Approaches to Semantic Markup for Natural Heritage Literature

Hong Cui
School of Information Resources and Library Science, University of Arizona.
hongcui@email.arizona.edu

ABSTRACT
The theme of this paper is the application of automated semantic markup techniques on natural heritage literature to address information needs of taxonomists. Two machine learning based techniques (supervised and unsupervised machine learning) are discussed and compared on a real world corpus. A prototype application that supports batch and online modes of converting free text documents to XML format is described.

Topics
Information organization, Information management, Information technology and services

Keywords
Semantic markup, natural heritage literature, taxonomic descriptions, biodiversity information, biodiversity informatics, supervised machine learning, unsupervised machine learning, XML

INTRODUCTION
Since over two hundred years ago, the work has started to systematically document various living organisms, their habitats, geographic distribution, classification, and identification keys. Current projects of such include the Flora of North American [6] (FNA) and the Flora of China [7](FoC). As a result, a wealth of natural heritage literature has been produced. However, the vast majority of natural heritage literature is still in print format. That in electronic format is mainly in html, pdf, or doc formats, for example, volumes of FNA and FoC are both in print and html formats. Only a small percentage of such information is stored in a structured digital format such as relational database or XML format [2]. To construct the “global commons” of natural heritage literature, as envisioned by the Biodiversity Heritage Library (an international collaboration among major natural heritage institutes, http://www.bhl.si.edu/), a number of digitization projects are being planned or carried out at different levels. Related standard organizations, such as the Biodiversity Information Standards, are working intensively on a number of biodiversity information content/encoding standards, including the Structured Descriptive Data (SDD, an XML application), to meet the immediate needs of digitization projects. For a review of XML schemas for biodiversity literature, see [14].

Natural heritage communities have come to the consensus that the reformatting of unstructured biodiversity literature, especially taxonomic descriptions of living organisms, into a machine readable structured format is necessary for the long term and effective management of the large body of literature.

This reformatting benefits both of the two major usages of the information, namely organism identification and information retrieval. For the attendees of the iConference, structured formats enable more accurate retrieval is probably a well-accepted fact that calls for no more elaboration. We should nonetheless note that the organism descriptions in a good structured format (that allows the encoding of biological characters and their states/values) also support faceted browsing and cross collection (i.e. federated) retrieval.

The specimen identification task is traditionally performed by using taxonomic keys, which are decision trees (upside-down trees with the root node on top) with contrasting statements about a biological character in each node and an identification at the end of each branch. It is worth noting that research has shown that full text information retrieval techniques are not effective for specimen identification tasks [8]. The creation of a good taxonomic key for a group of taxa relies on the completeness of the descriptions and exhaustive search and comparison of the descriptions of related organisms. Any taxonomist can testify that neither of the two conditions is supported well by free-text descriptions: it is easy for an author to leave out some characters, it is tedious to repeat shared characteristics again and again in many descriptions, and on top of these, it is also very time-consuming to search for missing characters and compare many free-text descriptions at one time. A structured format can alleviate these difficulties because it allows an information system to remind an author of the missing elements and to auto-fill the shared characteristics. It is also possible for computers to shoulder some work involved in comparing descriptions, thanks to the structures that are made explicit to computers, enabling more meaningful comparisons element by element, rather than blind keyword matching. If the descriptions are marked-up in very fine details, for example, to the level of characters and character states, it is even possible to automate many aspects of the generation of taxonomic keys.

A sustainable way to transform large sets of free text descriptions to structured formats should not rely on grant moneys and graduate assistants’ manual labor. Automatic semantic markup techniques must be explored to 1) convert newly digitized full text documents into a structured format and 2) allow taxonomists to convert a freshly composed description into the structured format by clicking a button.

RELATED RESEARCH
The majority of work on structuring taxonomic descriptions focuses on the paragraphs with limited semantics, such as taxon names and ranks etc. [9, 10, 13]. Far less research focused on cue-poor yet semantic-rich sections, for example morphological description sections, largely due to the lack of consistency in description contents, as shown by Lydon et al.[11]. Lydon et al manually compared descriptions of the same five species from six
different floras and found large variations in terms of information content and presentation style: only 9% of information was exactly the same, over 55% of information was from a single flora, and around 1% of information from different floras contradicted with each other. Earlier works [16, 1, 17] using syntactic parsing methods to extract information to populate relational databases or to mark up plant descriptions in an XML format have focused on a single collection. Syntactic parsing methods require handcraft grammar rules and extensive lexicons to guide the parsing process. Grammar rules can be collection-dependent, therefore the usefulness of a parser for different collections can be limited. A specialized XML editor for biosystematics literature has also been built [14], however, it requires the users to come up with regular expression rules to mark up documents. [18] extracted plant characters from descriptions of five species taken from six floras, using a hand-made gazetteer as a lookup list to link extracted terms with their semantic labels. It remains to be seen to what extent the manual knowledge engineering scales with larger data sets. On a much larger scale, [4] took a supervised machine learning approach to address the issues of inter-collection variations and automatically marked up descriptions of over ten thousands species from three floras down to sentence/clause level. By reusing domain knowledge learned from other collections in the past, their system (called MARTT) achieved better performance on a new collection, without any re-adjustment or fine tuning of the system. The marked-up taxonomic descriptions are currently available for public access at http://research.sbs.arizona.edu/gs/cgi-bin/library.

TWO MACHINE LEARNING TECHNIQUES

The machine learning approach has a number of advantages over other existing approaches. This approach does not require the markup or extraction rules to be provided to the system. Composing the rules by hand is difficult even for trained knowledge engineers or computer scientists, because one cannot easily tell the overall effect of a rule on a large collection—it may work very well with a small portion of descriptions, but results in a net negative effect if applied on the entire collection. If the goal is to equip domain experts, in this case, taxonomists, with the markup tool, this requirement is simply not reasonable. Taking advantage of ever increasing computational power, the machine learning approach, on the contrary, strives to find global optimum rules automatically.

A properly designed learning system also promotes portability of the technique. It is desirable for a system to be able to process different collections of documents of the same or similar domain, with little or no reengineering. A learning system that learns and makes use of domain knowledge is more portable than a system that learns presentation cues, because it is more likely for the documents to share domain knowledge than the presentation styles, such as font, size, color, or html tags. Since a machine learning system learns markup rules from document collections, the markup rules are always up to date and reflect the characteristics of the current collection. In practice, this works better than relying on hand-crafted rules that may not fit the current task well.

There are two main types of inductive learning techniques: supervised and unsupervised learning techniques. Supervised learning techniques require training examples, from which the learning system derives knowledge, and against which the learning system verifies its learning. Training examples for the MARTT system were marked-up descriptions manually prepared. The number of training examples required by a system varies, depending on the learning algorithm and the heterogeneity of the data collection. Initially the performance of a learning system increases with the number of training examples. At a certain point (typically before 100% accuracy) it typically reaches a performance plateau. When the plateau is reached, the system is said to be trained, as more training examples will not bring in any performance gain. Unsupervised learning techniques do away with training examples; they derive knowledge from the documents themselves by exploring the various regularities embedded in the documents. For either technique, there has to be some source of knowledge that can be exploited and the learning algorithm has to be able to discover something that is previously unknown to it.

3.1 The Supervised Learning Algorithm

Aforementioned MARTT system is a supervised learning system involving two phases: the training phase and the mark-up phase. In the training phase, the algorithm learns markup rules/models from training examples, while in the mark-up phase the trained algorithm uses the learned rules to mark up new descriptions. The algorithm is illustrated in List 1. Details can be found in [3], where the algorithm was labeled as SCCP.

The Training Phase

Name: Training
Input: Training Examples (TEs, XML documents with nested elements)
Output: Populated Learning Hierarchy (LH)
Algorithm:

- Initialize the empty LH with the root node “description”
- Extract elements from TE level by level to save element contents in the corresponding nodes in LH.
- If a node does not exist in the hierarchy of LH, create a new sibling or child node in LH to accommodate the new element at the desired level.
End Foreach

- Foreach node in LH
- CreateModel(“the element contents saved at the node”)
End Foreach

Name: CreateModel
Input: element contents of a node or element
Output: learned model

Algorithm:

- N-Grams = Word-level unigrams, 2-grams, and 3-grams extracted from the leading words of element contents.
- Foreach N-Gram in N-Grams
- support = the number of instances of the element containing the N-Gram / the total number of instances of the element
- confidence = the occurrence of the N-Gram in the element / the total occurrence of the N-Gram
- Save the rule “N-Gram -> element (confidence, support)”
The shortcomings of the existing automated markup systems for biodiversity literature prompted us to ask this research question: To what extent the inherent textual cues in the literature can be exploited to produce XML-based markups that are not biased for frequent features. In other words, is it possible for an unsupervised learning technique to achieve comparable performance as the supervised technique? To what extent the regularities in the text can replace the knowledge manifested in the training examples?

3.2 The Unsupervised Learning Algorithm

We started to answer these questions by investigating a bootstrapping approach [12]. This markup approach is a bottom-up approach, established on the literary warrant principle. It exploits the syntactic cues commonly found in domain corpora. Such cues include: sentences typically start with a subject, the subjects are typically nouns or noun phrases, periods are used to end a sentence, while commas or semicolons are used to set off clauses [5]. These cues were used in the unsupervised learning algorithm that marked up to the level comparable to that of MARTT. List 2 shows the main steps of the algorithm.

Name: Unsupervised Learning Algorithm
Input: To be marked-up documents in text format
Output: Marked-up documents in well-formed XML format
Algorithm:
Use the following simple heuristic rule to find a set of words (Nouns) used as nouns in the documents.

Heuristic rule: A word is used as a noun iff the documents contain its singular and plural forms but no verb forms.

Use English grammar rules to distinguish plural forms from singular forms.

SegmentSet = Segment documents into sentences/clauses (i.e. segments) at the punctuation marks: .,:;
SegmentSet1 = segments starting with a plural noun
SegmentSet2 = segments that do not belong to SegmentSet1
Foreach segment in SegmentSet1
    Extract the word after the first plural noun and save the word in BoundaryWords
End Foreach

(Nouns, BoundaryWords) = Bootstrap(Nouns, BoundaryWords, SegmentSet2)
Foreach segment in SegmentSet1
    If the segment’s subject main noun can be identified
        tag = the subject
    Else
        tag = “unknown”
    End If
    If the segment ends with any punctuation mark but a period
        tag = tag + “_block”
    End If
Use tag to mark up the segment
End Foreach
End

Name: Bootstrap
Input: Nouns, BoundaryWords, SegmentSet2
Output: A bigger set of Nouns, a bigger set of BoundaryWords
Algorithm:
Flag_NewDiscovery = false;
Foreach segment in SegmentSet2
  Generate a subject pattern based on the up to three leading words. For example, given a segment “Sporangiaster rare.”, if “rare” is in BoundaryWords but Sporangiaster is in neither BoundaryWords nor Nouns, the segment’s subject pattern is “?B”.
End Foreach
Sort (Patterns) // patterns with fewer ? are ranked higher
Do
  Foreach pattern in Patterns
    If (new nouns or boundary words are discovered)
      /*note: depending on the pattern. The previous example pattern “?B” would result in a new discovery: Sporangiaster is a noun. A pattern such as “??B” would not result in a new discovery, because it could be a “NNB”, “NBB”, or “MNB” (M for modifier)*/
        Flag_NewDiscovery = true
        update Nouns or BoundaryWords.
    End if
  End Foreach
While Flag_NewDiscovery

List 2: The Unsupervised Learning Algorithm

The details of the algorithm are still being refined, but the main idea is to start with a small set of Nouns and a small set of BoundaryWords (those are learned from SegmentSet1) to discover more Nouns and BoundaryWords, using the basic assumption “subjects (nouns) are followed by boundary words “. The bootstrap function runs in iterations. In each interaction, it tries to make new discoveries and replaces some “?” in the subject patterns with a “B” or “N”, until it fails to make any new discovery in an iteration. The learning process stops there.

The unsupervised learning algorithm has been tested on a small collection of algae descriptions (120 descriptions), two modest collections of plant descriptions (400-633 descriptions), and a modestly large collection of plant descriptions (2000+ descriptions). The performances on larger collections were better. A later section compares the unsupervised and supervised learning algorithm on the 630 FNA descriptions.

THE APPLICATION

We implemented a prototype of an application that supports both of the learning techniques. The user-friendliness of the application lies not only on the employment of sound interface design principles, but more importantly on the reasonable division of the job between domain experts and the computer.

The application does not assume the user has any knowledge engineering skills, such as crafting regular expression patterns for markup, but trusts the user has the ability of identifying misplaced tags. A few screenshots are included here to illustrate the design of the prototype. The application assumes the user either has folders of documents (i.e. batch processing) or will compose a description to mark up. The first screen the user sees once the application is started allows the user to either run a learning algorithm on a folder of documents or to skip the learning step and go directly to the main interface if the user wishes to access other functions (Figure 1). Note for supervised learning, the user needs to provide training examples. The main interface has functions that help with the preparation of training examples (Figure 3).

![Figure 1: The First Screen Allows](image)

If the user chooses to use either of the algorithms to mark up the collection, she will provide necessary information and instruct the application to “Learn to Mark up”. Since the learning and markup process can take a long time depending on the size of the collection, the interface informs the user the progress (Figure 2). If the documents in the “Job Folder” (i.e. the to-be-marked collection) already contain XML tags, the application will automatically assume the user’s intention is to test the performance of the learning algorithm. It will compare the given tags with those generated by the computer and compute performance scores (Figure 2). The performance scores are saved in a file so the user can compare the performances of different runs. Note once the learning is started, the “Learn to Mark up” button is disabled, but the user is still in control where she would like the application to go. Impatient user may cancel the lengthy learning progress and go either to the learning setting screen (Figure 1) to choose a different learning technique, or to the main interface to use other functions.

When the learning and mark up process is done, the user is led to the main interface, where she can verify the correctness of machine-generated markup (using either the supervised or unsupervised approach) by using the “Batch-Mode” functions (Figure 3) and make any corrections by invoking the popup schema menu.

The application maintains the list of files to be examined so the next time the application starts, the user can continue with the remaining of the files. The application automatically records any errors the user corrects, but also allows the user to add any error manually by using “Record an Error” function in “Tools”. A
module will be implemented in the future to pass along the identified errors to the learning algorithms so the algorithms can revise its markup rules. In the “Composing-Mode”, the user is presented a text editor window where she can compose a description, which will be marked up in XML format after the user clicks on the “Mark up” button. This one-click markup relies on the knowledge already learned by MARTT. The user can check for and correct errors in the composing mode in a similar manner as in the batch mode.

As mentioned earlier, the interface also supports the preparation of training examples. The user simply opens a list of examples, highlights certain text segments, and tags the segments by selecting appropriate elements from the popup schema menu. Or the user can use the documents marked-up by the unsupervised algorithm as the training examples for the supervised algorithm.

In short, the application supports the preparation of training examples, two different learning techniques for marking up documents, two different processing modes, the verification and correction of marked up documents, and feedback to the learning algorithms, all in a rather user-friendly manner. An earlier prototype was used by two non-computer science undergraduates and seemed to be straightforward and easy to use. However, being a prototype, there is still room for improvements.

COMPARISONS

We ran a simple experiment to demonstrate the differences between the two learning techniques when applied to semantic markup of taxonomic descriptions. We used 633 descriptions from Flora of North America, manually marked-up according to the aforementioned schema. For the supervised learning algorithm, the descriptions were evenly divided into two sets: training and test sets For the unsupervised learning algorithm, since no training was needed, all descriptions were involved in the bootstrapping.

Neither algorithms were optimized for speed at this time, but it took 2.3 times as long for the supervised learning algorithm to mark up 316 descriptions (28.45 mins) as for the unsupervised learning algorithm to mark up 633 descriptions (12.35 mins) on a Gateway 1.20 GHz Internal Core 2 CPU laptop, with the Java heap size set to 512 MB. The supervised learning algorithm put 96% of all words in the 316 descriptions in correct tags. The unsupervised learning marked 8400 of the 8557 sentences correctly (98%) and identified good tags for 3765 of the 3873 unique subjects. It marked 48 sentences (37 unique subjects) incorrectly and couldn’t make a decision on another 109 sentences (71 unique subjects).

Comparing a description for ARISTOLOCHIACEAE ARISTOLOCHIA serpentaria, marked up using different techniques (Table 1), we see the effects of the learning algorithms. The unsupervised learning produces well-formed XML documents and the tags are generated and applied as the description contents warrant, while supervised learning assumes a generic XML schema no matter the description contents. The effects of this distinction are shown in different ways: the unsupervised learning generally produced more detailed markup, especially in the leaf description, than the supervised learning; the unsupervised learning explicitly marked organs/structures hidden in “other features” element produced by the supervised learning; although not shown in this example, the unsupervised learning is less constrained in dealing with uncommon organization of descriptive contents, for example, describing multiple organs/structures (“leaves and inflorescence”) in one sentence (There were 63 sentences with compound subjects in the 630 FNA descriptions). On the other hand, the XML schema used in supervised learning carries domain knowledge, although implicitly but allowing the translations of
concepts at different levels. In this example, it marked up *herbs* as a type of *plant habit and life style*, and in

<table>
<thead>
<tr>
<th>Supervised Markup</th>
<th>Unsupervised Markup</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;description&gt;</code></td>
<td><code>&lt;description&gt;</code></td>
</tr>
<tr>
<td><code>&lt;plant-habit-and-life-style&gt;</code></td>
<td><code>&lt;herb&gt;</code> Herbs, erect to decumbent, to 0.6 m. &lt;/herb&gt;</td>
</tr>
<tr>
<td><code>&lt;phls-general&gt;</code></td>
<td><code>&lt;leaf_block&gt;</code></td>
</tr>
<tr>
<td><code>Herbs, erect to decumbent, to 0.6 m.</code></td>
<td><code>&lt;leaf&gt;</code> Leaves: &lt;/leaf&gt;</td>
</tr>
<tr>
<td><code>&lt;plant-habit-and-life-style&gt;</code></td>
<td><code>&lt;petiole&gt;</code> petiole 0.5 - 3.5 cm. &lt;/petiole&gt;</td>
</tr>
<tr>
<td><code>&lt;stem-general&gt;</code></td>
<td><code>&lt;blade_block&gt;</code></td>
</tr>
<tr>
<td>Young stem ridged, glabrous to hispid. &lt;/stem-general&gt;</td>
<td><code>&lt;blade&gt;</code> Leaf blade lanceolate to ovate, 5 - 15 * 1 - 5 cm, base truncate to cordate, sinus depth 0 - 1.5 cm, apex acute to acuminate; surfaces abaxially glabrous or hispid; venation pinnate. &lt;/blade&gt;</td>
</tr>
<tr>
<td><code>&lt;stems&gt;</code></td>
<td><code>&lt;surface&gt;</code></td>
</tr>
<tr>
<td>Young stem ridged, glabrous to hispid.</td>
<td><code>&lt;surface&gt;</code> surfaces abaxially glabrous or hispid;</td>
</tr>
<tr>
<td><code>&lt;leaves&gt;</code></td>
<td><code>&lt;other-features&gt;</code></td>
</tr>
<tr>
<td>Leaves; petiole 0.5 - 3.5 cm.</td>
<td><code>&lt;other-features&gt;</code> utricle pendent, pear-shaped to ovoid, 0.5 - 5 cm; syrinx present, ringlike, 1 mm, oblique; tube bent, cylindrical, 1 cm; annulus smooth; limb purplish brown, 3-lobed, lobes 0.5 Å 0.5 cm, glabrous; gynostemium 3-lobed, globose to crown-shaped, 1.5 mm;</td>
</tr>
<tr>
<td><code>&lt;leaf-general&gt;</code></td>
<td><code>&lt;flower_block&gt;</code></td>
</tr>
<tr>
<td>Leaves: petiole 0.5 - 3.5 cm.</td>
<td><code>&lt;flower&gt;</code> Flowers: calyx brown-purple, bent;</td>
</tr>
<tr>
<td><code>&lt;leaf-blade&gt;</code></td>
<td><code>&lt;calyx&gt;</code></td>
</tr>
<tr>
<td>Leaf blade lanceolate to ovate, 5 - 15 * 1 - 5 cm, base truncate to cordate, sinus depth 0 - 1.5 cm, apex acute to acuminate; surfaces abaxially glabrous or hispid; venation pinnate.</td>
<td><code>&lt;calyx&gt;</code> calyx brown_purple, bent;</td>
</tr>
<tr>
<td><code>&lt;flower&gt;</code></td>
<td><code>&lt;tube&gt;</code></td>
</tr>
<tr>
<td>Flowers: calyx brown-purple, bent;</td>
<td><code>&lt;tube&gt;</code> tube bent, cylindrical, 1 cm;</td>
</tr>
<tr>
<td><code>&lt;calyx&gt;</code></td>
<td><code>&lt;annulu&gt;</code></td>
</tr>
<tr>
<td>calyx brown_purple, bent;</td>
<td><code>&lt;annulu&gt;</code> annulus smooth;</td>
</tr>
<tr>
<td><code>&lt;syrinx&gt;</code></td>
<td><code>&lt;limb&gt;</code></td>
</tr>
<tr>
<td>syrinx present, ringlike, 1 mm, oblique;</td>
<td><code>&lt;limb&gt;</code> limb purplish brown, 3 - lobed, lobes 0.5 * 0.5 cm, glabrous;</td>
</tr>
<tr>
<td><code>&lt;tube&gt;</code></td>
<td><code>&lt;gynostemium&gt;</code></td>
</tr>
<tr>
<td>tube bent, cylindrical, 1 cm;</td>
<td><code>&lt;gynostemium&gt;</code> gynostemium 3 - lobed, globose to crown_shaped, 1.5 mm;</td>
</tr>
<tr>
<td><code>&lt;annulu&gt;</code></td>
<td><code>&lt;anther&gt;</code></td>
</tr>
<tr>
<td>annulus smooth;</td>
<td><code>&lt;anther&gt;</code> anthers 6;</td>
</tr>
<tr>
<td><code>&lt;limb&gt;</code></td>
<td><code>&lt;ovary&gt;</code></td>
</tr>
<tr>
<td>limb purplish brown, 3 - lobed, lobes 0.5 * 0.5 cm, glabrous;</td>
<td><code>&lt;ovary&gt;</code> ovary 3-locular, to 1.5 cm.;</td>
</tr>
<tr>
<td><code>&lt;gynostemium&gt;</code></td>
<td><code>&lt;seed&gt;</code></td>
</tr>
<tr>
<td>gynostemium 3 - lobed, globose to crown_shaped, 1.5 mm;</td>
<td><code>Seeds rounded, ovate, 0.5 * 0.4 cm.</code>;</td>
</tr>
<tr>
<td><code>&lt;anther&gt;</code></td>
<td><code>&lt;seed-general&gt;</code></td>
</tr>
<tr>
<td>anthers 6;</td>
<td><code>&lt;seed-general&gt;</code> Seeds rounded, ovate, 0.5 * 0.4 cm.</td>
</tr>
<tr>
<td><code>&lt;ovary&gt;</code></td>
<td><code>&lt;chromosomes&gt;</code></td>
</tr>
<tr>
<td>ovary 3-locular, to 1.5 cm.;</td>
<td><code>2n = 28</code>;</td>
</tr>
<tr>
<td><code>&lt;capsule_block&gt;</code></td>
<td><code>&lt;capsule&gt;</code> Capsule globose, 0.8 - 2 Å 1-2 cm, dehiscence basipetal.;</td>
</tr>
<tr>
<td><code>&lt;valve&gt;</code></td>
<td><code>&lt;valve&gt;</code></td>
</tr>
<tr>
<td>valves 6;</td>
<td><code>&lt;valve&gt;</code> valves 6;</td>
</tr>
<tr>
<td><code>&lt;septum&gt;</code></td>
<td><code>&lt;septum&gt;</code></td>
</tr>
<tr>
<td>septa absent.</td>
<td><code>&lt;septum&gt;</code> septa absent.</td>
</tr>
<tr>
<td><code>&lt;fruit&gt;</code></td>
<td><code>&lt;seed&gt;</code></td>
</tr>
<tr>
<td>Capsule globose, 0.8 - 2 Å 1-2 cm, dehiscence basipetal.;</td>
<td><code>Seeds rounded, ovate, 0.5 * 0.4 cm.</code>;</td>
</tr>
<tr>
<td><code>&lt;seeds&gt;</code></td>
<td><code>&lt;2n&gt;</code></td>
</tr>
<tr>
<td>Seeds rounded, ovate, 0.5 * 0.4 cm.</td>
<td><code>2n = 28.</code>;</td>
</tr>
<tr>
<td><code>&lt;chromosomes&gt;</code></td>
<td><code>&lt;chromosomes&gt;</code></td>
</tr>
<tr>
<td><code>2n = 28.</code>;</td>
<td><code>2n = 28.</code>;</td>
</tr>
</tbody>
</table>

**Table 1: The Comparison of the Same Description Marked-up by Different Learning Algorithms**

effect translated `<herb>` to its broader concept `<plant-habit-and-life-style>`. Similarly it took `<capsule>` as a type of `<fruit>`, and `<2n>` `<chromosome>` counts.

There were also 50 instances of organ/structure descriptions marked as “other features” by the supervised algorithm, including those about staminodial ring, utricle, 1st leaves, glandular hairs, intravaginal squamules, pigment cells, stomates, ring meristem, ostirole, air chambers, leaf primordia, terminal leaflet, abaxial ridges, stranded plants, hastula, cells, sporangiasters, staminate scales, and cystoliths, totally 19 organs/structures. Unsupervised learning, on the other hand, discovered good tags for 14 of these structures and marked 40 of these instances with correct tags.
DISCUSSIONS

What have we learned from the experiment on the randomly selected 633 FNA descriptions? First we learned that the documents from the real world will contain information that is not covered by a predefined XML schema. In the 633 descriptions (a tiny portion of FNA, new volumes of which are still being published), 19 organs/structures were not found in the XML schema. In addition, there are also 63 descriptive sentences with compound subjects, which cannot be tagged as any single organ/structure. We believe a semantic schema, let it be an XML schema or an ontology, is necessary for any semantic markup task to ensure consistent interpretation. The top-down approach to the creation of such a schema, however, does not seem to be a viable approach. In order to discover all organs/structures described in a collection, we believe a bottom-up approach based on the literature warrant principle such as the unsupervised learning method described in this paper is more feasible. To say the least, the unsupervised learning method can be used to generate element candidates and to identify difficult cases ("unknown") for the human experts to base their work on.

Second we saw that the strength and weakness of the two techniques are quite evident. The unsupervised technique is more flexible, more efficient, and capable of uncover new/rare organs/structures. But it has the shortcoming of unable to associate concepts with their broader concepts. For example, it marks different fruit types explicitly yet unable to tell that they share one broader concept "fruit". We often assume that the semantic relationships among domain concepts have long been codified in a thesaurus or something similar. While such assumptions are often unfounded, even if they are, we still face the issue of discrepancies among different thesauri: for example, if you trace the hypernymy relation of the word nut in WordNet, you will find nut is a seed. However, in the Oxford Plant Characters thesaurus, nut's parent term is fruit. In WordNet, a capsule is a seed vessel, but in Oxford it is a fruit. Fortunately, the unsupervised learning algorithm reduced the workload for human experts to make involved associations among concepts from reading X sentences to examine Y distinct tags. In case of the 633 FNA descriptions, X=8557, Y=280.

This identifies an additional component to be implemented as a part of the MARTT Interface application. In addition to support supervised and unsupervised learning, a module is needed to allow a human expert to establish semantic relationships among existing or introduced concepts. This is what we will do next.

7. REFERENCES


