Biodiversity Informatics, 5, 2008, pp. 20-40

CONVERTING TAXONOMIC DESCRIPTIONS TO NEW DIGITAL FORMATS

HONG CUI[1]

[1] *School of Information Resources and Library Science, University of Arizona, 1515 E. First Street, Tucson, AZ, 85742.* E-mail: mhongcui@email.arizona.edu

Abstract.— The majority of taxonomic descriptions are currently in print format. The majority of digital descriptions are in a format, such as DOC, HTML, or PDF, for human readers. These formats do not convey rich semantics in taxonomic descriptions for computer-aided processing. Newer digital formats, such as XML and RDF, accommodate semantic annotations that allow a computer to process the rich semantics on human's behalf, opening up opportunities for a wide range of innovative usages of taxonomic descriptions, including searching in more precise and flexible ways, integrating morphological, genomic, georeference, or other information, automatically generating taxonomic keys, and knowledge mining and visualizing taxonomic data etc. This paper reports our experience with the development of an automated semantic markup system named MARTT and discusses challenging issues involved. To address these challenging issues, a number of utilities were implemented to make MARTT a more operable system. The utilities can be used to speed up the preparation of training examples for MARTT, to facilitate the creation of more comprehensive annotation schemas, and to predict system performance on a new collection of descriptions. MARTT has been tested on several plant and alga taxonomic publications including Flora of China, Flora of North America, and Flora of North Central Texas.

Key words.—Digital formats, morphological descriptions, semantic markup, supervised machine learning, system evaluation , taxonomic descriptions, unsupervised machine learning, XML.

Taxonomic descriptions of living organisms are a major information resource used by systematists and evolutionary biologists. The majority of such information is in a print or digital format for human readers. On-going and planned digitalization projects such as those initiated by the Global Biodiversity Information Facility (GBIF, 2007) and the Biodiversity Heritage Library (BHL, 2007) will likely increase the volumes of taxonomic descriptions in legacy formats (e.g., DOC, HTML, or PDF). These documents will have to be converted to a new digital format such as XML or RDF to allow for any innovative usages beyond keyword-based search. Due to the scale of the problem, automated means for the conversion must be sought.

Large volumes of taxonomic descriptions, print or digital, have been produced over the past two hundred years. While descriptions created by trained taxonomists are of high quality and provide consistent information in general, there is not a well-defined and well-accepted standard to regulate the content of a description. A manual comparison among the descriptions of five plant species, found in six well-known floras, revealed surprisingly large variations in terms of description content and style (Lydon et al, 2003). Lydon and colleagues found that only 9% of information was exactly the same in six sources, over 55% of information was from a single source, and around 1% of information contradicted information from another source. Besides the large variation, these findings also suggest that descriptions from different collections are mostly complementary to one another.

As Lydon et al. (2003) concluded, any automatic markup software program must take the variation into account to avoid an overly-tailored system that works only on one or a few description collections. In other words, it is highly desirable for a system to be easily portable to a different description collection. Keeping this in mind, we designed and implemented a portable JAVA application called MARTT (MARkuper for Taxonomic Treatments), which has marked-up >15,000 descriptions from three floras (i.e. Flora of North America (FNA, 1993 onwards), Flora of China (FoC, 1994 onwards), and Flora of North Central Texas (Diggs, Lipscomb, & O'Kennon, 1999)) into a

predefined XML format quite successfully without reconfiguring the system.

This paper reports our experience with the development and evaluation of MARTT and discusses a number of challenging issues identified alone the way. The paper is organized as the following: Starting with the design rationale of MARTT, we go on to report a series of experiments involving the aforementioned floras (readers not caring about technical details can safely skip this section without loss of continuity) and summarize the experimental results. The identified challenging issues are then discussed in detail and the utilities implemented as solutions are examined. After a review of relevant research, we conclude the paper with a plan for future research.

SYSTEM DESIGN RATIONALE

The design goal of MARTT was a highly portable system that would work with all professionally prepared taxonomic descriptions in English without having to re-adjust the system on a collection by collection basis. We also designed the system to learn from its experience with well-prepared descriptions, with the hope that it would become capable of tagging less-well-prepared ones (e.g. those created by amateur taxonomists) in the future. More specifically, the system should be able to mark up a plain-text description into an XML document like the one shown in Figure 1. Note the design goal emphasizes more the system's ability of making the semantics of descriptions explicit by inserting appropriate tags than the resultant documents' compliance to an encoding standard. This is because once a description is in XML format, it is easy to convert it to a standard format such as RDF or SDD (Structure of Descriptive Data, an XML standard issued by the Biodiversity Information Standards[1]).

The high portability may be achieved by employing an approach called "supervised machine learning". In this approach, markup rules used to tag description sentences are not hard-coded but learned from examples of descriptions themselves. These examples are called training examples, which are selected descriptions tagged in a desired XML format by human experts according to an XML schema/DTD. A supervised machine learning algorithm examines/learns from training examples to come up with rules that may be used to tag unseen descriptions. Learning from examples affords a flexible system that automatically adjusts its behavior according to the task on hand. For example, if a flora focuses entirely on flowering plants, then the system will not concern itself with tagging seed cones or pollen cones; on the other hand, if only main organ level annotations (i.e. flower, leaf, etc.) are desired and included in the training examples, then the algorithm will gracefully produce markup at that level and not try to insert bract or stamen tags. Since the machine learning approach automatically learns markup rules from training examples, it does not require users to supply any rules. To taxonomists, preparing training examples is much easier than providing markup rules. On the other hand, we do realize that preparing training examples is time-consuming. This is one of the issues we shall address in later sections.

For markup rules to be reusable across collections, they should not be based on text format cues. For example, a rule "the first bold words represent an organ name" is unlikely reusable, as not all collections use bold face for organ names. Instead, the rules should be semantically rich and convey domain knowledge and/or convention, for example, "a berry is a type of fruit". This type of semantic association rules is likely reusable across collections.

Based on all these considerations, MARTT was implemented with three main components. The first component is a machine learning component, which learns markup rules from training examples and applies the rules to tag new descriptions. The second component is a knowledge induction component, which takes a tagged collection to induce semantic association rules from it. The third component is a storage component for the association rules learned over time and is named "the markup rule bank". When enabled, the markup rule bank answers queries initiated by the learning component. An example query may be "(according to the rule bank's knowledge), what could be a good tag for 'Berries fleshy to somewhat leathery'", and the rule bank would likely respond "fruit".

The learning component grows a learning hierarchy on the fly from the given training examples so the hierarchy is always the best fit for the markup task on hand. To illustrate this process,

[1] http://www.tdwg.org/standards/.

let us use the XML description shown in Figure 1 as an example. Initially, the learning hierarchy has one root node "description". When the XML description is read into the root node, the root node sees six elements (i.e. "taxon", "plant-habit-and-life-style", "leaves", "flowers", "fruit", and "seeds") in the description element. The root node thus creates six child nodes, one for each element, and dispatches the content of each element to its corresponding child node. For example the newly created child node "taxon" gets the family and genus names. Each child node then reads the content received and if needed, creates its child nodes to accommodate any new elements, for example, the node "taxon" creates its two child nodes ("family" and "genus"), one for the family element and the other for the genus element. The process continues until a terminal element is reached in each branch. In the process each node saves the content of its corresponding element as part of its training data to be used later. By the end of reading the XML description into the learning hierarchy, a simple learning hierarchy is created and this hierarchy corresponds exactly to the XML structure of the description. Each node in the hierarchy has one piece of training data: the "description" node has the entire description, the "taxon" node has the family and genus names, and the "family" node has the family name, etc. When another training example is read in, the learning hierarchy expands itself to accommodate any new elements not previously seen. Suppose the second training example has a stems element in its description element. When the "description" node checks and sees that it does not have a child node for "stems", it creates one to save the description of the stems there. If there are elements nested in the stems element, the newly created "stems" node creates its child nodes to accommodate those elements. By the time all training examples are read in the learning hierarchy, every element seen in the training examples will have a corresponding node in the hierarchy and the node will have its set of training data. A portion of the learning hierarchy is illustrated in Figure 2.

In addition to its training data, each node in the learning hierarchy is also equipped with a number of learning/markup algorithms. Each node learns how to tag its corresponding segments in a description. When a new description comes, the root node ("description") tags it into segments, such as plant-habit-and-life-style, leaves and stems, and then sends the segments to their corresponding child nodes, where the segments are further tagged. For example, the "leaves" node further tags its segment into pedicel, petiole, stipule, etc. segments. To see if new descriptions are tagged correctly at each node, the hierarchy also reads in and holds answer keys. In other words, each node is capable of calculating its performance scores. Note the disadvantage of this top-down markup strategy is that if an error is made at an upper node, the error is passed down to lower levels. The current implementation of MARTT does not support back tracking of errors.

```
<?xml version="1.0" encoding="ISO8859-1"?>
<description>
<taxon><family>BROMELIACEAE</family>
      <genus>GUZMANIA</genus></taxon>
<plant-habit-and-life-style><phls-general>Herbs,
       usually epiphytic, stemless to
       rarely caulescent.</phls-general></plant-
       habit-and-life-style>
<leaves><leaf-general>Leaves many-ranked,
       usually ligulate;</leaf-general>
       <leaf-blade>blade, margins entire.</leaf-
       blade></leaves>
<flowers><inflorescence-general>Inflorescences 5-
      many-flowered, many-ranked, mostly
       2-pinnate to less commonly single spike,
       flowers laxly to densely
       arranged;</inflorescence-general>
       <bract>floral bracts broad, conspicuous,
       mostly obscuring rachis.</bract>
       <flower-general>Flowers bisexual;</flower-
       general>
       <sepal>sepals distinct to connate over 1/2
       length,usually symmetric;</sepal>
       <petal>petals with claws adherent to
       subconnate petal, forming short tube,
        blade distinct;</petal>
       <stamen>stamens usually included, adherent
       to adnate with petal claws;</stamen>
        <ovary>ovary superior.</ovary></flowers>
<fruit><fruit-general>Capsules cylindric,
       dehiscent.</fruit-general></fruit>
<seeds><seed-general>Seeds with basal, usually
       tan-brown plumose appendage.</seed-
      general></seeds>
</description>
```

Figure 1. An example taxonomic description tagged in XML.

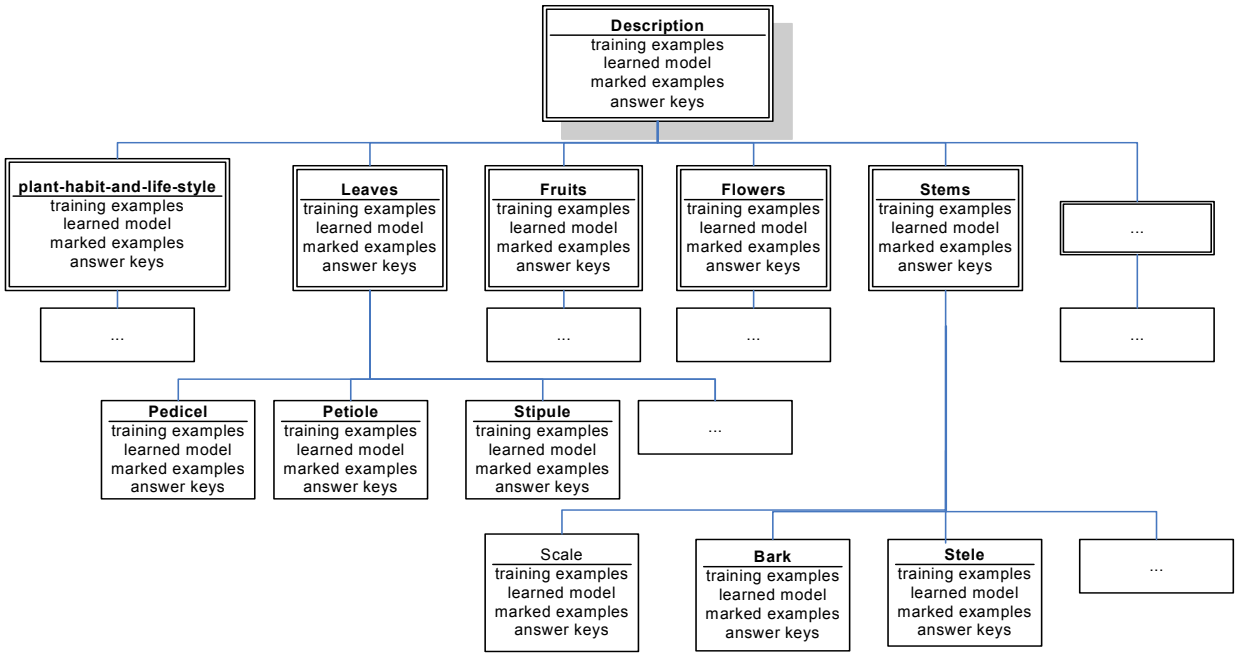


Figure 2. A portion of a learning hierarchy in the learning component. Illustration from Cui & Heidorn (2007) with permission.

Several markup algorithms are available at each node, including a Naïve Bayesian (NB) classifier, Support Vector Machine (SVM) classifier, and a number of homemade algorithms, in order to compare their performance. Once descriptions are segmented into sentences, the task of semantic markup essentially becomes the task of text classification, hence NB and SVMs may be used to assign class labels (i.e. tags) to text segments. For SVMs, we used the implementation in the Bow Toolkit (McCallum, 1996). For NB, we implemented a version based on the algorithm described in Mitchell (1997). Experiments showed that NB and SVMs did not perform as well as some of our homemade algorithms, especially on elements with little training data. The lack of training data makes it difficult for NB to accurately estimate probabilities and for SVMs to identify good support vectors. Details of the learning algorithms and their performance comparison can be found in Cui (2005b) or Cui & Heidorn (2007). The following section describes the best homemade algorithm, SCCP (Semantic Classes and Character Patterns), and reports the performance of MARTT/SCCP on the three floras. Readers not caring about technical details can safely skip *The Machine Learning Algorithm* and *The Experiments with MARTT System* without loss of continuity.

THE MACHINE LEARNING ALGORITHM

SCCP markup algorithm first segments descriptions into sentences and then learns to tag the segments. SCCP segments descriptions by periods (.) and semicolons (;), which are the typical punctuation marks used in taxonomic descriptions to set off semantic units. SCCP uses a set of heuristics to avoid false segmentations at the periods used as a decimal point (e.g., 2.5) or in an abbreviation (such as var., subsp., H. L. James, diam. etc.) or at the semicolons that are part of HTML entities (e.g.,). SCCP does not perform any text normalization procedures such as stemming or converting text to lower case. SCCP does not use a part of speech (POS) tagger to identify nouns or noun phrases because available POS taggers are typically for the general domain and do not work well with taxonomic descriptions due to differences in grammar and lexicons. Instead, SCCP uses a frequent pattern and association rule learning method, originated from data mining research, to learn rules of the form: *n*-gram → element (*confidence*, *support*), which reads "the *n*-gram is associated with the element

with *confidence* (a numerical value) and *support* (a numerical value)". In association rule learning, *confidence* and *support* are a pair of scores measuring the strength of an association. Rules scored higher than a pair of user-defined thresholds areassumed to be good (Han & Kamber, 2000). Adapting from the standard definitions, we define *confidence* as the ratio of the occurrence of an *n*-gram in an element and the total occurrence of the *n*-gram, and *support* as the ratio of the occurrence of the *n*-gram in the element and the number of segments (i.e. sentences) belonging to the element.

SCCP learns the association rules from training examples by first generating sets of *n*-grams and then calculating the confidence and support scores for the candidate rules based on the occurrences of the *n*-grams in different elements. The leading *l* (a user defined variable) words in the sentences are used to generate $\sum_{1\le n\le m} l-n+1$ *n*-grams, where $m \le l$ is another user defined variable that defines the length of the longest *n*-grams. For example, a word sequence "a b c d" with $m = 4$, $l = 4$ generates four unigrams: a, b, c, and d; three bigrams: a b, b c, and c d; two *3*-grams: a b c and b c d; and one *4*-gram: a b c d; totally ten *n*-grams, $1 \le n \le 4$. We call the *m*-grams the "sub-grams" of an *n*-gram when they are generated from the same *n*-word sequence and $m < n$. The generation of *n*-grams of varied sizes creates a pool of noun phrase candidates. These noun phrases and all possible elements form candidate association rules. The strength of the association between an *n*-gram and an element is evaluated by the confidence and support scores, calculated from the occurrences of the *n*-gram in different elements in the training examples. Note under this scheme, sub-grams inherit the occurrence counts of their *n*-grams. This causes undesirable consequences in some cases. Suppose the *n*-gram "Seed cones" occurs very frequently in the "seed cones" element and is recognized as a significant indicator of the element, the counting method automatically makes all its sub-grams (i.e. "Seed" and "cones") good indicators of the element as well, while in fact they are not (e.g. "Seed" should be an indicator of the "seeds" element). To avoid this problem, the sub-grams are not allowed to inherit its *n*-gram's occurrence count when the confidence and support scores of the *n*-gram are greater than a pair of pre-set values (meaning the *n*-gram is likely a phrase and should be treated as one semantic unit). The pair of pre-set values should not be confused with the confidence/support thresholds for the association rules. The former values are set lower than the latter and they serve different purposes as described above. In the experiments reported below, we empirically set $l = m = 3$, the pre-set value pair was set to 0.7 for confidence and 0 for support, and the confidence threshold was set to 0.8 and support threshold was set to 0.035. Settings close to these seemed to produce very similar performance.

To mark up a new example, SCCP segments the text and takes the first *l* words of the segments to generate *n*-grams, $1 \le n \le l$. For each segment, by looking up the *n*-grams in the list of association rules learned earlier, SCCP obtains a number of matching rules with confidence and support scores above the thresholds. The matching rules are ranked according to the following criteria applied in this order: the length of the *n*-gram (i.e., *n*), the location of the *n*-gram in the segment, the support score, and the confidence score. Rules containing longer *n*-grams are ranked higher. Rules matching *n*-grams closer to the beginning of the segment are ranked higher. The support score takes priority over the confidence score to favor the rules with more frequent *n*-grams. The top ranked rule determines the tag for the segment.

SCCP is also designed to recognize simple character patterns of the elements containing no words. The current version has only one such pattern for recognizing chromosome counts which take a form like "2n = 24" or "x = 12" in description text.

EXPERIMENTS WITH MARTT

The data sets for the experiments were taken from the published volumes of Flora of China (FoC), Flora of North America (FNA), and the monograph of Flora of North Central Texas (FNCT) with permission. Three sets of training examples were manually prepared, including 378 examples selected from 12,000 FoC descriptions, 310 from 1300 FNA descriptions, and 378 from 1200 FNCT descriptions.The tags used in the training examples and the resultant XML documents, such as "plant habit and life style", were defined in an XML schema (Cui, 2005a). The schema was a result of consulting a number of sources, including a plant systematics textbook

(Radford, 1986), the DELTA format (Dallwitz, 1980), and a plant taxonomist.

The standard 10-fold cross-validation protocol routinely used to evaluate performance of a machine learning system was used to obtain the performance scores of MARTT. According to this protocol, each set of training examples was divided into ten equal-sized subsets. In each run, MARTT used nine subsets to learn markup rules and then tested the markup rules on the tenth subset. The ten subsets allowed for ten such runs, each with a different test set. The average performance over the ten runs was recorded as the final performance score on a collection.

The soundness and completeness of the markup produced by MARTT were measured element by element (i.e., node by node). The soundness was measured by precision (p), which was defined as the ratio of the text segments tagged as an element e correctly and the total segments tagged as e by the algorithm. The completeness was measured by recall (r), which was defined as the ratio of the text segments tagged as e by the algorithm correctly and the total e segments in the collection. The harmonic mean of recall and precision, F-measure = $2pr / (p + r)$, was then calculated. Precision, recall and F-measure are standard measures routinely used to evaluate performance of information retrieval systems. These measures were borrowed to measure the soundness and completeness of tag assignments.

The performance of MARTT on the main organ level markup on each training set using SCCP learning and markup algorithm is shown in Table 1. The performance on each flora is displayed element by element with four columns: the number of examples (N), precision (P), recall (R), and F-measure (F). Note the "taxon" element shown in Figure 1 was a result of a straightforward parsing of the text and was not involved in the learning process. Blanks (i.e. no data) in Table 1 were due to the variations in the descriptions, for example, FNCT descriptions include discussions about the taxa, while FNA and FoC do not. The overall performance across all elements is a weighted average of recalls on N, indicating the percentage of correctly tagged segments. Without any reconfiguration but relying solely on training examples, MARTT marked 94-98% of segments correctly on different collections (Table 1). MARTT then used SCCP and its learned rules to tag the entire collections of FNA and FoC to build the markup rule bank. Finally, MARTT performance on FNCT using the rule bank in different ways was compared with the performance without using the rule bank. These results are shown in Table 2.

Table 2 shows the performance of MARTT on FNCT with three different settings: the first was the normal training and learning process done by SCCP, the second used the rule bank alone without SCCP learning from the training examples, and the third used both—MARTT first queried the rule bank, if no good rule was returned, it used the rules SCCP learned from the training examples. In other words, in this setting, the rule bank was used as the primary knowledge source while the training data was secondary. The results show higher precision scores when the rule bank alone is used, suggesting the rules learned from FNA and FoC are in general highly reliable and applicable on FNCT. One exception here is the discussion element. This is due to the fact that FNA and FOC do not include any discussions in descriptions (see Table 1, N column), so nothing about discussion can be learned from FNA or FOC. MARTT assumed that segments that did not belong to any other elements were discussion, resulting in a high recall (98%) yet a low precision (58%). The other exception is on phenology element. FNA contains little information on phenology. In FoC, all phenology elements start either with "Fl." for flowering time or "Fr." for fruiting time, while FNCT uses normal English to describe when a plant gives flowers or fruits. Thus the rules learned from FoC do not apply to FNCT. The lower recall scores (especially on flowers, only 0.34) are due to the limited coverage of the rules—which were learned from only two other floras (the published volumes only). Overall, the rule bank alone tagged 69% of all segments from FNCT correctly. The correct ratio of using training examples alone was 94%. When the rule bank and the training are combined, the overall performance is improved from 94% to 95%—the rule bank helped to correct 1/6 of the errors made by SCCP. More interestingly, when MARTT used the training examples as the primary knowledge source and the rule bank secondary, the performance improvement was not that obvious, suggesting the rule bank was a more reliable source than the training examples, even though the rule bank was created from other collections.

Table 1: MARTT Performance in Precision, Recall, and F-measure on FNA, FoC, and FNCT Using SCCP

	FNA-N	*P*	*R*	*F*	*FoC-N*	*P*	*R*	*F*	*FNCT-N*	*P*	*R*	*F*
plant habit and life style	202	0.98	1.00	0.99	241	0.99	0.99	0.99	298	0.94	0.90	0.92
Roots	28	1.00	0.94	0.97	30	0.95	0.90	0.92	6	0.83	0.72	0.77
Buds	21	0.97	0.93	0.95	11	0.87	0.95	0.91	4	0.50	0.50	0.50
Stems	230	0.92	0.98	0.95	278	0.92	0.97	0.94	111	0.92	0.91	0.91
Leaves	296	0.99	0.98	0.98	343	0.97	0.98	0.98	270	0.93	0.94	0.93
Flowers	198	1.00	0.99	0.99	345	0.99	0.99	0.99	307	0.94	0.94	0.94
Fruit	192	0.98	0.96	0.97	233	0.98	0.96	0.97	178	0.94	0.88	0.91
Cones	20	0.98	0.96	0.97	14	0.97	0.95	0.96	3	0.89	0.78	0.83
Seeds	119	1.00	0.98	0.99	115	0.98	0.98	0.98	31	0.99	0.97	0.98
spore-related structures	68	0.97	0.96	0.96					7	0.57	0.50	0.53
gametophyte	19	1.00	0.96	0.98								
chromosomes	191	0.97	0.89	0.93	53	1.00	1.00	1.00	3	1.00	1.00	1.00
phenology					269	1.00	1.00	1.00	234	0.97	0.98	0.97
Discussion									638	0.95	0.97	0.96
Total	1584				1932				2090			
Overall			0.97				0.98				0.94	

Further markup to the sub-organ level involves more than 240 elements. The element-by-element performance scores are shown in the appendix. In the appendix the hierarchical relations between elements are denoted by "/". "phls/leaves" may seem strange, but this was how some descriptions had been written. MARTT made no attempt to rearrange original descriptions. The results suggest that at this markup granularity, there are more cases of *other features* element to accommodate sub-organs not covered by the XML schema. Further, variations in element distributions across collections and within collections are more evident. The data also show that many elements have only one training example, which inevitably results in zero performance, because in a 10-fold cross-validation, the training example is either placed in the training set, leaving no test data, or in the test set, leaving no training data. Excluding these elements, the overall markup performance at this level is 91% for FNA, 94% for FoC, and 89% for FNCT (this figure drops to 87% if *discussion* element is excluded). The overall performance is 1% lower if these elements are counted. The calculation of the overall performance only involves the terminal elements and not their parent organ elements.

We evaluated the reusability of the rule bank at sub-organ level markup as well, but limited the evaluation in *stems*, *leaves*, *flowers*, and *fruit* four elements since other main organ elements in FNCT either do not have enough examples (e.g., *roots, buds, cones, spore-related structures,* and *chromosomes*), or do not have a counterpart in FNA or FoC (e.g., *discussion*), or do not have a good number of sub-elements (e.g., *plant habit and life style, seeds,* and *phenology*) to make the evaluation interesting (see the appendix). The results of the evaluation in *stems, leaves, fruits,* and *flowers* elements are shown in Table 3-6 respectively. Improved performance scores (compared to "training alone") are highlighted in the tables.

The results show that the sub-organ level markup in *stems*, *leaves*, and *fruit* elements benefits from the rule bank—using rule bank alone achieved about the same level of performance as that using hundreds of training examples. Combining the rule bank and the training, the performance was further improved.

However, for *flowers* element, the rule bank alone only marked 29% of the segments correctly. This is not entirely supervising because 1) The flower is the most complex organ of a flowering plant. 2) FNCT contained descriptions of grass families and hence had specific sub-organs of grass

Table 2: MARTT Performance in Precision, Recall, and F-measure on FNCT W / W/O the Rule Bank

FNCT		*Training alone*			*Rule bank alone*			*Rule bank + training*		
	N	P	R	F	P	R	F	P	R	F
plant habit and life style	298	0.94	0.90	0.92	0.96	0.63	0.76	0.93	0.91	0.92
Roots	6	0.83	0.72	0.77	0.83	0.89	0.86	0.83	0.89	0.86
Buds	4	0.50	0.50	0.50	0.75	0.75	0.75	0.75	0.75	0.75
Stems	111	0.92	0.91	0.91	0.94	0.88	0.91	0.89	0.97	0.93
Leaves	270	0.93	0.94	0.93	0.98	0.84	0.90	0.94	0.95	0.94
flowers	307	0.94	0.94	0.94	0.99	0.34	0.51	0.96	0.92	0.94
Fruit	178	0.94	0.88	0.91	0.98	0.83	0.90	0.93	0.90	0.92
Cones	3	0.89	0.78	0.83	0.92	0.83	0.87	0.93	0.89	0.91
Seeds	31	0.99	0.97	0.98	0.95	1.00	0.98	0.95	1.00	0.98
spore-related structures	7	0.57	0.50	0.53	0.86	0.74	0.79	0.86	0.74	0.79
chromosomes	3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
phenology	234	0.97	0.98	0.97	0.00	0.00	0.00	0.97	0.98	0.98
discussion	638	0.95	0.97	0.96	0.58	0.98	0.73	0.95	0.97	0.96
Total	2090									
overall			0.94			0.69			0.95	

flowers, such as pappus, ligule, glume, lemma, and palea etc, while FNA and FoC collections did not. 3) The recall on the *flowers* element was as low as 34% (see Table 2). If a segment is not correctly identified as *flowers*, the further markup of its sub-organs cannot be correct because of the hierarchical markup strategy. Despite the overall low performance in *flowers*, the rule bank did help to improve the performance on some of its sub-elements (Table 5).

SUMMARY OF MARTT EXPERIMENTS

The experiments with MARTT show that the machine learning approach is highly portable: on all three floras MARTT achieved very good performance (in the range of 87% to 98%, depending on the markup granularity and data collection). Biodiversity and other factors contribute to the rather skewed distributions of elements in description collections (see the appendix). MARTT fails at many elements with no or few training data. On the other hand, the results suggest that the induced knowledge (i.e. the rule bank) is reliable and reusable, in some circumstances, the rule bank provides more reliable rules than the training examples do. The rule bank is shown to help to improve the markup performance on elements with good coverage. Continuing to enrich the rule bank with the markup rules learned from other description collections is likely to improve its coverage and make the rule bank more powerful. Overall, the experiments showed that MARTT achieved its goal on portability and performance.

Using the learned rules, MARTT tagged all the 15,000 descriptions into XML format and turned them into three Greenstone collections which can be searched by element[2] (Witten et. al. 2000) is an open source digital library software which supports search in specified elements, such as in leaves element. If the collections are tagged according to one schema, like what we have done with FoC, FNA, and FNCT, Greenstone also supports cross-collection search.

The experiments with MARTT and the three floras also identified a number of issues calling for further research, including the issues surrounding training examples, schema coverage, and performance variations. We shall discuss these issues and our current solutions in detail next.

[2] http://research.sbs.arizona.edu/gs/cgi-bin/library.Greenstone.

Table 3: MARTT performance in precision, recall, and F-measure in stems with and without the rule bank.

Stems	*Training alone*				*Rule bank alone*			*Rule bank+Training*		
	N	P	R	F	P	R	F	P	R	F
stem-general	97	0.88	0.89	0.89	0.96	0.94	0.95	0.90	0.94	0.92
bark	3	0.67	0.67	0.67	0.33	0.33	0.33	0.33	0.33	0.33
node	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
culm	7	0.80	0.80	0.80	0.20	0.20	0.20	1.00	1.00	1.00
twig	2	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00
branch	2	0.00	0.00	0.00	0.67	1.00	0.80	0.67	1.00	0.80
branchlet	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
compound	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
overall	114		0.84			0.84			0.91	

Table 4: MARTT Performance in precision, recall, and F-measure in leaves with and without the rule bank.

Leaves	*Training alone*				*Rule bank alone*			*Rule bank+training*		
	N	P	R	F	P	R	F	P	R	F
leaf-general	206	0.92	0.96	0.94	0.97	0.96	0.97	0.95	0.97	0.96
petiole	18	0.72	0.72	0.72	0.89	0.83	0.86	0.83	0.83	0.83
stipule	10	1.00	0.94	0.97	0.00	0.00	0.00	1.00	0.94	0.97
sheath	9	0.79	0.71	0.75	0.00	0.00	0.00	0.79	0.79	0.79
leaf-blade	77	0.95	0.75	0.83	0.95	0.73	0.83	0.90	0.73	0.81
leaflet-general	32	0.91	0.80	0.85	1.00	0.93	0.96	0.97	0.94	0.95
spine	9	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00
tendril	3	1.00	1.00	1.00	0.33	0.33	0.33	1.00	1.00	1.00
ligule	11	0.71	0.79	0.75	0.00	0.00	0.00	0.79	0.86	0.82
gland	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
compound	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
overall	379		0.87			0.79			0.90	

THE TRAINING EXAMPLE ISSUE

The training example problem has two aspects: one has to do with the effort required to prepare training examples and the second is about the skewed distribution of training data in different elements.

Manually inserting tags in hundreds of documents is time-consuming and error-prone. To alleviate this problem, we developed a user-friendly utility that makes use of the rule bank induced from the FoC, FNA, and FNCT collections to automate the training example preparation process. Some screenshots of the interface are shown in Figure 3. Figure 3a shows a text description in the editing area. A click on the "Mark up" button on the tool bar invokes MARTT to tag the description using the rule bank, which essentially tags every clause in the description as shown in Figure 3b. In Figure 3b, the hierarchy in the left pane displays the element structure of the tagged description. If a wrong tag is inserted by MARTT, the user can easily correct the error by bringing up the tag menu with a right-click on the mouse. The identified errors are saved automatically by the utility for further analyses. Because of the shared domain knowledge across plant taxonomic descriptions, the rule bank can mark a large portion of a description with good tags, saving a significant amount of manual effort.

Table 5: MARTT Performance in precision, recall, and F-measure in fruits with and without the rule bank.

Fruits		*Training alone*			*Rule bank alone*			*Rule bank+Training*		
	N	P	R	F	P	R	F	P	R	F
fruit-general	176	0.94	0.94	0.94	0.98	0.94	0.96	0.97	0.97	0.97
infructescence-general	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
pedicel	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mericarp	2	0.50	0.50	0.50	0.00	0.00	0.00	1.00	1.00	1.00
beak	4	0.00	0.00	0.00	0.33	0.33	0.33	1.00	1.00	1.00
wing	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
pappus	12	0.22	0.28	0.25	0.00	0.00	0.00	0.00	0.00	0.00
style	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
other-features	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
overall	202		0.84			0.82			0.87	

For taxonomic descriptions that do not have a corresponding rule bank in MARTT (e.g., ant descriptions or alga descriptions), the utility has another feature to help with the manual markup as shown in Figure 3c, where a selected text segment can be tagged with a tag chosen from the pop-up tag menu, which is populated from a specified XML schema (Cui, 2005a). This interface ensures a tagged example is valid or at least well-formed.

The second issue related to training examples has to do with the unbalanced distribution of elements. In description collections, due to the diversities in living organisms, authorship, and editorial policies, the coverage of different organs are quite uneven, resulting in a very skewed distribution of training data for individual elements: for example, in the 310 FNA training examples, there were more than two hundred examples for "leaf blade" but zero for "tendril". The training data distribution (see the appendix, N column) shows many sub-organs with zero or one examples. There were 42 elements from FNA training examples, 34 from FoC, and 20 from FNCT with only one example, making learning impossible for SCCP. This problem is somewhat alleviated by the induced knowledge from other collections (i.e. the rule bank), for example the markup rules learned from the several examples of "tendril" in FoC and FNCT training examples can be applied to FNA descriptions. But we also investigated an unsupervised approach that would address this issue in a more direct manner, since no training examples are required at all. Because this approach also helps to make rare organs more visible in the XML schema, we shall explain the unsupervised learning approach in detail in the next section.

THE SCHEMA COVERAGE ISSUE

Even though the XML schema (Cui, 2005a) we created for the MARTT experiments was quite comprehensive to start with, there were occasions when we had to edit the schema to include new (sub)organs discovered from the training examples. We also had to use the *other-features* elements to accommodate any uncovered organs remaining in the collections (see the appendix for the occurrences of *other-features* elements). Since a standard list covering all organs of living organisms does not exist, it is often difficult to enumerate in an XML schema all organs described in a sizeable collection. It is more difficult to create a comprehensive XML schema for multiple description collections. Although it is not always necessary to formalize organ names at the schema level (e.g., SDD does not), from the application's perspective, the need to tag all organs described in a collection and the need to search across collections basing on a common schema call for explicit declaration of all organ names. In absence of a comprehensive dictionary covering all organs, a simple way to discover them from collections of descriptions is needed in order to build a complete schema incrementally. In addition, the method can be used by MARTT to address the lack of training examples problem, because it can identify organ names without any training examples.

Table 6: MARTT performance in precision, recall, and F-measure in flowers with and without the rule bank.

Flowers		*Training alone*			*Rule bank alone*			*Rule bank+Training*		
	N	P	R	F	P	R	F	P	R	F
inflorescence-general	187	0.84	0.82	0.83	1.00	0.13	0.23	0.89	0.65	0.75
bract	35	0.81	0.73	0.77	0.20	0.05	0.08	0.90	0.79	0.84
peduncle	4	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00
scape	3	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00
pedicel	16	0.89	0.92	0.90	0.00	0.00	0.00	0.89	0.92	0.90
rachis	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
rachilla	2	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00
branch	5	0.75	0.63	0.68	0.00	0.00	0.00	0.00	0.00	0.00
involucre	9	1.00	0.92	0.96	0.00	0.00	0.00	1.00	0.92	0.96
flower-general	132	0.86	0.90	0.88	0.76	0.91	0.83	0.73	0.94	0.82
perianth	24	0.81	0.82	0.81	0.10	0.05	0.07	0.90	0.88	0.89
corolla	96	0.95	0.93	0.94	0.30	0.05	0.08	0.98	0.93	0.95
corona	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
pappus	15	0.24	0.28	0.26	0.00	0.00	0.00	0.24	0.28	0.26
ligule	7	0.70	0.60	0.65	0.00	0.00	0.00	0.80	0.60	0.69
calyx	40	0.85	0.86	0.86	0.80	0.28	0.41	0.90	0.86	0.88
glume	11	1.00	0.86	0.92	0.00	0.00	0.00	1.00	0.86	0.92
lemma	24	0.88	0.93	0.91	0.00	0.00	0.00	0.88	0.93	0.91
palea	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sepal	19	0.80	0.77	0.78	0.70	0.43	0.54	0.93	1.00	0.96
petal	59	0.94	0.93	0.94	0.90	0.46	0.61	0.97	0.94	0.96
tepal	2	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00
lip	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
hood	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
carpel	6	1.00	1.00	1.00	0.80	0.80	0.80	1.00	1.00	1.00
anther	9	1.00	0.83	0.91	1.00	0.92	0.96	1.00	0.92	0.96
style	15	0.96	1.00	0.98	0.14	0.14	0.14	0.96	1.00	0.98
stamen	38	0.97	0.98	0.98	1.00	0.72	0.84	0.97	0.98	0.98
pistil	6	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00
stigma	10	0.94	0.92	0.93	0.00	0.00	0.00	0.94	0.92	0.93
filament	6	0.60	0.50	0.55	0.40	0.40	0.40	0.40	0.40	0.40
ovary	12	1.00	0.86	0.93	0.00	0.00	0.00	1.00	0.86	0.93
placenta	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
receptacle	3	0.67	0.67	0.67	0.00	0.00	0.00	0.67	0.67	0.67
gynostegium	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
hypanthium	3	1.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	1.00
keel	2	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50	0.50
pollen	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
nectary	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
gland	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
compound	10	0.67	0.50	0.57	0.00	0.00	0.00	0.67	0.50	0.57
other-features	10	0.29	0.19	0.23	0.00	0.00	0.00	0.29	0.19	0.23
overall	835		0.83			0.29			0.81	

To this end we developed a utility that simply take a collection of descriptions to generate a draft XML schema, which contains the names of the organs described in the collection. The utility employs an unsupervised machine learning algorithm that takes advantage of the formality in professionally prepared descriptions. In particular, we notice that. Collectively, subjects of sentences in descriptions likely represent the complete set of organs described. The algorithm tries to separate subjects from remaining parts of sentences, and then collects organ names from the subjects and organ characters from the remaining parts for future markup at a finer granularity. Being an unsupervised algorithm, this algorithm does not need any training examples. Making use the organ names and the regularity in punctuation usage in the descriptions, the utility generates a raw but rather comprehensive XML schema that can be easily refined by a domain expert.

Here is how the unsupervised algorithm works on a collection: Plain-text descriptions in the collection are segmented into sentences at full stops or semicolons. The algorithm makes the first three leading words of the sentences candidate subjects so no potential organ names is left out. Next it finds nouns from the description collection in question by using the following heuristic rule: a word *w* is noun, *iff* the collection contains singular and plural forms of *w*, but no past, past participle, or present participle forms. Seed nouns (nouns given to the algorithm are called seed nouns) may also be provided by the user directly or collected from a glossary. With the list of nouns, the algorithm marks the words in the candidate subjects as either *noun* or *unknown*. Then, all the sentences in the collection are sorted according to the number of known nouns in their candidate subjects. Next, the algorithm uses the following bootstrap procedure to infer the roles of the *unknown* words.

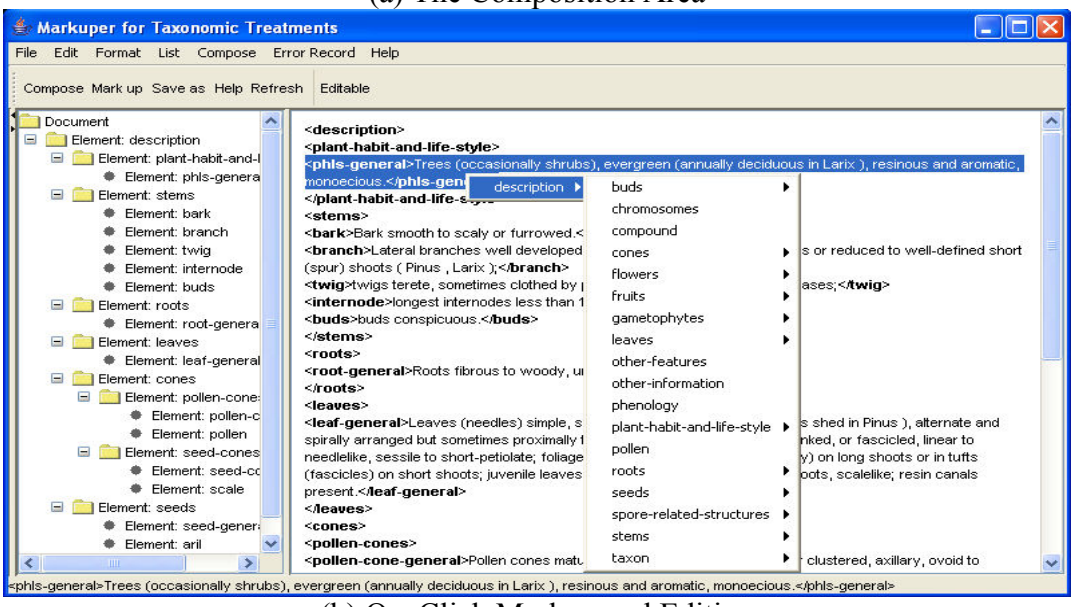
The bootstrap procedure works in iterations and stops when no new discoveries are made in an iteration. New discoveries are used immediately in the next iteration to make other discoveries. A discovery is an identification of either a modifier – the word before a head noun (e.g. "basal" in "basal leaf"), a boundary word – the word following a head noun (e.g 2 in "cells 2"), or a noun. When the bootstrap procedure terminates, the algorithm uses the discovered modifiers, nouns, and boundary words to verify the candidate subjects: a verified subject is a noun with or without modifiers and is followed by a boundary word. If a subject can not be verified, the algorithm takes all the words up to the first known noun (inclusive) in the sentence as the subject.

When the roles of the words in the subjects are known, it is straightforward to group different subjects to their head nouns, for example, "pistillate flowers" and "staminate flowers" are "flowers". This in effect identifies an "is type of" relationship between the three concepts: pistillate flowers and staminate flowers are types of flowers. The relationship "is part of" may also be discovered by looking at the punctuation marks. Many floras adopt the convention to "place each major structure in a separate sentence and separate subparts by semicolons" (FNA Editorial Committee, 2006). This convention can be used to identify relationships such as sepals are a part of a flower. These relationships are integrated in the resultant raw schema, which is a good start for a domain expert to make refinements. The organ names and relationships will also be useful for a semantically richer ontology to be developed in the future.

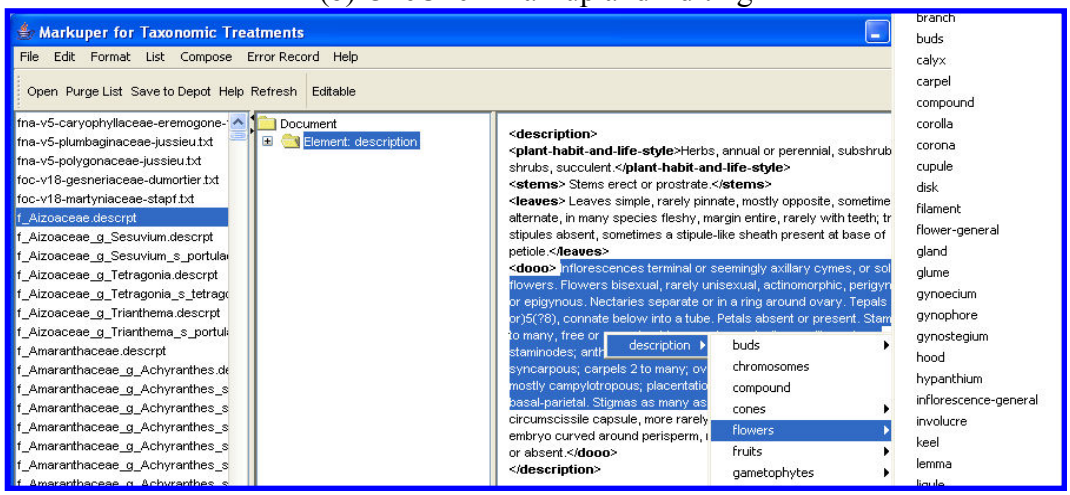
In addition, the subjects and their head nouns can be used as XML tags to tag the descriptions into well-formed XML documents. The well-formed XML documents may be imported to the training example preparation utility (Figure 3(c)) to generate training examples for MARTT at a much reduced cost. MARTT may also directly use the tags to mark up elements with few training examples. Hence the simple unsupervised learning algorithm addresses the schema coverage problem and the lack of training example problem at the same time.

The bootstrap algorithm was tested, without being given any seed nouns, on three collections: one contained 120 algae descriptions extracted from Feist, et. al, (2005), another contained 200 FNA descriptions, and the third contained 2367 FNA descriptions. Table 7 shows the evaluation results. From the 538 sentences of the algae descriptions, the algorithm learned 37 good singular nouns (correct rate = 95%) and 13 good plural nouns (correct rate = 87%), and tagged 476 sentences correctly (correct rate = 88%). From the

(a) The Composition Area



(b) OneClick Markup and Editing



(c) Manual Markup

Figure 3. Training example preparation and verification utility interface.

3195 sentences of the FNA descriptions (labeled as FNA-1 in the table), the algorithm learned 152 good singular nouns (correct rate = 99%) and 90 good plural nouns (correct rate = 100%), and tagged 3140 sentences correctly (correct rate = 98%). An example correct tag is "inner petals" while an incorrect one may be "petals generally" or "in some species base" (Figure 4a).

Note the number of unique tags does not grow linearly with the size of description collections. This ensures that a visual display of the learned tags and their structure will not get overly crowded with larger collections. As Table 7 shows, while the number of sentences in FNA-1 is 6 times of that in algae collection, the number of tags learned from FNA-1 is only 2 times of that from algae. To confirm this observation, a larger FNA collection (labeled FNA-2 in Table 7) with 31387 sentences was processed and the result shows that, comparing FNA-2 with FNA-1, while the number of sentences increases 9-fold, the number of tags only increases 2-fold. The number of unique tags increases at a much lower rate than the number of sentences and is expected to reach a plateau.

The diagram in Figure 4 visualizes the resultant XML schema, including the discovered tags and their structural relationships. Figure 4a and 4b shows the interactive diagrams generated from the algae and FNA-1 descriptions respectively. The "is part of" relationships are displayed in the diagrams by connecting sub-organs to their parent organs. The visualization readily shows the organs and how consistently periods and semicolons were used in the text. FNA descriptions often use periods and semicolons to set off major structure descriptions and subpart descriptions respectively, hence we see rather clearly the main organ elements such as *leaves, inflorescences, flowers, fruits,* and *seeds* as the first level elements and their subparts as the second level elements (Figure 4b). In contrast, the algae descriptions do not follow the same convention in using periods and semicolons; instead, they use mostly semicolons to separate different descriptive segments. Therefore in the diagram there is no clear-cut main organs level or subparts level (Figure 4a). The diagram may be further explored; for example, when a tag is selected, the interface displays the original descriptions on which the tag is applied. A visual interface like this assists the human expert in refining the raw schema to make it fit the descriptions better.

THE PERFORMANCE VARIATION ISSUE

The results from the MARTT experiments show that the system performed better on the FNA and FoC collections than on the FNCT collection. Performance differences were also seen among different elements, for example *flowers* elements were more difficult than others. What characteristics of data sets cause the performance difference? Can these characteristics be measured and used to predict MARTT performance? A performance prediction model helps to answer questions such as "how well will this system work on this description collection?" Instead of asking the user to prepare hundreds of training examples to test the system out, we developed a prototype utility that has the potential to predict the performance with just a few dozens of examples. At the center of the utility are two modules: one module measures characteristics of a set of examples, and the other uses the prediction model to make the prediction basing on the measurements.

The prediction model was established and tested on FNA, FoC, and FNCT descriptions using the following procedure:

1. A set of 11 corpus characteristic measures were derived.
2. 177 collections of description segments (5-56 files per collection) were created from FNA, FoC, and FNCT training examples.
3. The characteristics of each collection were measured.
4. MARTT performance on these collections was evaluated.
5. Statistical analyses were carried out to find correlations between the characteristic measures and system performance.

Steps 1 and 3: characteristic measurements

We derived the following 11 corpus characteristic measures that can potentially have an impact on system performance. The statistical analyses carried out in step 5 will reveal the ones with statistically significant impact.

Table 7: Performance of the unsupervised algorithm on alga and two FNA collections.

	Alga	FNA-1	FNA-2
Descriptions	120	200	2367
Sentences	538	3195	31387
Sentences correctly tagged (%)	476(88)	3140(98)	*
Unique tags	61	143	444
Singular nouns learned	39	154	504
Correct singular nouns(%)	37(95)	152(99)	490
Plural nouns learned	15	90	297
Correct plural nouns(%)	13(87)	90(100)	295
Boundary words learned	44	317	932
Correct boundary words	44	317	931
Process time	1 minute	1 minute	15 minute

1. *Instance Count* is the number of examples (i.e. documents) in a collection.
2. *Class Count* is the number of unique terminal elements in a collection.

3-5. *N-gram Count (N ∈ [1,2,3])* is the number of unique *n*-grams in a collection.

6-8. *N-gram Distribution Score (N ∈ [1,2,3])* gauges the distinctiveness of *n*-gram distributions in terminal elements in a collection. If an *n*-gram occurs *m* (*m*>*1*) times in a collection and all occurrences are in one terminal element *e*, then we say the distribution of the *n*-gram is very distinctive in that the presence of the *n*-gram itself suggests the element. If all *n*-grams have such a distinctive distribution, the markup task would be trivial. At the other extreme, if the *m* occurrences are uniformly distributed in the elements, then the presence of the *n*-gram is of little value to the markup task. The final *n*-gram distribution score is the mean distinctiveness scores of all *n*-grams counted in a collection. The formula for the measure is

$$distri.score = \frac{\sum_{g_i \in G} \frac{\max_{classes} |g_i|}{|g_i|} \left(\frac{|g_i|-1}{|g_i|} \right)}{|G|}$$

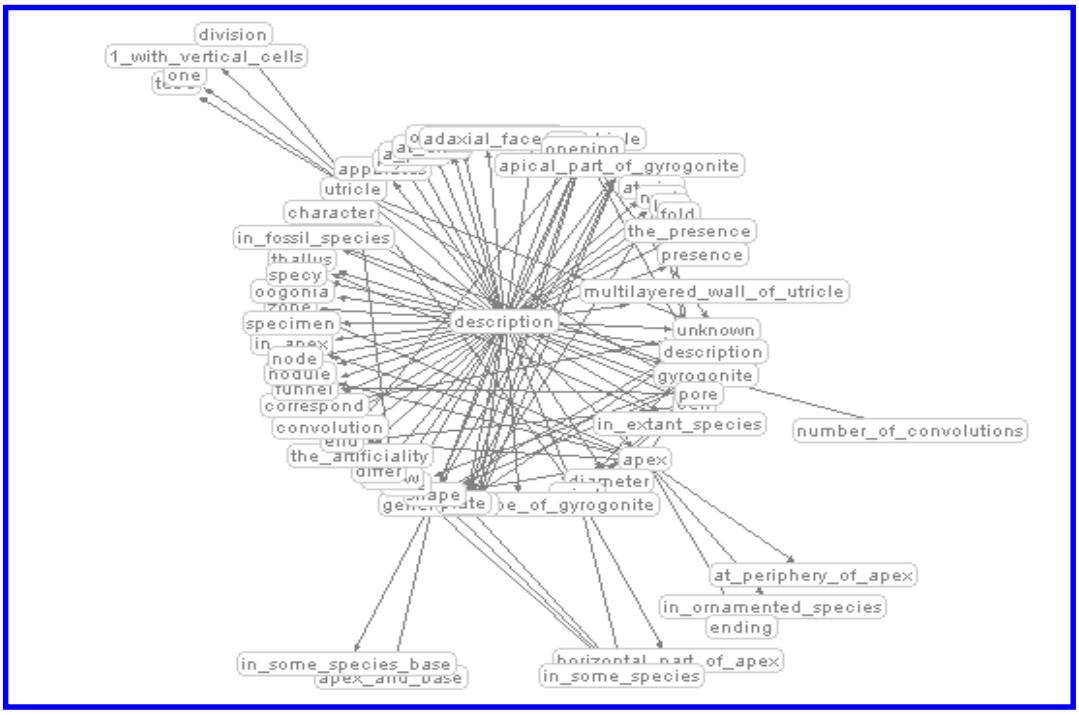
where $G = (g_1, ..., g_n)$. In the formula, an *n*-gram g_i's maximum occurrence in all terminal elements is divided by g_i's total occurrence in the collection. This simple division roughly measures the distinctiveness of g_i's distribution. The factor $\left(\frac{|g_i|-1}{|g_i|} \right)$ is used to discount the effect of rare *n*-grams. The final score is obtained by taking the average over all *n*-grams to remove possible sample size effect. The score is a value between *0* and close to *1*.

9. *Delimiter Score* measures the consistency of delimiters. Here a delimiter is a textual pattern that separates a previous element from the current one and the current one from the next one. For example, a delimiter pattern ". /Fruit berry/. /" indicates that following a period, a fruit type description starts with the words "Fruit berry" and ends with another period. The delimiter score uses information entropy (IE) to measure the distribution of delimiting patterns in a collection. The lower the IE score, the more distinctive the distribution. If all examples in a collection shares one delimiting pattern, the markup task would be much easier than in a collection where each element has a unique pattern. The formula for this measure is

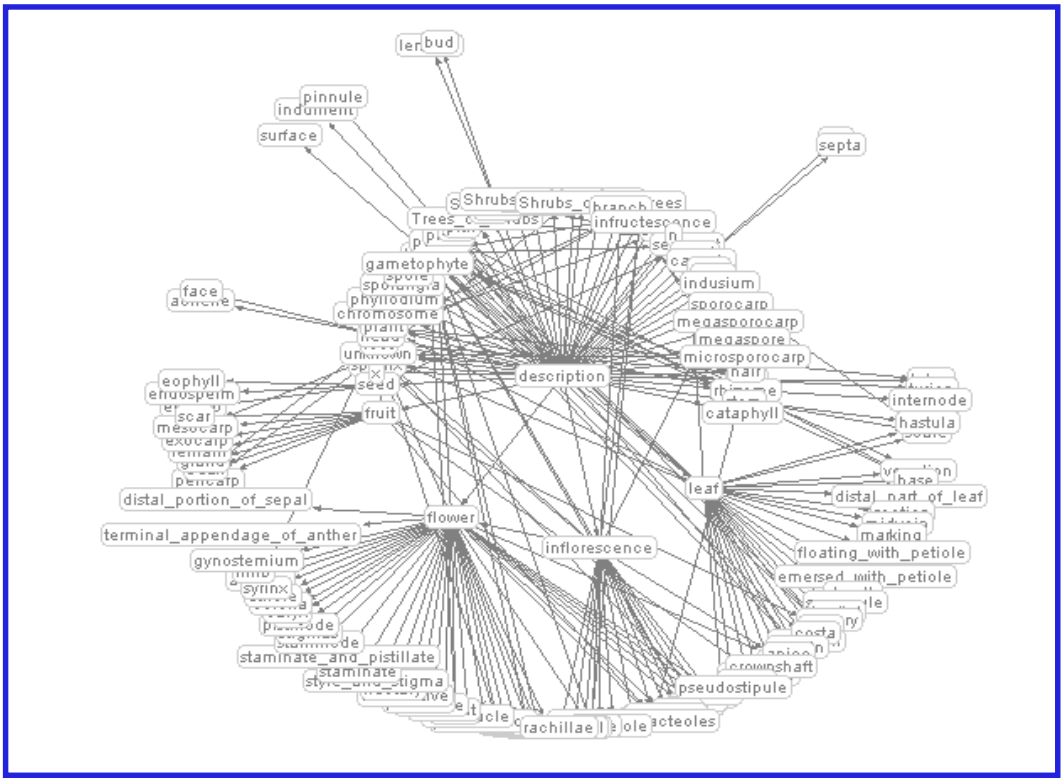
$$IE = -\sum_{d_i \in D} \frac{|d_i|}{|D|} \log\left(\frac{|d_i|}{|D|} \right)$$

$$Max.IE = -\sum_{d_i \in D} \frac{1}{|D|} \log\left(\frac{1}{|D|} \right)$$

$$Delimiter\ Score = 1 - \frac{IE}{Max.IE}$$



(a) Visualization of the Alga Collection



(b) Visualization of the FNA-1 Collection

Figure 4. Visualizations of learned tags and their structural relationships.

where $D = (d_1, \ldots, d_n)$. To find the delimiter score for a collection, the delimiters $d_1, \ldots, d_n$ of terminal elements are gathered. The standard IE is calculated using the occurrences of different patterns in the collection. The IE reaches its maximum when each pattern occurs only once. The maximum IE is used to make the delimiter score a positive measure of the distinctiveness of a distribution (i.e. the higher the score, the more distinctive the distribution). The score is a value between *0* and *1*.

10. *Class Order Score* and the next measure evaluate the consistency of the element sequences in a collection. Class order score deals with the order of the terminal elements. An example of an order may be "inflorescence, sepal, petal, style" in a flower description. Descriptions with some or all of these four terminal elements presented in that order are said to "fit" that sequence. Consistent sequences are useful for a markup algorithm to make sound decisions on some otherwise difficult cases. Similar to the delimiter score, the maximum IE is used here. This score is calculated as the following:

$$IE = -\sum_{o_i \in [o_1 \ldots o_n]} \frac{\#\ of\ examples\ fit\ o_i}{total\ examples} \log\left(\frac{\#\ of\ examples\ fit\ o_i}{total\ examples}\right)$$

$$Max.\ IE = -\sum_{all\ orders} \frac{1}{\#\ of\ all\ orders} \log\left(\frac{1}{\#\ of\ all\ orders}\right)$$

$$Class\ Order\ Score = 1 - \frac{IE}{Max.\ IE}$$

To get the score for a collection, the sequences of terminal elements are collected and the examples fitting a sequence are counted. The maximum IE is calculated based on the number of all possible sequences, which is either the number of the total examples in the collection, or the number of all permutations of terminal elements, whichever is smaller. Similar to the delimiter score, the class order score is a positive measure with a value between *0* and *1*.

11. *Class Presence Score* considers the presence/absence patterns of terminal elements regardless of their order. The score is calculated in a rather similar way as the class order score. For maximum IE, the number of all possible patterns is either the number of the total examples in the collection, or the number of all combinations of terminal elements, whichever is smaller. The class presence score is a positive measure with a value between 0 and 1.

$$IE = -\sum_{p_i \in [p_1 \ldots p_n]} \frac{\#\ of\ example with\ p_i}{total\ examples} \log\left(\frac{\#\ of\ example with\ p_i}{total\ examples}\right)$$

$$Max.\ IE = -\sum_{all\ present\ patterns} \frac{1}{\#\ of\ all\ patterns} \log\left(\frac{1}{\#\ of\ all\ patterns}\right)$$

$$Class\ Presence\ Score = 1 - \frac{IE}{Max.\ IE}$$

Step 2: Creation of 177 collections

The 177 collections were created using the following procedure. First, 1500 descriptions from the three floras (633 from FNA, 492 from FoC, and 378 from FNCT) were manually marked-up in the XML format. The sample sizes were increased from those used in the MARTT experiments to generate enough collections for statistical analyses. These descriptions were then randomly divided into 30 sets of 50 descriptions. Then each description was split into several parts, each of which contained a text segment describing a main organ (e.g. flowers, fruit, etc). From this point on, each part was treated as an individual document. The documents that were in the same set and contained the same main organ element formed a collection. Of the resultant 200 collections, 23 collections had fewer than 5 documents and were removed because they were too small to measure MARTT performance using a 5-fold cross-validation routine. Each remaining collection consisted of 5 to 54 (mean = 23) documents. Among the 177 remaining collections, 135 random collections were used in the statistical analyses to derive the performance prediction model, and the remaining 42 collections were reserved to test the prediction model. The collections produced provide a reasonable representation of the taxonomic description population, as the documents were drawn from three different sources. They also preserve the element distribution variations seen in the original descriptions. In the end, each document contained a 2-level, flat XML structure. This simple model

allowed us to focus on the effect of characteristic measures on system performance. The more involved multi-level hierarchical structures will be examined in the future.

Step 4: Performance measurement
Instead of precision/recall, we used a single-valued cosine similarity-like measure to evaluate markup accuracy, which is essentially a normalized value characterizing the proportion of the words tagged correctly in a description.

Step 5: Statistical analyses
The SPSS linear regression analysis on 135 of the 177 collections between characteristic measurements (the independent variables) and system performance (the dependent variable) constructed the following model:

$$\begin{aligned} performance = {} & 0.725 + 0.176 * (class\ presence) \\ & + 0.372 * (unigram\ distribution) \\ & - 0.002 * (class\ count) \end{aligned}$$

This model explained 64% of the original variance in performance and the residual of the model is normally distributed, as Figure 5 shown (the closer the plot of the residual to the diagonal line, the closer the distribution to the normal distribution), indicating the linear model is a good fit. The model shows that among the eleven characteristics measured, *class presence, unigram distribution,* and *class count* are the statistically significant factors for determining the performance score.

The prediction model was tested on the reserved 42 of the 177 collections. The performances of MARTT on the 42 collections ranged from 60% to 100%. The differences between observed performance and predicted performance are plotted in Figure 6, which shows the residual distribution is quite close to normal. The residual of over *50%* of the cases is in □*0.03* range (meaning the predicted value is *0.03* less or more than the observed value). This result seems very promising.

The reader should keep in mind that the prediction model was derived basing on the data from the three floras. At this time, the coefficients should not be interpreted literally. We will continue to test and refine the prediction model with more data from other sources.

LITERATURE REVIEW

The majority of studies on structuring plain-text taxonomic descriptions have relied on handcrafted rules which make heavy use of formatting and textual cues. Organism nomenclature, for example, conforms closely to prescribed rules and can be reliably extracted by software programs using a combination of contextual rules and a language lexicon (Kirkup et.al., 2005; Koning et al., 2005). Sautter et al. (2006) built on top of Koning et al.'s system a Named Entity Recognition system for taxonomic names, using both hand-crafted rules and some learning components. Fewer studies have focused on cue-poor yet semantic-rich sections (e.g. morphological descriptions) largely due to the lack of consistency in description contents. Lydon et al. (2003)'s manual comparison revealed surprisingly large inter-collection variations among descriptions of the same species. Earlier studies using syntactic parsing methods to extract information to populate relational databases or to mark up plant descriptions in XML have focused on a single collection (Taylor, 1995; Abascal et.al., 1999; Vanel, 2004). Recently, Wood et. al.(2004) extracted plant features from the descriptions of five species found in six floras, using a hand-made gazetteer as a lookup list to link extracted terms with their tags. They also showed that features extracted from different sources were complementary to each other. The research reported in this paper involves multiple description collections and multiple user-friendly approaches, minimizing manual work as much as possible.

GoldenGATE (Sautter et al., 2007) is an XML editor that facilitates the markup of plain-text taxonomic descriptions in XML. It works with complete documents and the user can invoke different functions to paginate documents and to tag taxonomic names and taxon names, in other words, to tag a document to TaxonX level 1. TaxonX is an XML schema that defines five levels of markup. The sentence level markup described in this paper is between TaxonX level 2 and 3. GoldenGATE relies on regular expressions and pre-compiled dictionaries to tag description text. This approach can be sensitive to text variations

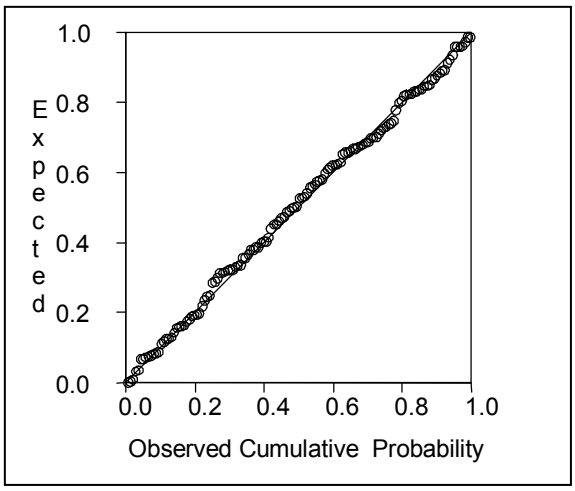


Figure 5. Normal distribution of the residual of the linear regression model.

and is limited by the availability of the dictionaries and user skills in constructing regular expression patterns. GoldenGATE supports manual editing of tagged text in a similar way as MARTT's training example preparation utility. Others, such as Cui et. al (2002), used text classification algorithms such as SVMs to mark up description paragraphs as nomenclature, description, distribution, discussion, and reference, etc. with good accuracy.

Few studies linked characteristic measures of text corpora to system performance statistically. An exception is Bagga & Biermann (1997) who proposed a measure called "fact level" to evaluate the complexity of a text corpus in the context of information extraction, basing on the observation that it is more difficult to extract a fact when its components are scattered around in the text. The study showed that higher fact levels are associated with lower performances in information extraction systems, indicating that fact level may be an appropriate measure for extraction difficulty. However, fact level is not applicable to the semantic markup scenario discussed here.

CONCLUSIONS AND FUTURE WORK

Our experience with taxonomic descriptions confirms Lydon et.al (2003)'s conclusion that large variations exist among collections of descriptions. Domain practices (e.g., use of punctuation marks) are not adopted uniformly across collections. These variations demand any automated semantic markup systems to enhance not only its accuracy but also its portability.

The uniqueness of MARTT lies in its ability to store and reuse markup rules learned over time from different description collections. This makes it highly portable across collections as demonstrated in the experiments with FNA, FoC, and FNCT. Because the learned markup rules are collection-independent, we hope that these rules accumulated over time will be also useful for tagging more free-style text related to taxonomy. As a machine learning system, MARTT compares candidate markup rules learned from training examples to select the rule with the lowest expected error rate and the highest expected correct rate. This feature releases the user from the difficult and time consuming task of crafting markup rules. To make the system more efficient and user-friendly, a number of utilities are also being developed. The training example preparation utility can significantly reduce the cost of training examples. The unsupervised learning utility identifies main concepts (organ names) from a description collection without any training example and helps the user to create a more comprehensive XML schema and training examples at low cost. Lastly, the performance prediction utility shows the potential of predicting MARTT performance on a collection with only a few dozens of tagged examples.

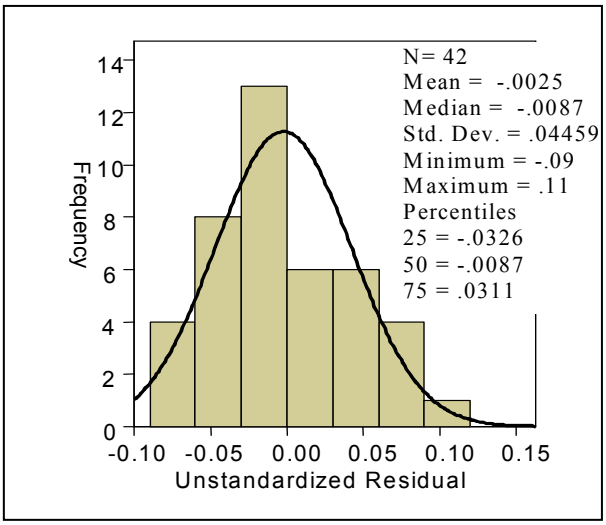


Figure 6. The difference between observed performance and predicted performance on 42 test collections.

In the course of developing the MARTT system and its utilities, we essentially have tested two machine learning approaches: the first is a supervised learning approach where an XML schema and a set of training examples guide the markup decisions of the algorithm; the second is an unsupervised learning approach where the algorithm exploits implicit regularities in the description text without any training examples or a schema. Although both approaches are capable of producing well-formed XML documents from plain-text taxonomic descriptions, the latter is more efficient but the former integrates more domain knowledge. For example, "is part of" and "is type of" relationships are more accurately represented in the supervised approach. It is important to note, however, the two approaches are mutually beneficial in that the unsupervised approach helps to create a comprehensive XML schema and training examples that the supervised approach needs, while the schema and the rules learned by the supervised approach can help to improve the performance of unsupervised approach (e.g., by providing good seed nouns).

While the markup at the sentence level can benefit information retrieval by supporting fielded searches, in the immediate future we will further develop MARTT to tag at an even finer granularity; that is, to tag characters and character states in descriptions. The character level markups will prove more useful and powerful: they can be used to support database-like queries, to merge descriptions from multiple collections, to generate taxonomic keys either in a semi-automated or automated manner, and to compare descriptions along multiple dimensions, to name just a few possibilities. We will format the tagged description in standard formats such as SDD to share them with the community. SDD does not prescribe a standard set of characters to be included, but leaves the decision to individual applications. To ensure our SDD documents interoperate with others, a conceptual model (i.e., ontoglogy) with broad coverage is indispensable. We will look into the issues on ontology construction and how to use the ontology to guide the markup practice.

As well, we will conduct further evaluation of the entire system from a more user-centered perspective. We will examine in a systematic manner the effort required on the user's side to mark up a sizeable collection using MARTT and its utilities. To provide a comprehensive and useful evaluation, the author is more than willing to collaborate with contributors and rights-holders of any taxonomic collection.

ACKNOWLEDGMENTS

This work was in part supported by an Internal Research Grant of Faculty of Information and Media Studies, University of Western Ontario. The author thanks the editorial committees and authors of the taxonomic works for the permissions to use their text in this project. The author thanks Dr. Richard McCourt, Dr. P. Bryan Heidorn, Dr. Linda Smith, and others for their constructive comments on the design and evaluation of MARTT and its utilities. The author also thanks the BI editor and reviewers for their valuable paper revision suggestions.

REFERENCES

Abascal, R. & Sánchez. 1999. X-tract: Structure extraction from botanical textual descriptions. Proceedings of the String Processing & Information Retrieval Symposium and International Workshop on Groupware, SPIRE/CRIWG, pp. 2-7.

Bagga, A., & Biermann A.W. 1997. Analyzing the complexity of a domain with respect to an information extraction task. Proceedings of the Tenth International Conference on Research on Computational Linguistics (ROCLING X), pp. 175-194.

BHL 2007. Biodiversity Heritage Library. Accessed 10 July 2007 from http://www.bhl.si.edu/.

Cui, H. 2005a. The XML Schema for MARTT, Accessed 10 July 2007 from http://publish.uwo.ca/~hcui7/research/xmlschema.xsd.

Cui, H. 2005b. MARTT: Using knowledge based approach to automatically mark up plant taxonomic descriptions with XML. Proceedings of the Annual Meeting of American Association of Information and Technology. Oct 28-Nov 2. 2005 Charlotte, North Carolina, USA.

Cui, H. 2005c. Automating semantic markup of semi-structured text via an induced knowledge base: A case-study using floras. Doctoral Dissertation. The University of Illinois at Urbana-Champaign.

Cui, H., Heidorn, P.B., & Zhang, H. 2002. An approach to automatic classification for information retrieval. Proceedings of the Joint Conference of Digital Libraries 2002, 96-97.

Cui, H., & Heidorn, P.B. 2007. The reusability of induced knowledge for the automatic semantic markup of taxonomic descriptions. J. Am. Soc. Inf. Sci. Technol. 58:133-149.

Dallwitz, M.J. 1980. A general system for coding taxonomic descriptions. Taxon 29:41-46

Diggs, G.M, Lipscomb, B.L., & O'Kennon R.J. 1999. Shinners & Mahler's Illustrated Flora of North Central Texas. Center for Environmental Studies and Department of Biology, Austin College, Sherman, Texas, and Botanical Research Institute of Texas (BRIT), Fort Worth, Texas.

Feist, M., Crambast-Fessard, N., Guerlesquin, M., Karol, K., Huinan, L., & McCourt, R. M. et al. 2005. Treaties on Invertebrate Paleontology, Part B: Protoctista 1 Volume 1: Charophyta. Boulder, Colorado: Geological Society of America, Inc. & Lawrence, Kansas: University of Kansas.

FNA Editorial Committee 2006. Flora of North America North of Mexico Guide for Contributors. Accessed July 10, 2007 from http://www.fna.org/FNA/Guide/guide_2006.pdf.

FNA Flora of North America Editorial Committee. (Eds.). 1993 onwards. Flora of North America North of Mexico. Accessed July 10, 2007 from http://www.fna.org/.

FOC Flora of China Editorial Committee. (Eds.). 1994 onwards. Flora of China. Beijing/St. Louis: Science Press/Missouri Botanical Garden Press. Accessed July 10, 2007 from http://flora.huh.harvard.edu/china/.

GBIF. 2007. Global Biodiversity Information Facility Accessed July 10, 2007 from http://www.gbif.org/.

Han, J. & Kamber, M. 2000. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers.

Kirkup, D., Malcolm, P., Christian, G., & Paton, A. 2005. Towards a digital African flora. Taxon 54:457-466.

Koning, D., Sarkar, I.N., & Moritz, T 2005. TaxonGrab: Extracting taxonomic names from text. Biodiv. Inf. 2:79-82.

Lydon, S.J., Wood, M. M., Huxley, R., & Sutton, D. 2003. Data patterns in multiple botanical descriptions: Implications for automatic processing of legacy data. Syst. Biodiv. 1:151-157.

McCallum, A.K. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. Accessed April 20, 2003. http://www.cs.cmu.edu/~mccallum/bow.

Mitchell, T. 1997. Machine Learning. McGraw Hill: New York, NY.

Radford, A.E. 1986. Fundamentals of Plant Systematics. Harper & Row, Publishers, Inc.: New York, NY.

Sautter, G., Agosti, D., & Böhm, K. 2007. Semi-Automated XML Markup of Biosystematics Legacy Literature with the GoldenGATE Editor, Proceedings of PSB 2007, Wailea, HI, USA. Accessed July 10, 2007 from http://psb.stanford.edu/psb-online/proceedings/psb07/sautter.pdf.

Sautter, G., Agosti, D., & Böhm, K. 2006. A combining approach to find all taxon names (FAT). Biodiv. Inf. 3:46-58.

Taylor, A.1995. Extracting knowledge from biological descriptions. Proceedings of 2nd International Conference on Building and Sharing Very Large-Scale Knowledge Bases. pp114-119, Enschede, The Netherlands, April.

Vanel, J.-M. 2004. Worldwide Botanical Knowledge Base. Accessed July 5, 2007 from http://wwbota.free.fr/.

Witten, I.H., McNab, R.J., Boddie, S.J. & Bainbridge, D. 2000. Greenstone: A comprehensive open-source digital library software system. Proceedings of Digital Libraries 2000, pp. 113-121, San Antonio, Texas, June.

Wood, M., Lydon, S., Tablan, V., Maynard, D. & Cunningham, H. 2004. Populating a database from parallel texts using ontology-based information extraction. Proceedings of Natural Language Processing and Information Systems, 9th International Conference on Applications of Natural Languages to Information Systems. pp.254-264, Salford, UK, June.