Hong Cui
Faculty of Information and Media Studies, University of Western Ontario
London, Canada

MARTT: A General Approach to Automatic Markup of Taxonomic Descriptions with XML

Abstract: Despite the sub-language nature of taxonomic descriptions of animals and plants, researchers have warned about the existence of large variations among different description collections in terms of information content and its representation. These variations impose a serious threat to the development of automatic tools to structure large volumes of text-based descriptions. This paper presents a general approach to mark up different collections of taxonomic descriptions with XML, using two large-scale floras as examples. The markup system, MARTT, is based on machine learning methods and enhanced by machine learned domain rules and conventions. Experiments show that our simple and efficient machine learning algorithms outperform significantly general purpose algorithms and that rules learned from one flora can be used when marking up a second flora and help to improve the markup performance, especially for elements that have sparse training examples.

Résumé : Malgré la nature de sous-langage des descriptions taxinomiques des animaux et des plantes, les chercheurs reconnaissent l’existence de vastes variations parmi différentes collections de descriptions, en termes de contenu informationnel et de leur représentation. Ces variations présentent une menace sérieuse pour le développement d’outils automatiques pour la structuration de larges volumes de descriptions textuelles. Cet article présente une approche générale pour le balisage de différentes collections de descriptions taxinomiques avec XML, en utilisant comme échantillons deux flores d’envergure. Le système de balisage MARTT est basé sur les méthodes d’apprentissage automatique et est amélioré par les règles et les conventions de l’apprentissage automatique. Les expériences démontrent que notre simple et efficace algorithme d’apprentissage automatique a surclassé de manière significative les algorithmes par objet et que les règles d’apprentissage d’une flore peuvent être utilisées lors du balisage d’une deuxième flore et aider ainsi à améliorer la performance du balisage, notamment pour les éléments qui ont des exemples d’apprentissage peu courants.

1. INTRODUCTION

Taxonomic information organization and access is a major component of biodiversity informatics research. Despite the recent development of taxonomy databases, only “a trivially small amount of descriptive data exists in a structured form that is amenable to manipulation by software”(Blum, 2000). One of the major informatics challenges surrounding taxonomic character data is to develop ways of "mining" or parsing structured data from the existing text-based descriptions. XML (eXtensible Markup Language) is a well-accepted standard for structuring and exchanging textual data. Taxonomic descriptions marked up in XML format can be used to improve the performance of information retrieval systems, to facilitate federation and integration of multiple description collections, and to aid the generation of interactive keys for specimen identifications. Because a huge amount of text-based descriptive data has been developed over the past two hundred and fifty years, a manual approach to markup is not feasible. An automatic markup system can structure the legacy textual descriptions, convert textual descriptions freshly written by taxonomists to XML format on the fly, and at the same time help to ensure the parallelism of characters presented in the descriptions.

Previous efforts in structuring taxonomic descriptions used syntactic parsing method and focused on structuralize single description collection at a time. Taylor (1995) and Abascal and Sánchez (1999) constructed grammars and lexicons to parse the Flora of New South Wales and Flora of North America (FNA, http://www.fna.org) respectively, to extract a set of triples of specimen part, attribute and values from each description. Jean-Marc Vanel's Worldwide Botanical Knowledge Base (http://wwbota.free.fr/) also took the approach of parsing, but aimed to mark up descriptions with XML. None of these works reports their scientific evaluation of system performance.

The published works have all used parsing techniques that require an extensive lexicon. The parsing approach takes advantage of the sublanguage nature of formal floras. Fig 1 shows the taxonomic treatment of *Chamaecyparis lawsoniana* in FNA with different sections marked. Lehrberger (1982) summarized the characteristics a sublanguage possesses, including: limited subject matter; lexical, semantic, and syntactic restrictions; deviant rules of grammar; high frequency of certain constructions; text structure; and use of special symbols. However, research has found a great amount of variations among descriptions of the same taxonomic objects in different floras. To assess the feasibility for automatic processing of botanical legacy data, Lydon et al. (2003) manually compared and contrasted the descriptions of five common species in six English floras. The finding shows that only 9% of information is presented in all six sources in exact format while 55% of information comes from only one source. They also find 1% of information is in conflict among different floras. The remaining 35% of information is represented in different format in different sources, for example, using different terms for the same concepts etc. The authors suggest that automatic processor should take caution and expect to work with different data sets with large variations.

Given the large variations in the data source, the drawbacks of syntactic parsing approach become significant: (1) the dependence on the coverage of the lexicon and hand-crafted rules; (2) more importantly, the limited portability. Due to the large variations of data sets, the parser tailored for one collection will very likely have a significantly reduced performance on a different collection.

In this paper, we report our progress in developing a general approach to automatic markup of taxonomic descriptions with XML. This research differs from others in that it aims to create an evolvable system that is capable of marking up not just a specific flora, but all floras and faunas in English. In previous work (Cui et al, 2002), we created a procedure based on Support Vector Machine algorithm and marked up at paragraph level FNA taxonomic treatments with around 95% accuracy. In this study, deep markup of description paragraphs is the focus. In this paper, we present our findings in testing our markup framework and evaluating our markup system, MARTT (MARkuper for Taxonomic Treatments), with plant description data.

The paper is organized as follows. We will describe the MARTT framework in section 2. We devote section 3 to learning system design. In section 4, we describe a set of three markup algorithms for marking up to sentence/clause level. In section 5, we report experiments of marking up two large plant description collections to main structure level. Finally, in Section 6, we will conclude and describe our future plans.

2. **Chamaecyparis lawsoniana** (A. Murray bis) Parlatore, Ann. Mus. Imp. Fis. Firenze. n.s. 1: 181. [preprint p. 29]. 1864.	taxon
Port-Orford-cedar, ginger-pine	common names
Cupressus lawsoniana A. Murray bis, Edinburgh New Philos. J., ser. 2, 1: 299, plate 10. 1855	naming history
Trees to 50 m; trunk to 3 m diam. Bark reddish brown, l0--20(--25) cm thick, divided into broad, rounded ridges. Branchlet sprays predominantly pinnate. Leaves of branchlets mostly 2--3 mm, apex acute to acuminate, facial leaves frequently separated by paired bases of lateral leaves; glands usually present, linear. Pollen cones 2--4 mm, dark brown; pollen sacs red. Seed cones maturing and opening first year, 8--12 mm broad, glaucous, purplish to reddish brown, not notably resinous; scales 5--9. Seeds 2--4 per scale, 2--5 mm, wing equal to or broader than body. 2n = 22.	description
Forests of the Coast Ranges with isolated inland populations at higher elevations in the Siskiyou Mountains and on Mt. Shasta Forests of the Coast Ranges with isolated inland populations at higher elevations in the Siskiyou Mountains and on Mt. Shasta; 0--1500 m; Calif., Oreg.	distribution
A. J. Rehder (1949) listed, with bibliographic citations, 66 published varieties and forms best considered as cultivars.	discussion

Fig 1 Taxonomic Description of *Chamaecyparis lawsoniana* in FNA

2. MARTT FRAMEWORK

MARTT is based on inductive machine learning methods. An inductive learning algorithm learns the characteristics of data sets by analyzing a training set, which in this case contains plant descriptions marked up by a human expert. Thus the adaptability and portability of the markup system are improved. Due to the sublanguage nature of descriptions, we have good reasons to hope for good performance of learning algorithms. But for marked-up plant descriptions to be truly useful for different information applications, we would like to boost the performance even further. Noticing “the limited subject matter” aspect of floras, we observe the explicitness of domain knowledge conveyed in plant descriptions: by just analyzing the information content of descriptions, one can easily learn that, for example, “herbs do not have trunks”, “conifers do not have flowers”, etc. These rules are true for any flora. Our assumption is that rules like these can help improve the performance of an inductive learning system in marking up taxonomic descriptions. Basing on this observation, we propose a two-phase learning framework to mark up taxonomic descriptions:

Phase 1: Use inductive learning methods to mark up taxonomic descriptions in some large-scale floras, for example FNA. Learn domain-wise rules from marked-up descriptions and save the rules in a rule bank.
Phase 2: Use the inductive learning component, enhanced with access to the rule bank, to markup new descriptions, for example, Flora of China.

Newly marked-up descriptions may be fed back to phase 1 to help evolve the domain rules.

In the following sections, we will show that very good markup performance can be achieved using our simple and efficient learning algorithms for Phase 1. We will also show some promising results of using learnt rules from FNA to improve markup performance on Flora of China (FOC, http://flora.huh.harvard.edu/china/), especially for elements with sparse training data.

3. SYSTEM DESIGN

A top-down iterative strategy is taken to mark up the descriptions, that is, higher level elements are marked up before any lower level elements in each higher element are marked up. The learning system consists of a hierarchy of nodes, each of which corresponds to an XML element. The hierarchy corresponds to the structure of targeted XML markup. Every internal node is a learning unit, in charge of marking up its corresponding element, while leaf nodes only receive their text segments produced by their parents and do not perform learning or mark up. All nodes also evaluate the markup performance of their parent node by comparing the marked up segments with the answer keys. Fig 2 illustrates the hierarchical structure.

The learning hierarchy is initialized from training examples. While the structure is initialized, nodes are created and their portions of training examples are extracted and saved in the nodes. Starting from the root, if a node for an element does not exist in the hierarchy, the node is added as a child node to its parent node. At the same time, the content of the element is added to the new node's training pool. In the end, the hierarchy is initialized with all possible elements in the training data represented as nodes in the hierarchy, where each node contains the chunks of training instances relevant to its task. Nested elements in the training instances are flattened out so that training instances for each internal node are flat 1-level deep XML fragments.

For example, the root node "description" is in charge of marking up the input text with child tags such as "plant-habit-and-life-style", "stems", "leaves" etc. When the root node is done with markup, each marked-up segment is dispatched to its corresponding second level nodes, where the element will be marked up one level deeper if necessary. Each node decides on the approach it should take, text segmentation (to identify large chunks of text, for example, all text that is about flower) or information extraction (to extract small pieces of text, for example, shape of a leaf), by examining its training examples. The content of marked-up elements is dispatched in this manner until they reach the leaf node. Marked up examples are then read off the hierarchy.

The hierarchical structure corresponds naturally to the tree structure of XML documents. Using training examples to initialize the structure makes the learning systems very adaptable to different requirements on the depth of markup: if one needs to train the system to mark up to sentence level, just feed the system with training examples with sentence level markup. It is also easy to focus on the markup of a specific element, as any sub-tree itself is a hierarchy and every node is addressable by using its XPath.

The system is implemented in an object-oriented programming language, JAVA to maximize the flexibility for plugging in new learning algorithms and new applications of the hierarchy. The hierarchical structure is separated from the methods that access the tree to give the flexibility of updating access methods without having to modify the tree in any ways and new learning algorithms can be plugged in with ease.

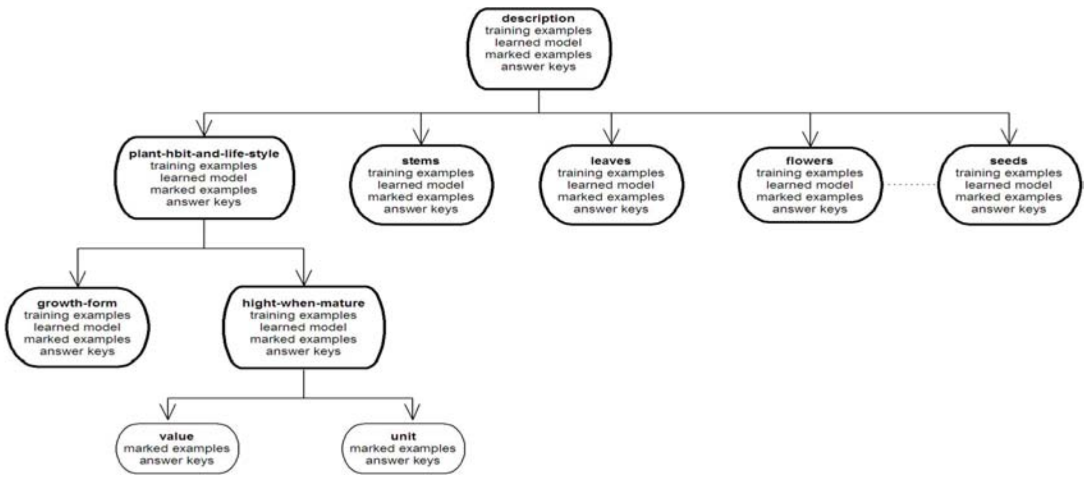

Fig 2 Learning Hierarchy

Nodes with heavy border are composite elements that contain other elements. Nodes with light border are leaf elements that do not contain other elements. All composite elements are associated with certain learning functions and are capable of marking up a text segment using their child elements. Segments that belong to their child elements are dispatched to them respectively. At evaluation phase, answer keys are read into each node, composite or leaf. Evaluation can then be conducted within each node

4. LEARNING ALGORITHMS

SegTagger1: Noticing in descriptions, sentences often start with a noun phrase, which is the name of a plant structure. Based on this observation, we create a simple markup algorithm, SegTagger1, which uses the leading words of sentences to decide the class labels for the sentences. First, training examples are used to score all words that appear in training examples. Each word is scored based on the likelihood it suggests different classes (an XML element is a class), the class score of a word is the ratio of the occurrence of the word in the class and the total occurrence of the word.

Neither stemming nor the change of cases is done for terms. For the taxonomic description domain, morphology of the terms and their cases seem to carry significant meanings. For example, “Seed” often occurs in “Seed cones” as a modifier for “cones”, while “Seeds” often starts the description for the seeds of the plant.

To mark up a new description, SegTagger1 decides class label sentence by sentence. For a sentence, SegTagger1 adds the score for each class for each of *l* leading words. The class scoring highest is the class for the sentence. If all classes are scored zero, the sentence is assumed to have the same class label as the previous sentence.

SegTagger2: SegTagger1 does not make use of information that appears in the body of the sentences. Nor does it try to deal with the cases where one sentence contains descriptions for multiple plant parts. For example, sometimes *stems* and *buds* descriptions are mixed with *plant habit and life style* in one sentence, as in "Shrubs 1.5 m; stems erect." SegTagger2 attempts to tackle these issues and it also tries to make use of the sequence of the classes as they appear in the training examples. SegTagger2 learns class scores for words, punctuation mark scores, and class transition matrix. Word class scores are obtained as in SegTagger1. The maximum likelihood for an element to end with certain punctuation marks is gathered in a similar manner. A transition matrix that describes maximum likelihood that class C_1 is followed by C_2 is also calculated.

When marking up, the algorithm first selects *n* candidate classes by scoring the *l* leading words of to-be-marked text *T* or by looking at the transition matrix when all class scores for *l* are zeroes. Each candidate class proposes a segment from *T* (starting from the beginning of *T*). The candidate classes and their proposed segments are evaluated and given a score. The class and segment pair that scored highest wins and the segment is marked with the class. The algorithm will start again to mark up the remainer of *T*. To propose a segment, a candidate class may initially look at more than one segment using different delimiters that have been seen in training examples. Segment candidates are obtained by using different delimiters to segment *T*. The segment that scored highest for the class is the one the class will propose. This algorithm gives a chance to segment at punctuation marks other than the period (.).

SegTagger3: A second way to improve the SegTagger1 is geared toward learning semantic class for leading words. SegTagger3 learns significant noun phrases and nouns that have clear class indications, for example *Seed cones* and *Roots* are significant indicators for class *cones* and *roots*, respectively. Since there is no good part of speech tagger available for this domain text, SegTagger3 uses frequent pattern and association rule learning methods (originated from data mining research) to learn rules in the format of *n-gram => class (confidence, support)*, where confidence is defined as the ratio of the occurrence of the n-gram and the occurrence of the n-gram in the class; support is defined as the ratio of the occurrence of the n-gram and the number of sentences in the class. Rules whose confidence and support scores are higher than user-defined thresholds are deemed as good rules.

To learn the rules, first the leading *l* words of sentences are used to generate $l(l+1)/2$ different size of n-grams. For example, in addition to one 3-gram *a b c,* leading words *a b c* can generate two 2-grams *a b,* and *b c,* and three 1-grams *a, b,* and *c,* the latter are called sub-grams of *a b c*. The occurrences of these n-grams in different classes are gathered and the confidence and support scores are computed. Note: one occurrence of *a b c* is counted multiple times, one for each of its sub-grams. This is necessary because it is unknown if the n-gram or its sub-grams are the frequent noun phrases. However, this also may cause problems. Suppose *a b c* is a noun phrase and *a b c* has to be together in that order to have the semantic meaning. If *a b c* is a significant indicator for a class, this multiple counting method makes all its sub-grams good indicators for the class as well, when in fact *a, b, c* alone may have very different meaning. To eliminate this side effect, when calculating the scores, if an n-gram is deemed a good indicator, then its occurrence is deducted from the occurrence of all its sub-grams so that its sub-grams will not become frequent because of the n-gram. On the other hand, if an n-gram is just a collocation by chance, its occurrence will not affect the discovery of any of its sub-grams as a significant indicator. The confidence threshold for "good" indicators is set to 0.8 and support threshold is set to 0.035.

SegTagger3 marks up descriptions sentence by sentence as SegTagger1. It decides the class label for a sentence by examining the good association rules (those have confidence and support scores higher than the thresholds: confidence=0.8, support=0) that contain the n-grams generated by its *l* leading words. The rules are sorted and the first rule will be used to decide the class label. The sorting criteria are size of the n-gram, containing starting word, support, and confidence, applied in that order.

5. EXPERIMENTS

In these experiments, the task is to mark up main structures of a plant: *plant-habit-and-life-style, roots, stems, buds, leaves, flowers, pollen, fruits, seeds, cones, spore-related-structures, gametophytes, timelines, chromosomes,* and *compound*. Class *compound* is used to label the cases where more than one main structure is described in one clause, for example "twigs and foliar buds silky-pubescent."

In the first set of experiments, we compare three SegTaggers with a Naïve Bayesian(NB) based learning method and a Support Vector Machines (SVM) classifier and show SegTaggers outperform the two general purpose learning algorithms, with SegTagger3 being the best. We use SegTagger3 to mark up the entire set of FNA descriptions. These experiments show the feasibility of machine learning approach to mark up multiple collections. This activity corresponds to Phase 1 of the MARTT framework.

In the second set of experiments, we first show the markup performance on the entire FNA set is satisfactory. We use the marked up set as the training examples to train SegTagger3 and test it on the original 490 training examples. The results show that higher performance is achieved compared to 10-fold validation on original training examples. Finally and most importantly, we show that semantic information learned from marked-up FNA collections can be used to improve the markup performance of FOC descriptions. This activity corresponds to Phase 2 of the MARTT framework.

Data: We extracted 2,374 descriptions from published volumes (v2, 3, 4, 22, and 23) of FNA. We also have 13,478 descriptions from FOC provided to us from other sources. We prepared 490 training examples for each collection. Table 1 shows the breakdown of the number of training instances for each element for FOC and FNA.

Table 1 Number of Training Instances of Each Class

	plant	stems	leaves	cones	seeds	flowers	fruits	timeline	chrom.	roots	buds	spore	pollen	gameto	comp
FOC	320	374	446	18	149	447	303	348	70	44	23	0	0	0	13
FNA	253	387	458	33	173	320	302	2	270	55	51	90	2	13	1

Performance Measures: Performance evaluation in terms of correctness of markup is quite involved, because it often occurs that some parts of an element are marked up correctly, or that elements are not placed in the exact right place in a branch. The learning hierarchy allows us to do evaluation at any level or any node, which helps to work round the problem. MARTT calculates the precision and recall at each node/element. Precision is defined as the percentage of the segments that are marked as the element by machine are marked as such by human. Recall is defined as the percentage of the segments that are marked as the element by human are marked as such by machine.

The First Set of Experiments--Feasibility of Machine Learning Approach: We compared our algorithm with Naïve Bayesian (NB) algorithm and Support Vector Machine (SVM) algorithm. These are two widely used text classification algorithms. We build a NB-based markup system by replacing SegTagger3's additive class scoring method with NB formula:

$$score_{c,segment} = P(class = c) \sum_{t \in segment} P(term = t \mid class = c),$$

For SVMs, we used Bow Toolkit by McCallum(1996) to test its performance on classifying description sentences. We run 10-fold validation using different algorithms on FOC and FNA training examples. The results are shown in Table 2. Paired T-tests for the differences of the performance between SegTaggers and NB and SVM show our algorithm significantly out-performed NB and SVM algorithms with the significance levels ranging from 0 to 0.067. In terms of computational cost, it takes SVMs 24 hours to finish the ten-fold validation over 490 examples, it takes SegTagger2 15 minutes, and SegTagger1 and SegTagger3 finish the validation in 3 minutes, all run on Windows XP Pentium M 1.3GHz, 512 MB DDR machine.

In general, despite the complexity of SegTagger2, it failed to improve much the markup performance, compared to the simple-minded SegTagger1. *stems* and *buds* are the elements that are frequently embedded in *plant habit and life style* element. The improvement to them is very little or even negative which suggests that using the frequency of ending punctuation marks for elements may not be helpful. Comparing three SegTaggers across all the elements suggests that leading words are the most reliable cues for mark up task, and content words are helpful in some situations but they are more likely to be sources of confusion. SegTagger3 improves the way of selecting and scoring leading words and achieves the best performance overall.

The most difficult elements to mark up seem to be the *compound* element. As any main structure of a plant can be in a *compound* element, none of the word-based learning algorithms does well in identifying this element, because they fail to capture patterns such as "*element1 and element2". cones* and *buds* are difficult elements for FOC and all algorithms have poor performance on them. This is most likely due to the training data for both elements are sparse, and *buds* are sometimes embedded in other elements. Sparseness of training data for some elements is a common problem for mark up task, because of the uneven distribution of the elements in a collection. In the next section, we will discuss a way that helps to deal with this situation.

More importantly, the results show that SegTaggers work relatively equally well on both FNA and FOC descriptions, demonstrating a good portability of the machine learning approach.

Table 2 Performance Comparisons among Different Learning Algorithms

	SegTagger1				SegTagger2				SegTagger3				NB				SVMs			
	FOC		FNA		FOC		FNA		FOC		FNA		FOC		FNA		FOC		FNA	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
plant	99.3	91.5	100	97.6	99.7	97.9	100	99.6	98.1	100	97.7	100	95.2	82.6	98.0	85.8	97.2	97.2	100	82.1
stems	93.4	84.5	91.1	85.3	92.3	85.5	93.7	89.4	95.5	90.7	95.3	90.7	89.4	65.0	89.8	69.1	70.4	84.8	90.5	83.8
leaves	96.7	96.9	96.3	98.8	97.0	95.7	97.5	99.0	99.1	98.0	99.6	98.8	75.6	68.5	73.9	63.7	87.3	73.0	92.9	70.5
flowers	89.2	96.7	97.0	98.5	93.6	100	96.7	98.5	93.5	95.6	99.4	99.1	69.5	65.2	83.2	77.7	87.6	69.1	88.1	82.5
fruits	96.1	91.4	97.9	95.8	95.9	85.4	99.0	97.7	97.4	97.5	100	99.4	89.7	59.1	87.8	79.3	73.5	54.6	97.4	62.3
chrom.	100	98.6	99.6	100	100	98.6	99.6	100	100	98.6	99.6	99.6	96.3	47.5	99.4	71.6	54.3	100	0	0
seeds	97.7	88.4	100	92.8	98.6	90.7	100	94.0	99.2	99.2	100	97.1	91.3	59.3	91.5	68.0	90.5	59.4	100	72.7
buds	80.0	58.0	95.0	86.0	80.0	53.0	94.2	75.0	60.0	36.0	100	88.7	50.0	19.5	95.0	52.4	50.0	33.3	100	42.9
time	100	99.7	0	0	97.0	95.7	0	0	100	99.7	100	100	98.7	71.5	100	100	98.9	97.7	0	0
cones	63.3	71.7	93.0	98.0	70.0	66.7	93.0	93.0	81.7	81.7	100	100	58.3	60.0	63.2	52.7	42.9	42.9	100	62.5
roots	100	78.0	100	95.7	100	79.7	100	82.7	100	94.2	100	97.3	97.5	53.2	100	41.9	83.3	100	100	85.7

spore.			87.3	83.8			80.3	72.2			97.7	95.5			75.0	46.0			100	87.5
pollen			0	0			0	0			100	100			100	100			0	0
gameto.			85.7	85.7			85.7	85.7			71.4	47.6			50.0	34.5			100	66.7
comp.	0	0	0	0	0	0	0	0	57.1	36.2	0	0	0	0	0	0	100	50	0	0

The Second Set of Experiments-Using FNA to Boost FOC Performance: Since SegTagger3 achieve better performance on FNA than FOC overall, we then mark up the 2,374 descriptions from FNA and use the whole set to help to improve the mark up performance for FOC. But first, we would like to show the quality of mark up on the entire set is good by using these descriptions as training data and test SegTagger3 on the original 490 training examples. The result (Table 3) shows a improved performance on almost all elements, compared to original ten-fold validation. This suggests the markup performance over the 2,374 descriptions is satisfactory.

From the entire marked up set of FNA descriptions, we used the SegTagger3's algorithm to learn significant class indicators. We wrap these association rules into an independent component, called rule bank, and provide an interface for querying about the rules. To see if learned class indicators are useful, we revise SegTagger3's markup algorithm as follows: if a good association rule can not be found from training examples, the algorithm will query the rule bank to see if the rule bank has a good rule (confidence=0.9, support=0.01) to use. If so, the rule will be used to decide the class label. Comparing the performance of SegTagger3 and revised SegTagger3, we can see the rule bank is helpful to classes that have few training examples, in this case, *cones* and *buds*. Revised SegTagger3 achieves the best performance on these two elements over all algorithms tested.

6. CONCLUSION

In this paper, we used two sets of experiments to show the feasibility of the MARTT framework to mark up plant descriptions. We show that our learning algorithms work well on both FNA and FOC without any modification to the system, that our learning algorithms are simple and efficient, and that semantics of the word and phrases learned from one flora can be used to improve mark up performance for another flora. So far we have just made use of the semantics of the words, which may be sufficient at main structure or even sentence level of mark up. Other features that may help should also be explored further. Some of the features are 1) the sequence in which elements often appear in descriptions, for example, if description about *leaves* never appears before *stems*, the learning systems should be able to tell "leaf scar" is a description about *stems* not *leaves*. 2) the relationships among plant structures, for example, *cones* and *flowers* should never appear in the same description. 3) the frequent patterns, for example, patterns like "*structure1* and *strucure2*", or "*structure1*, *structure2*, and *structure3*" would be very good indicators for *compound* element. Frequent patterns will play a more important role in deeper levels of mark up.

Table 3: Train on Marked up FNA and Test on Original FNA Training Examples, Using SegTagger3

	FNA-10-fold		FNA trained on entire set	
	P	R	P	R
plant habit	97.7	100	100	100
stems	95.3	90.7	95.8	92.2
leaves	99.6	98.8	100	99.3
flowers	99.4	99.1	100	99.7
fruits	100	99.4	100	100
chrom.	99.6	99.6	99.6	100
seeds	100	97.1	100	97.1
buds	100	88.7	100	90.2
time	100	100	100	100
cones	100	100	100	100
roots	100	97.3	100	96.4
spore.	97.7	95.5	98.9	96.7
pollen	100	100	100	100
gameto.	71.4	47.6	100	100
comp.	0	0	100	100

Table 4: Compare Performance of SegTagger3 with the Revised SegTagger3 on FOC

	SegTagger3		Revised SegTagger3	
	P	R	P	R
plant habit	98.1	100	98.1	100
Stems	95.5	90.7	94.3	91.5
leaves	99.1	98	98.7	97.8
flowers	93.5	95.6	93.3	95.6
fruits	97.4	97.5	97.4	97.5
chromosomes	100	98.6	100	98.6
seeds	99.2	99.2	99.2	99.2
buds	60	36	80	60
timeline	100	99.7	100	99.7
cones	81.7	81.7	85	85
roots	100	94.2	100	94.2
compound	57.1	36.2	57.1	36.2

The MARTT framework eliminates the efforts of making hand-crafted rules and the compilation of lexicons. The performance is expected to improve as the rule bank evolves and grows. In theory, MARTT is generally enough to be used in other similar domains, for example, faunas. In the near future we will use and continue to develop MARTT to mark up the floras to deeper levels and to apply the system to other collections.

REFERENCES

Abascal, R. et al, 1999. X-tract: Structure Extraction from Botanical Textual Descriptions. *Proceeding of the String Processing & Information Retrieval Symposium and International Workshop on Groupware, SPIRE/CRIWG*. pp. 2-7.

Blum, S.D., 2000. An Overview of Biodiversity Informatics. http://www.calacademy.org/research/informatics/sblum/pub/biodiv_informatics.html accessed on April 1, 2004

Borkar, V. et al, 2001. Automatic Segmentation of Text into Structured Records. *ACM SIGMOD 2001*. pp 175-186.

Cui, H. et al., 2002. An approach to automatic classification for information retrieval. *Proceedings of the Joint Conference of Digital Libraries.* Portland, USA, pp. 96-97.

Lehrberger, J., 1982. Automatic Translation and the Concept of Sublanguage. In Kittredge, R. and Lehrberger J. (Ed.), *Sublanguage. Studies of Language in Restricted Semantic Domain.* Berlin/New York: Walter de Gruyter

Lydon, S. et al, 2003. Data Patterns in Multiple Botanical Descriptions: Implications for Automatic Processing of Legacy Data. *Systematics and Biodiversity*. Vol. 1, No. 2, pp.151-157

McCallum, A, K. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow, accessed on April 1, 2004

Taylor, A. 1995. Extracting Knowledge from Biological Descriptions. *Proceedings of 2nd International Conference on Building and Sharing Very Large-Scale Knowledge Bases*. pp114-119.