

V. EVALUATION

We focus in this section on the performance evaluation of the two phases: SVM classification and Information extraction using text mining. For each phase, we define measures to evaluate its performance and effectiveness. We also show the comparative results and discuss the best case achieved for each phase.

A. Evaluation of SVM Classification Results

Classification effectiveness is usually measured in terms of the classic IR notions of *Precision (P), Recall (R)* and *F-measure (F).* 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:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F_{\beta} = \frac{\left(1+\beta^2\right) * P * R}{\beta^2 * P + R}$$

where β allowing differential weighting of *P* and *R*.

Our experiments show that the best accuracy achieved with the FD only method is P=77.5% and R=50.7% with FD = 145. We continue to evaluate how the results achieved are varied in the best method of FD only.

Because we want to perform information extraction only on truly relevant documents, we want a metric that is biased towards high precision versus high recall. We chose to use the *F-measure* with a β value that weights precision

4 times higher than recall, i.e., *β=0.25*. We calculated the F-measure for a range around the best performing method, i.e., FD values from 130-161. For each of these FD values, we varied the SVM classification thresholds from -1 to 1 in steps of 0.1. The calculated *F-measure* results vary regularly in this range, indicating that we are getting low sensitivity with the FD method. Figure 8 shows the F-measure results for the best performing thresholds. We found that the best-performing FD approach produced an F-measure (*β=0.25)* of 81.6% with a threshold of 0.8 and FD value=159.

B. *Evaluation of Information Extraction Results*

In order to measure the effectiveness of our information extraction phase, we use the classic IR metrics of Precision, Recall and F-measure. We define these measures as following:

Precision (P): measures the percentage of the correct words identified by our algorithm that matched those from the candidate words.

$$P = \frac{\#\_correct\_tokens\_identified}{\#\_candidate\_tokens}$$

Recall (R): measures the percentage of the correct words identified by our algorithm that matched those from the truth list words.

$$R = \frac{\#\_correct\_tokens\_identified}{\#\_truth-list\_tokens}$$

F-measure (F): is calculated as following

$$F_{\beta} = \frac{(1+\beta^2)*P*R}{\beta^2*P+R}$$

Because we want to enhance the ontology with only truly relevant words, we want a metric that is biased towards high precision versus high recall. We chose to use the F-measure with a β value that weights precision higher than recall. From several explorations, we found that β=0.25 is an adequate value, so we used this value in our experiment.

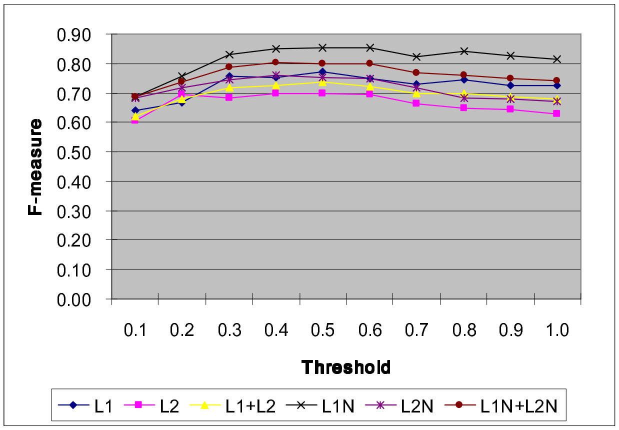


Figure 9. F-measure of the tests in Group_A

We evaluate results by comparing the candidate word lists that were extracted from the relevant documents using our algorithms with the judgments submitted by our human domain expert. We chose threshold values *t* from 0.1 to 1.0 corresponding to the percentage of top candidate words that are extracted (e.g., *t*=0.1 means that top 10% words are selected). We carried out 6 different tests corresponding to the four candidate lists, i.e., *L1*, *L2*, *L1N*, *L2N)* and two more cases *L1+L2* (average of *L1* and *L2*) and *L1N+L2N* (average of *L1N* and *L2N*) as input to our algorithm. These tests are named by their list names *L1, L2, L1+L2, L1N, L2N* and *L1N+L2N*. Figure 9 presents the F-measures achieved by these tests using various threshold values.

Figure 9 shows that the best result was achieved in the test *L1N*, using the highest weighted nouns extracted from individual documents. By analyzing results, we find that if we want a higher precision, the recall and F-measure values would decrease. We harmonize the two important values of precision and F-measures, so the best performance is achieved with a threshold *t*=0.6, i.e., the top 60% of the words (277 words total) in the candidate list are used (c.f. Table 1). This threshold produced precision of 88% and recall of 58% meaning that 167 words were added to the ontology of which 147 were correct.

Table 2 reports in more detail on the number of candidate words and how many correct words can be added to the ontology through the text mining process with the document-based selection and restricting our words to nouns only, i.e. the *L1N* test with threshold 0.6 on the validation documents, *Group_B*.

TABLE I. BEST RESULT OF THE TEST L1N (B =0.25)

Threshold	Precision	Recall	F-Measure
0.10	1.00	0.12	0.69
0.20	0.91	0.20	0.76
0.30	0.93	0.31	0.83
0.40	0.91	0.40	0.85
0.50	0.89	0.49	0.85
0.60	**0.88**	**0.58**	**0.85**
0.70	0.84	0.64	0.82
0.80	0.85	0.75	0.84
0.90	0.83	0.82	0.83
1.00	0.81	0.89	0.82

TABLE II. NUMBER OF WORDS CAN BE ADDED

Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
#candidate words	28	55	83	110	139	167	194	222	249	277
# words added	22	50	77	101	124	147	162	188	206	225

TABLE III. VALIDATED RESULT WITH GROUP_B

Threshold	Precision	Recall	F-Measure
0.6	0.77	0.58	0.70

We also observe that the top words extracted using this technique are very relevant to the domain of amphibian ontology, for example, the top 10 words are: *frog, amphibian, yolk, medline, muscle, embryo, abstract, pallium, nerve, membrane.*

To confirm our results, we validated the best performing algorithm, i.e., test case *L1N*, using the 30 previously unused relevant documents in *Group_B*. We applied the document-based selection algorithm using nouns only with a threshold value 0.6. Table 3 presents the achieved results of *P, R* and *F-measure* with threshold *t*=0.6. This shows that, although precision is a bit lower, overall the results are reproducible on a different document collection. In this case 183 words were added to the ontology of which 141 were correct

C. *Discussion*

Our ontology learning framework was empirically tested based on the seed amphibian ontology with retrieved Web documents by using focused crawler. An interactive system of focused crawling was created that allowed us to easily create queries from existing concepts in the ontology and submit them to Web document search engines. This system has returned many good documents since we have only taken top high-ranked search results from trusted search engines (i.e., Google, Yahoo) and through domain restricted queries. The preliminary results of relevant document classification support our hypothesis that we can use SVM to improve the identification of documents suitable for the ontology learning process. In comparison with the baseline method that used all features and produced only 53.93% accuracy, the feature selection methods generally achieve accuracy 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. We have come up to conclude that the results are better with documents retrieved selectively by focused crawling, then filtered through the SVM classification.

For the information extraction using text mining, among four proposed approaches, we got the best results using a vector space approach with the document-based selection and restricting our words to nouns only. Overall, our algorithm produced good accuracy, over than 81% for all cases. If we restrict our candidates to only the top-weighted candidates extracted from the documents, the precision is higher but the recall decreases. In the best case, where the F-measure is maximized, the precision is 88% on the test collection. Our algorithm was also validated with another dataset (i.e. documents in *Group_B*), the precision in this case decreases to 77% which is still acceptable and does not affect significantly to the number and quality of relevant words extracted.

VI. Conclusion

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. Our approach was empirically tested based on the seed amphibian ontology with retrieved Web documents. 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. In the next step, only documents most likely to be relevant are passed along for information extraction.

In comparison with our previous work [1], this paper has added new content and results of the information extraction phase that enables to complete our ontology learning process. Instead of using pattern-based extraction methods, e.g., GATE tool or statistical NLP algorithms, we have applied text mining methods to identify attributes to enrich the ontology. Different experiments of text mining techniques were carried out and the precision of information extraction effectiveness which is 88% has strengthened our belief that this ontology learning process could be used semi-automatically in future to allow experts to get useful information for ontology enrichment.

VII. Future Work

Our main tasks in the future are to validate the focused crawler on a wider range of documents, experiment further with information extraction techniques to get better corpus for ontology enrichment, implement and evaluate a variety of ontology learning methods based on the domain-specific corpus.

Considering the ultimate usability of the text mining approach, it depends on the number and quality of the documents collected by the topic specific spider. In addition, although it extracts good words, these words are not matched with particular concepts within the ontology. A further pairing process, for example a matching process using WordNet vocabulary, is needed to complete the ontology enrichment process.

In future, we hope to combine this text mining approach with the one of lexical expansion using WordNet [15] to exploit the strengths of each. For example we can use WordNet pair the text mining with concepts and use the documents to identify help disambiguate the multiple senses for the concept words found in WordNet. Our other main task is to validate our approach on ontologies from other domains, to confirm that it is domain-independent. Finally, we need to incorporate the results of this work into a complete system to automatically enrich our ontology.

Acknowledgment

This research is partially supported by the NSF grant DBI-0445752: Semi-Automated Construction of an Ontology for Amphibian Morphology.

References

[1] H. Luong, S. Gauch and Q. Wang, "Ontology-based Focused Crawling", International Conference on Information, Process, and Knowledge Management (eKNOW 2009), Cancun, Mexico, Feb. 1-7, 2009, pp. 123-128.

[2] E. Agirre, O. Ausa, E. Havy and D. Martinez, “Enriching Very Large Ontologies Using the WWW”, *ECAI 1st Ontology Learning Workshop*, Berlin, August 2000.

[3] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web”, *In Scientific American,* 2001, pp. 35-43.

[4] P. Buitelaar, P. Cimiano and B. Magnini, “Ontology Learning from Text: Methods, Evaluation and Applications”, *IOS Press (Frontiers in AI and applications*, vol. 123), 2005.

[5] S. Chakrabarti, M. Berg and B. Dom, “Focused crawling: a new approach to topic-specific Web resource discovery”, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Elsevier North-Holland, May 1999, **31**(11-16), pp. 1623 – 1640.

[6] C-C. Chang and C-J. Lin, “LIBSVM : a library for support vector machines”, 2001, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[7] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. “GATE: A framework and graphical development environment for robust NLP tools and applications”, *Proceedings of Computational Linguistics*, 2002.

[8] M. Ester, M. Gross and H.-P. Kriegel, “Focused Web Crawling: A Generic Framework for Specifying the User Interest and for Adaptive Crawling Strategies”, *27th Int. Conf. on Very Large Databases*, Roma, Italy, 2001.

[9] S. Gauch, J. M. Madrid, S. Induri, D. Ravindran, and S. Chadlavada, “KeyConcept: A Conceptual Search Engine”, Center, Technical Report: ITTC-FY2004-TR-8646-37, University of Kansas.

[10] A. Gómez-Pérez, and D. Manzano-Macho, “A survey of ontology learning methods and techniques”. Deliverable 1.5, IST Project IST-2000-29243 - OntoWeb, 2003.

[11] T. Gruber, “Towards principles for the design of ontologies used for knowledge sharing”. Int. J. of Human and Computer Studies, 1994, (43), pp.907–928.

[12] A.H. Tan, “Text mining: The state of the art and the challenges”. In Proceedings of the Pacific Asia Conf on Knowledge Discovery and Data Mining PAKDD'99 workshop on Knowledge Discovery from Advanced Databases, pages 65-70, 1999.

[13] A. Hotho, A. Nürnberger, and G. Paaß, “A Brief Survey of Text Mining”. LDV-Forum, 20(1):19–62, .2005.

[14] T. Joachims, “Text categorization with support vector machines: learning with many relevant features”, *Proceedings of the 10th ECML-1998,* pp. 137–142.

[15] H.Luong, S.Gauch and M.Speretta, “Enriching Concept Descriptions in an Amphibian Ontology with Vocabulary Extracted from WordNet”. The 22nd IEEE Symposium on Computer-Based Medical Systems (CBMS 2009), New Mexico, USA, August 2-4, 2009, pp 1-6.

[16] A.M. Maglia, J.L. Leopold, L.A. Pugener and S. Gauch, “An Anatomical Ontology For Amphibians”, *Pacific Symposium on Biocomputing*, 2007, (12), pp.367-378.

[17] A. Maedche and S. Staab, “Ontology Learning for the Semantic Web”. *IEEE Intelligent Systems, Special Issue on the Semantic Web*, March 2001, **16**(2), pp. 72 - 79.

[18] A. Maedche, G. Neumann and S. Staab, “Bootstrapping an Ontology-Based Information Extraction System”. Studies in Fuzziness and Soft Computing, Intelligent exploration of the web, Springer, 2003, pp.345 – 359.

[19] G-A. Miller, “WordNet: a lexical database for english”, Comm. ACM, Vol 38, No. 11, pp. 39-41, 1995.

[20] B. Omelayenko, “Learning of ontologies for the Web: the analysis of existent approaches”, *Proceedings of the international workshop on Web dynamics*, London, 2001.

[21] M. Shamsfard and A.A. Barforoush, “The State of the Art in Ontology Learning”, *The Knowledge Engineering Review*, Cambridge Univ. Press, 2003, 18(4), pp. 293 – 316.

[22] M. Speretta and S. Gauch, "Using Text Mining to Enrich the Vocabulary of Domain ontologies", 2008 IEEE/WIC/ACM Int. Conference on Web Intelligence, Sydney, Australia, Dec. 9-12, 2008, pp. 549-552.

[23] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami, “Inductive learning algorithms and representations for text categorization”, *Proceedings of CIKM-98,* pp. 148–155.

[24] V. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, 1995.

[25] M.W. Berry, “Survey of Text Mining”, Springer-Verlag New York, Inc., Secaucus, NJ, 2003.

[26] Y.Yang and J.O. Pederson, “A comparative study on feature selection in text categorization”, (*ICML)*, 1997.

[27] Y. Yang, J. Zhang and B. Kisiel, “A scalability analysis of classifiers in text categorization”, *Proceedings of 26th ACM SIGIR Conference*, July-August 2003, pp 96-103.