MARTT: Using Induced Knowledge Base to Automatically Mark up Plant Taxonomic Descriptions with XML

Hong Cui

Faculty of Information and Media Studies, 254 North Campus Building, University of Western Ontario, London, Ontario Canada N6A 5B8. hcui7@uwo.ca

Despite the sub-language nature of taxonomic descriptions of plants, researchers warned about the large variations among different collections of descriptions in terms of information contents and presentations. These variations impose a serious challenge to the development of automatic tools for the semantic markup of large volumes of free-text descriptions. This paper presents a new approach to automatic markup of multiple collections of taxonomic descriptions with XML. The effectiveness of the approach was demonstrated with markup experiments using three contemporary floras. The markup system, MARTT, was based on supervised machine learning algorithms and enhanced by machine learned association rules representing certain types of domain knowledge and conventions. Experiments showed that our simple and efficient markup algorithm outperformed popular general-purpose algorithms (including SVMs) across different floras. More importantly, the results demonstrated that the domain knowledge learned from one flora was useful for improving the markup performance on a second flora, especially on elements with sparse training examples. The system design and the evaluation of markup algorithms are reported in this paper. The study on the effectiveness of the induced knowledge base will be reported in a later paper. In this paper, common practices of flora authors and the potentials of MARTT system for improving the efficiency and effectiveness of the creation, organization, and utilization of plant descriptions are also discussed.

Introduction

The organization and access of taxonomic information is a major component of biodiversity informatics research. Despite the recent development in 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 generating structured data from the existing free-text descriptions.

Figure 1 shows the free-text taxonomic description of *Chamaecyparis lawsoniana* in Flora of North America. For each paragraph, a semantic label was added in the right column. In this paper, the term "flora" is used to refer to a collection of taxonomic treatments of plants appearing in a geographic region as in Flora of North America (FNA), Flora of China (FoC) and Illustrated Flora of North Central Texas (FNCT).

2. Chamaecyparis lawsoniana (A. Murray bis) Parlatore, Ann. Mus. Imp. Fis. Firenze. n.s. 1: 181. [preprint p. 29]. 1864.	taxonomic information
Port-Orford-cedar, ginger-pine	commonnames
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.	morphological 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

Figure 1: Treatment of *Chamaecyparis lawsoniana* in FNA

DELTA (Description Language for Taxonomy) is a well known data formats for describing the structure of taxonomic data (Dallwitz, 1980). The Taxonomic Database Working Group (TDWG) is currently developing an XML-based standard for describing the structure in collections of taxonomic records, Structure of Descriptive Data (Thiele, 2003). DELTA and related systems can be used to generate and typeset descriptions and conventional keys. However, they do not provide means to automatically convert legacy data into the structured format.

Previous studies on automatically structuring taxonomic descriptions used mainly syntactic parsing methods and focused on structuring single collection of descriptions. Taylor (1995) and Abascal and Sánchez (1999) constructed grammars and lexicons to parse the Flora of New South Wales and Flora of North America respectively to extract triples of specimen part, attribute and values. Jean-Marc Vanel's Worldwide Botanical Knowledge Base (http://wwbota.free.fr/) also took the parsing approach, but aimed to mark up descriptions with XML. None of these works reported their scientific evaluation of markup accuracy.

The published works used syntactic parsing techniques that require extensive lexicons and handcrafted grammar rules. The parsing approach takes advantage of the sublanguage nature of formal floras. Lehrberger (1982) summarized the characteristics of a sublanguage: 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 (Lydon et al., 2003) suggested that automatic processors should expect to work with collections of taxonomic descriptions marked with large variations. To assess the feasibility of automatic processes of taxonomic legacy data, Lydon and colleagues manually compared and contrasted the descriptions of five common species from six English floras. The main findings included: 9% of information was presented in exact same format in all six sources; 55% of information came from only one source, 1% of information was in contradiction in different floras; and the remaining 35% of information was in more or less different formats (e.g. using different terms for the same concepts, etc.) in different sources. Our experience with a number of floras supports their findings.

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

In this paper, we report our progress in developing a machine learning based approach to automatic markup of taxonomic descriptions with XML using a schema with elements drawn from DELTA and other taxonomic ontologies. This research differs from others in that it aims to create an evolvable system that is capable of marking up not only one specific flora, but a large range of floras and possibly other similar text collections (for example, faunas). In previous work, we created a procedure based on Support Vector Machine (SVM) algorithm and marked up FNA treatments at paragraph level with around 95% accuracy (Cui, Heidorn & Zhang, 2002). In this study, the focus is on the deep markup of morphological description paragraphs. Figure 2 illustrates the current level of markup of morphological descriptions.

```
<?xml version="1.0" encoding="ISO8859-1" ?>
    <description>
      <plant-habit-and-life-style>
          <phls-general>Trees to 50 m;</phls-general>  <stems>trunk to 3 m diam.</stems>
      </plant-habit-and-life-style>
      <stems>
          <bark>Bark reddish brown, l0--20(--25) cm thick, divided into broad, rounded ridges.</bark>
          <branchlet>Branchlet sprays predominantly pinnate.</branchlet>
      </stems>
      <leaves>
          <leaf-general>Leaves of branchlets mostly 2--3 mm, apex acute to acuminate, facial leaves frequently separated
        by paired bases of lateral leaves;</leaf-general>
          <gland>glands usually present, linear.</gland>
    </leaves>
      <cones>
        <pollen-cones>
            <pollen-cone-general>Pollen cones 2--4 mm, dark brown;</pollen-cone-general>
            <pollen>pollen sacs red.</pollen>
      </pollen-cones>
        <seed-cones>
            <seed-cone-general>Seed cones maturing and opening first year, 8--12 mm broad, glaucous, purplish to
          reddish brown, not notably resinous;</seed-cone-general>  <scale>scales 5--9.</scale>
       </seed-cones>
    </cones>
      <seeds>
          <seed-general>Seeds 2--4 per scale, 2--5 mm, wing equal to or broader than body.</seed-general>
    </seeds>
  </description>
```

Figure 2: The Marked-up Morphological Description Paragraph of *Chamaecyparis lawsoniana*

The paper is organized as follows. We start with the basic idea behind the MARTT (MARkuper for Taxonomic Treatment) system and elaborate on the system design and the learning algorithms. We then report and discuss the results of selected experiments on

algorithm evaluation. Due to space limitation, experiments on the creation and evaluation of the knowledge base will be discussed in a later paper. We conclude the paper with a discussion on the professional practice of treatment authors and the potential of the MARTT system in improving the efficiency and effectiveness of the creation and access of taxonomic descriptions.

MARTT System

Syntactic parsing systems by design need some forms of common or domain knowledge, typically taking the form of lexicons and/or grammar rules. Domain knowledge is also of great value for improving the performance of a supervised learning system such as MARTT. However, the manual acquisition of domain knowledge, known as the "knowledge acquisition bottleneck", is often expensive and time consuming. Moreover, for any non-trivial domain, the coverage of the acquired knowledge is seldom complete. To work around the difficulty, we made use of the following observation: In the domain of taxonomic descriptions, a good amount of domain knowledge is presented explicitly in the text and shared by different floras. We hypothesize that the domain knowledge learned from one flora is effective for improving the markup performance of another flora. Specifically, we propose a two-phase learning framework for an evolvable markup system:

> Phase 1: Use inductive learning algorithms to mark up selected collections of taxonomic descriptions with richer structural cues. Mine domain knowledge from marked-up descriptions and populate the knowledge base.
>
> Phase 2: Use the inductive learning component enhanced by the knowledge base to mark up other descriptions. Newly marked-up descriptions may be fed back to phase 1 to evolve the knowledge base.

System Design

Figure 3 illustrates the major components of MARTT: the learning & markup component (LMC), the knowledge extraction component (KEC), and the induced knowledge base (KB). The system works as follows: In phase 1, a set of semi-structured plant description collections is selected as the base corpora. LMC then marks up the base corpora using machine learning algorithms. Next, the

marked-up corpora are fed to KEC, where certain types of domain knowledge and conventions are mined and saved to KB. At the end of phase 1, KB is populated and MARTT is ready to mark up new sets of descriptions. In phase 2, when marking up other unseen floras (i.e. test floras), KB facilitates the markup task of LMC by answering the queries initiated by LMC. In the remaining of this section, we describe LMC in detail.

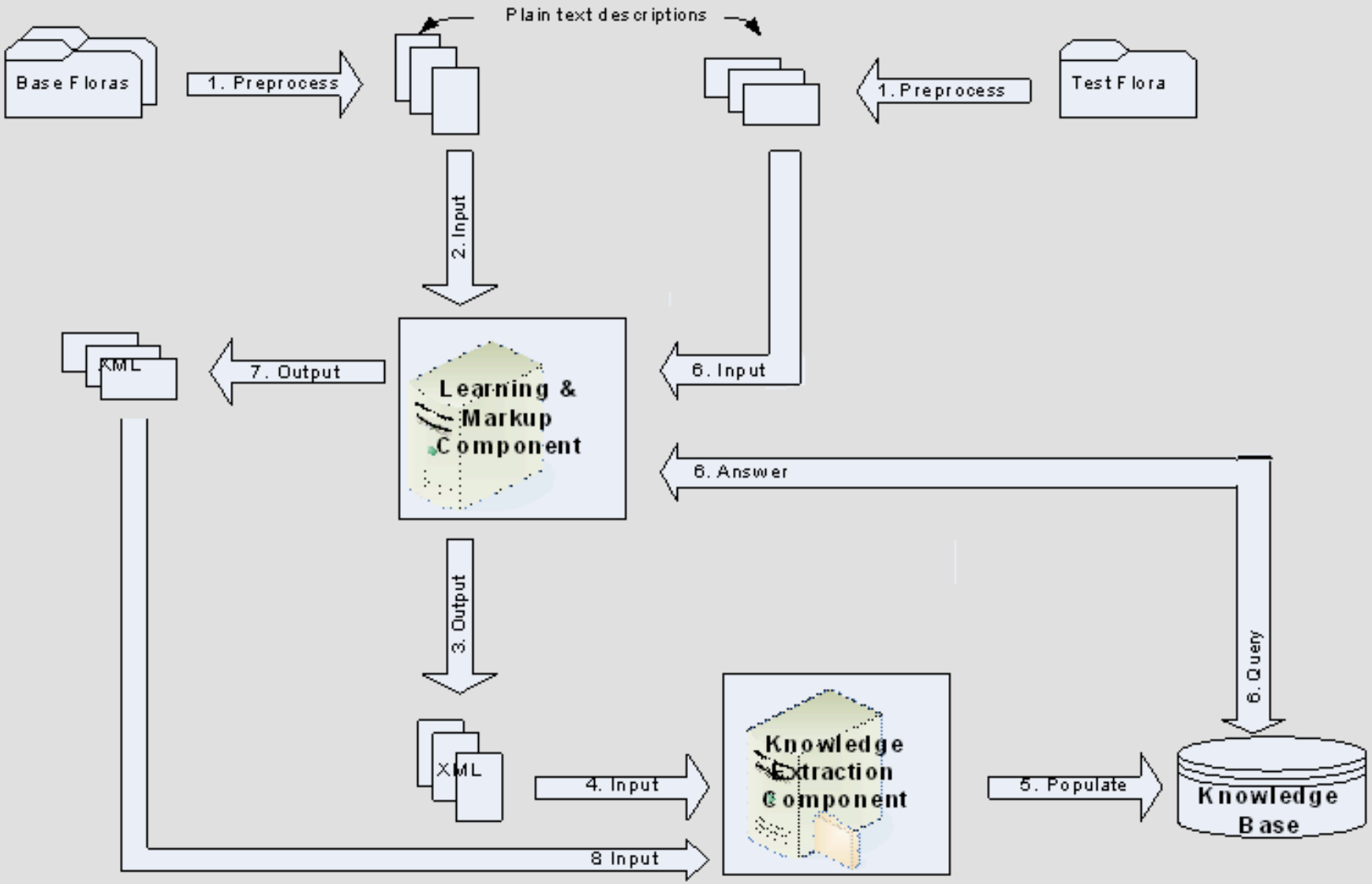

Figure 3: MARTT System Structure and Data Flow

LMC consists of a hierarchy of nodes, each of which corresponds to an XML element. The whole hierarchical structure corresponds to the structure of the target XML markup. Each internal node acts as a learning unit, which is in charge of learning and marking up its corresponding element. In contrast, leaf nodes do not perform learning or marking up but merely receive their text segments marked-up by their parent nodes. All learning nodes also evaluate their markup performance by comparing the marked-up segments with the answer keys. Answer keys contain the correct markup annotated by human experts. Answer keys are read into the hierarchy in such a manner that each node receives their segments of the answer. Figure 4 illustrates the hierarchical structure.

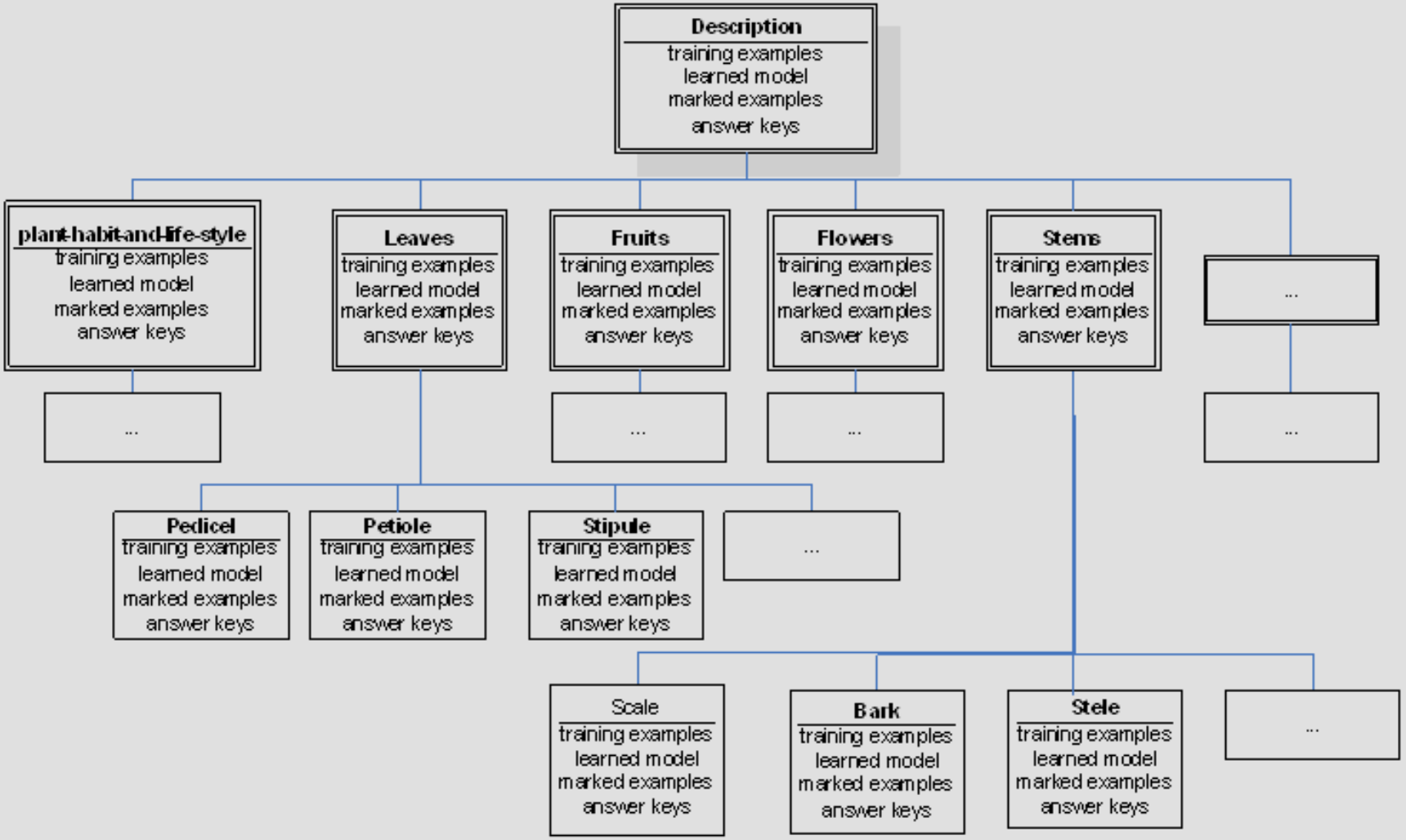

Figure 4: Learning Hierarchy

The learning hierarchy is initialized from training examples (i.e. human marked-up taxonomic descriptions, similar as answer keys). Starting with an empty hierarchy, LMC reads in training examples one by one. If LMC encounters a new element, it creates a new node and inserts the node at an appropriate location in the hierarchy, reflecting the location of the element in the XML structure of

the training example. LMC also extracts the content of the element and saves it in the node's pool of training examples. LMC repeats these operations until all training examples are read in. In the end, the hierarchy contains every element seen in the training examples, with each element represented as a node. Each node contains the segments of training instances relevant to its markup task. Nested elements in the training segments are flattened out so that training segments for each internal node are flat 1-level deep XML segments.

Once the learning hierarchy is initialized, each learning node starts to learn a model from its training segments. The learned models contain the information needed for marking up new examples. Using the learning hierarchy in Figure 4 as an example, the markup process can be described as follows: A new description is read into the root node-- *description*. The node *description* uses its learned model to mark up the input text with its child elements such as *plant-habit-and-life-style, stems*, and *leaves* etc. as tags. When the root node is done with the markup, a copy of the marked-up segment is saved to its pool of marked-up examples and the newly tagged sub-segments are dispatched to their corresponding nodes at the lower level where they will be marked up one level deeper if necessary. For example, the segment marked as *stems* by *description* node is sent to the node *stems*, where it will be further marked up with tags such as *scale, bark*, and *stele*, etc. The contents of marked-up elements are dispatched in this manner until they reach a leaf node. Since each node keeps its marked-up segments, depth-first traverses of the hierarchy can be easily controlled to produce complete marked-up descriptions with varied granularities.

Each node is free to choose its markup approach that is appropriate for its situation. Having examined its training examples, the node *description* would use one of the available text segmentation and classification algorithms to identify the boundaries of relatively large chunks of text that describe stems, leaves, and flowers etc. and label them with the correct tag. On the other hand, the node *stipule* may decide to use a regular expression based information extraction algorithm to extract smaller pieces of text, for example, the color of a stipule.

To evaluate the markup performance, answer keys are read into the hierarchy in the same way as the training examples. Each node would then have machine marked-up segments and segments from the answer keys, making it possible to check for the correctness of the machine-generated markup. To summarize, each node in the learning hierarchy contains training examples, marked-up examples, answer keys and the learned model (Figure 4). Learned models of leaf nodes are empty as these nodes do not perform learning or markup tasks.

Using training examples to initialize the hierarchy makes LMC adaptable to different level of markup: if sentence level markup is desired, merely feed LMC with training examples that are marked-up to sentence level. Marking up any selected element is also straightforward, 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. The hierarchical structure is separated from the methods that access the hierarchy to give the maximum flexibility to update and/or integrate new access methods without having to modify the hierarchy in any way.

Learning Algorithms

A number of segmentation and classification algorithms and one information extraction algorithm were implemented and available to LMC. In this section, we explain four segmentation and classification algorithms used in the experiments. None of the four algorithms used morphological operations, such as stemming, on the text of taxonomic descriptions because the morphological features of terms and even their cases seem to provide good clues for segmentation and classification. For example, the term "Seed" often occurs in "Seed cones" as a modifier for "cones", while the term "Seeds" often starts the description of the seeds of a taxon.

CT: The content-based markup algorithm learns terms' element scores (formula 1), elements' punctuation mark scores (formula 2), and an element transition matrix.

$$element\ score_{t,e} = \frac{occurrence\ of\ term\ t\ in\ element\ e}{total\ occurrence\ of\ term\ t} \quad (1)$$

$$punc\ score_{e,p} = \frac{|punc\ mark\ \ p\ ends\ element\ e|}{total\ occurrence\ of\ element\ e} \quad \textbf{(2)}$$

The transition matrix is a square matrix of the probabilities of one element appearing immediately after another element as seen in the training examples.

When marking up a new description *T*, CT first finds the sum of the element scores of the *l* leading words from *T* for each possible element. The *n* elements with the highest scores are selected as tag candidates. If all scores are zero, the *n* tag candidates are selected by using the transition matrix, picking the top *n* elements that are most likely to follow the previous element.

Each tag candidate then proposes a segment candidate from *T*. A tag candidate first obtains a set of segments from the *T* by using different punctuation marks seen for this element in training examples. The segments are ranked by segment scores (formula 3) to find the most probable segment for the candidate element. The segment with the highest score is selected.

$$segment\ score = element\ score\ of\ segment \times \sqrt{punc.\ mark\ score} \quad \textbf{(3)}$$

Each tag candidate and its proposed segment are then evaluated as one markup decision and given a score using formula (4).

$$mscore = (element\ score\ of\ l\ words)^2 \times segment\ score \quad \textbf{(4)}$$

The top two decisions are then subject to the last test using the element transition matrix. Let p_{e_1,e_2} be the probability that e_1 is

followed by e_2, the final score is

$$fscore = mscore \times p_{e_1,e_2} \quad (5)$$

The decision with higher *mscore* and *fscore* is the final decision. In cases where no decision meets the criterion, the decision with *mscore* three times higher than the other is the final decision, otherwise, the decision with the higher *fscore* wins. The segment is tagged according to the winning decision. The algorithm then starts again to mark up the remainder of T.

CT considers a number of different combinations of tag and segmentation options. It is based on the classic assumption of text classification that if a segment belongs to a class, it must contain many terms that are good indicators of the class.

NB: Naïve Bayes based markup algorithm is based on the same assumption, but uses a Bayes theorem based segment scoring method instead (formula 6). All other aspects of the algorithm are the same as CT.

$$score_{e,segment} = P(element = e) \sum_{t \in segment} P(term = t \mid element = e) \quad (6)$$

SCCP: The semantic class based markup algorithm focuses its learning on the subjects of the sentences and clauses in taxonomic descriptions. These noun phrases often indicate exactly to which element the text belong (Figure 2).

SCCP learns nouns and noun phrases that have clear class indications, for example “Seed cones” and “Roots” are good indicators for element cones and roots respectively. SCCP does not use a part of speech (POS) tagger to identify the nouns and noun phrases because the available POS taggers are often for general domain and do not work well with taxonomic descriptions. Instead, SCCP uses frequent pattern and association rule learning methods, originated from data mining research, to learn rules of the format: *n-gram* → *element (confidence, support*). The rule should read “*n-gram* is associated with *element* with confidence *confidence* and support *support*”. In association rule learning, confidence and support are a pair of popular metrics measuring the strength of the

associations. Rules with scores higher than user-defined thresholds are assumed to be good rules (Han & Kamber, 2001). For our setting, confidence was defined as the ratio of the occurrence of the *n*-gram in the element and the total occurrence of the *n-gram*; support was defined as the ratio of the occurrence of the *n*-gram and the number of segments in the element.

SCCP learns the association rules by first generating sets of *n*-grams as the candidate nouns and noun phrases and then calculating the confidence and support scores via counting the occurrences of *n*-grams in different elements. The leading *l* words of sentences/clauses are used to generate $\sum_{1 \le n \le m} l - n + 1$ *n*-grams, where $m < l$ and is a user defined variable. For example, a word sequence "a b c d" with $l=4$, $m=4$ can generate four 1-grams: a, b, c, and d; three 2-grams: a b, b c, and c d; two 3-grams: a b c and b c d; and one 4-grams: a b c d. With $l=4$, $m=3$, the word sequence can generate a, b, c, d, a b, b c, c d, a b c, and b c d, nine *n*-grams, $1 \le n \le 3$. We call the *m*-grams the sub-grams of the *n*-gram when they are generated from the same *n*-word sequence and $m < n$. The generation of varied sized *n*-grams creates a pool of noun phrase candidates. These noun phrases then become the bodies of the association rules. The association between the *n*-grams and different elements are evaluated by computing the confidence and support scores, basing on the occurrences of the *n*-grams in different elements in the training examples.

If each of the *n*-grams were counted as one occurrence as it's generated, then a single occurrence of an *n*-gram would actually be counted $\sum_{1 \le n \le m} l - n + 1$ times, all but one count comes from its sub-grams. This causes a problem: Suppose a b c is a noun phrase but any of its sub-grams is just a collocation-by-chance. If a b c is a significant indicator for an element, the counting method makes all its sub-grams frequent as well, while in fact the sub-grams alone may seldom appear in the element but frequently in another element. To avoid the problem, when the confidence and support scores of an *n*-gram are greater than a pair of pre-set values, its occurrence is deducted from the occurrences of all its sub-grams. The deduction prevents the sub-grams from becoming significant simply because their *n*-gram is significant. On the other hand, if an *n*-gram is just a collocation-by-chance and not significant, its occurrence will not affect much the discovery of any of its sub-grams as a significant indicator. The pre-set values should not be confused with the thresholds for the association rules. The former is set lower than the latter and they serve different purposes as described above. In the experiments reported in this paper, we empirically set $l = m = 3$, the pre-set value pair was set to 0.7 and 0, and the confidence threshold was set to 0.8 and support threshold was set to 0.035.

To mark up a new example, SCCP takes the first l words to generate $\sum_{1 \le n \le m} l - n + 1$ *n*-grams. By looking up the *n*-grams in the set of the learned association rules, SCCP obtains a number of matching rules whose confidence and support scores are above the thresholds. To decide which matching rule to use to mark up this segment, the matching rules are ranked according to the following set of criteria applied in that order: the size (i.e. *n*) of the *n*-gram, the location of the *n*-gram in the segment, the support score, and the confidence score. Rules containing longer *n*-grams are ranked higher than those with shorter ones. Rules with *n*-gram that locates at the beginning of the segment are ranked higher than others. Confidence score is the last criterion to check because the lookup procedure has already ensured that all retrieved rules have relatively high confidence scores. The top ranked rule determines the tag for the segment. The SCCP uses the nearest punctuation mark to determine the ending point of a text segment. The segment is consequently marked as the suggested element and the algorithm starts again to mark up the remainder of the text.

SVM: Support Vector Machine classification algorithm. We used SVM implementation in the Bow Toolkit (McCallum, 1996). We segmented plant descriptions by sentence for SVMs. Compared with classifying paragraphs (Cui, Heidorn & Zhang, 2002), the performance of SVM in classifying sentences is less desirable, as shown in the following section.

Experiments

In this section, we report selected experiments for evaluating the performance of different markup algorithms on plant morphological descriptions of three floras: FNA, FoC and FNCT. In these experiments, plant descriptions were marked up to the level of sub-organs of major organs such as leaves and flowers. The performance comparisons of different markup algorithms also showed the portability of the algorithms and their sensitivity to the size of training examples.

Data

A stratified random sampling method was used to select 310 descriptions from the 2,374 descriptions from the published volumes

of FNA[1] (vol. 2, 3, 4, 22, and 23). The same method was used to select 378 descriptions from 2,622 descriptions of FNCT[2]. The stratification was done on the genus level, resulting in the selection of at least one description from any genus. Due to the large size of FoC[3] collection (13,478 descriptions from the published volumes of FoC), the stratified random sampling method would have had produced around 1,000 training examples. Manually annotating this amount of examples was considered too costly. Instead, a simple random method was used to select 378 descriptions from FoC collection.

XML Schema

A XML schema for plant descriptions was created based on a number of existing sources, including DELTA format. The schema is too long to be presented in this paper, but can be access at this URL: http://publish.uwo.ca/~hcui7/research/xmlschema.xsd.

Performance Evaluation

The performance of the algorithms on three sets of plant descriptions was evaluated using 10-fold cross-validation. MARTT calculated precision and recall for every learning unit in the hierarchy. Precision was defined as the ratio of the number of segments that were marked as *E* correctly and the total number of segments marked as *E* by an algorithm. Recall was defined as the ratio of the number of the segments that are marked as *E* correctly by an algorithm and the total number of *E* segments in validation examples. F-measure with $\beta=1$ was also calculated and used to directly compare the performance of different markup algorithms.

Experiment Results

Although MARTT marked up plant descriptions to the level suggested by the training examples in one shot, for the sake of clear

presentation, we present markup performance element by element, starting with the root element *description*. We will then move on to present the markup performance on the sub-elements of *description*, using *leaves* element as an example.

Table 1 shows the average number of training examples for the sub-elements of *description* in 10-fold cross-validations.

Table 1. Element Distribution of the Training Sets in 10-Fold Cross-Validations

	FNA(279)	FoC(340.2)	FNCT(340.2)
phls	181.8	216.9	268
roots	25.2	27	5.4
buds	18.9	9.9	3.6
stems	207	250.2	99.9
leaves	266.4	308.7	243
flowers	178.2	310.5	276
pollen	0.9	0	0
fruits	172.8	209.7	160
cones	18	12.6	2.7
seeds	107.1	103.5	27.9
spore-related-structures	61.2	0	6.3
gametophytes	17.1	0	0
chromosome	171.9	47.7	2.7
phenology	0	242.1	211
compound	0.9	8.1	3.6
other-feature.	0.9	0.9	1.8
other-information	0	0	574

The markup performance of different algorithms was plotted in Figure 5, 6, and 7 for FNA, FoC, and FNCT training sets respectively. The performance plots show that SCCP had the best performance on all three sets of descriptions. For certain elements where the average number of training examples was 0.9, all four algorithms had zero performance. This was because these elements had exactly one training example in the entire training set. When the example was put in the validation set, an algorithm would not see any relevant training example in the training set. In short, the complete lack of training examples resulted in the zero performances on these elements. The performance plots also suggest that SCCP was less sensitive to small training sizes: All other algorithms had more zero performances than SCCP on elements whose training examples were sparse.

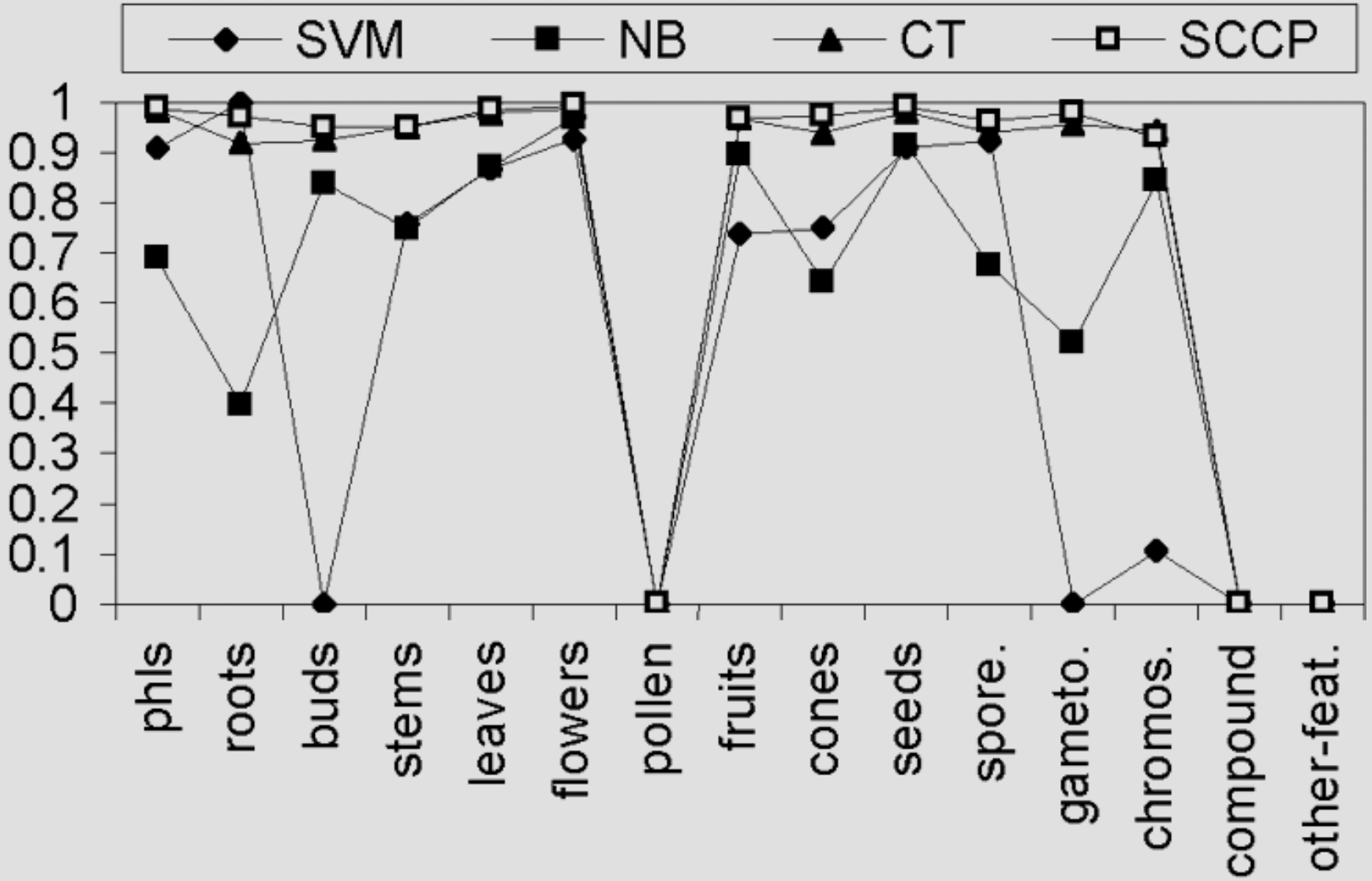

Figure 5: Performance Comparison of SVM, NB, CT and SCCP on Description Element in FNA Set

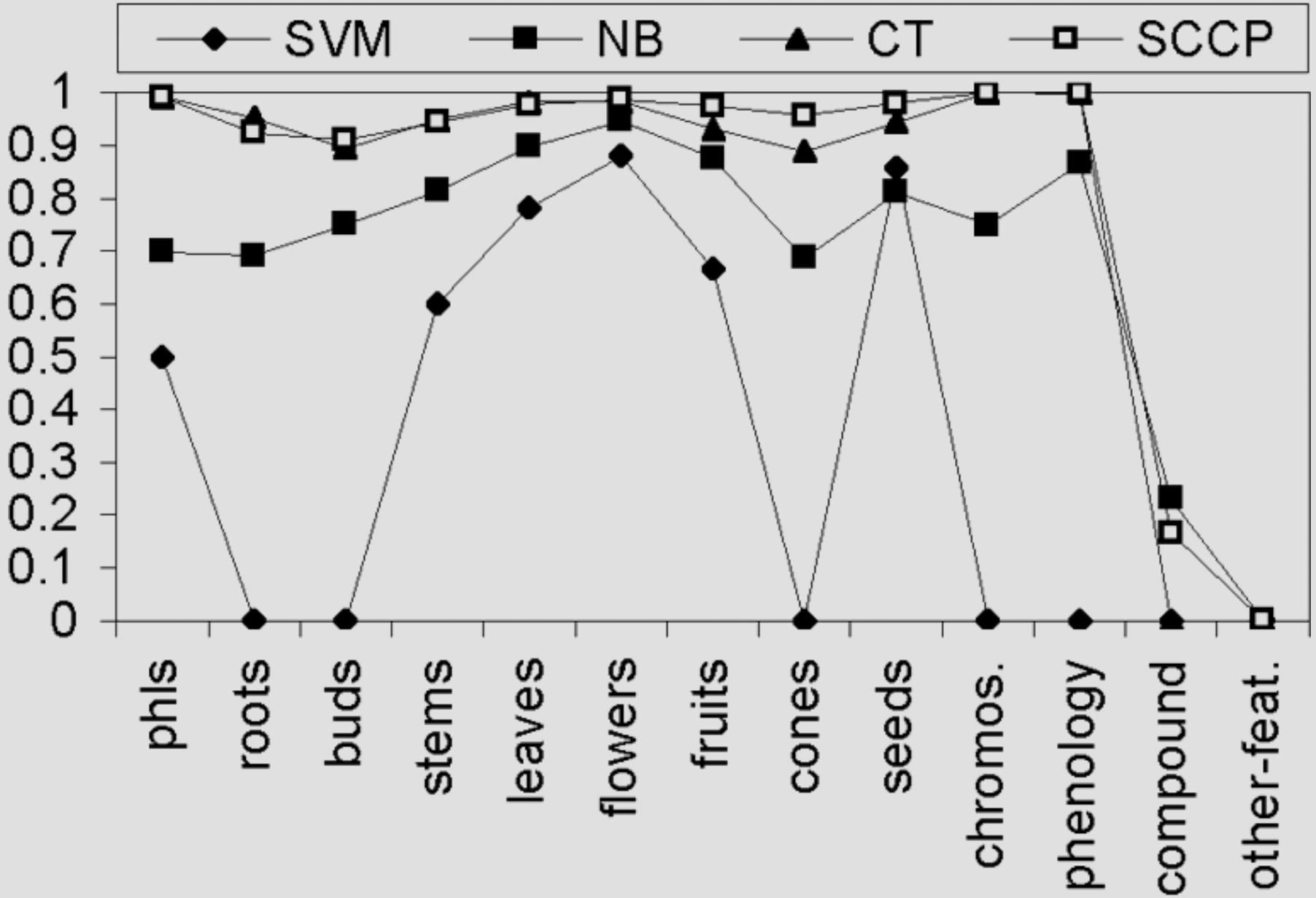

Figure 6: Performance Comparison of SVM, NB, CT and SCCP on Description Element in FoC Set

SVM
NB
CT
SCCP
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
phls
roots
buds
stems
leaves
flowers
fruits
cones
seeds
spore.
chromos.
phenology
compound
other-feat.
other-info.

Figure 7: Performance Comparison of SVM, NB, CT and SCCP on Description Element in FNCT Set

A comparison of the three plots clearly suggests that NB, CT, and SCCP had better performance on FNA and FoC than on FNCT data set. This may be explained by looking more carefully into the data sets themselves. Prior to the experiments, we evaluated the richness and consistency of the structural cues presented in the three data sets using a set of measures constructed by the author. The measurements suggested that FNA and FoC data sets seemed to be more structured than FNCT. The editorial policies of the floras contributed to this variation. Differ from FNA and FoC, the FNCT's editorial policy states that all plant characters that are included in the keys should not be repeated in the plant descriptions. The policy caused the descriptions in FNCT contain less consistent characters (because many common characters were described in the keys), resulting in larger variations from one description to another. This trait of FNCT helps to explain the high performance of SVM on certain elements (e.g. *roots*, *cones*, and *spore-related-structures*) in FNCT set, as SVMs are good at picking up distinctive features from even relatively small amount of training examples. The descriptions in FNCT are different from those in FNA and FoC on another dimension. As shown in Table 1, there is at least one *other-information* in every FNCT description. This was due to the layout of FNCT text: a taxon's morphological description and the information on distribution, habitat, and discussions are included one paragraph. The information other than morphological description was tagged as *other-information* in the markup. Because of the heterogeneous content of *other-information*, the need to distinguish this element from others made the markup task more difficult.

Due to space limitation, we present the markup performance measured at a deeper level using the *leaves* element as an example. Table 2 shows the element distribution in *leaves* elements across three training sets in 10-fold cross-validations. While there was a difference in element distributions in different data sets at the higher level markup, the variation at the deeper level was more evident. This was the case for other deeper level elements such as *stems*, *fruits*, and *flowers* etc. as well. At the deeper level, we also found more elements with single training example (i.e. training count = 0.9) or very few training examples.

Table 2. Element Distribution of the Training Sets in 10-fold Cross-Validations in Leaves Element

	FNA	FoC	FNCT
leaves	266.4	308.7	243
l/leaf-general	290.7	267.3	185.4
l/petiole	107.1	168.3	16.2
l/stipule	16.2	7.2	9
l/sheath	18	1.8	8.1
l/leaf-blade	231.3	218.7	69.3
l/leaflet-general	66.6	27	28.8
l/leaflet-blade	76.5	11.7	0
l/petiolule	1.8	4.5	0
l/rachis	16.2	1.8	0
l/buds	3.6	0.9	0
l/cataphyll	0	0.9	0
l/crownshaft	0.9	0	0
l/spine	0.9	0	8.1
l/tendril	0	2.7	2.7
l/ligule	0	2.7	9.9
l/gland	5.4	0	2.7
l/indumenta	9.9	0	0
l/abscission-zone	0	0.9	0
l/compound	4.5	0.9	0
l/other-features	27	0	0.9

The performance comparisons among different algorithms are plotted in Figure 8, 9, and 10 for FNA, FoC, and FNCT data sets respectively. SVMs were dropped from this markup level due to its low performance at the higher level markup.

Figure 8 shows when marking up the sub-elements of *leaves* in FNA descriptions, SCCP outperformed NB and CT. On each sub-element, the performance of SCCP was much higher than or equal to the performance of other algorithms. Note also the performance difference between CT and SCCP was enlarged as the markup moved to more detailed level: CT and SCCP had the similar performance on *leaves* elements (CT: precision=0.958 and recall=0.996; SCCP: precision=0.985 and recall=0.983), but the performance difference became much greater on sub-elements of *leaves* element such as *sheath* and *gland* etc.

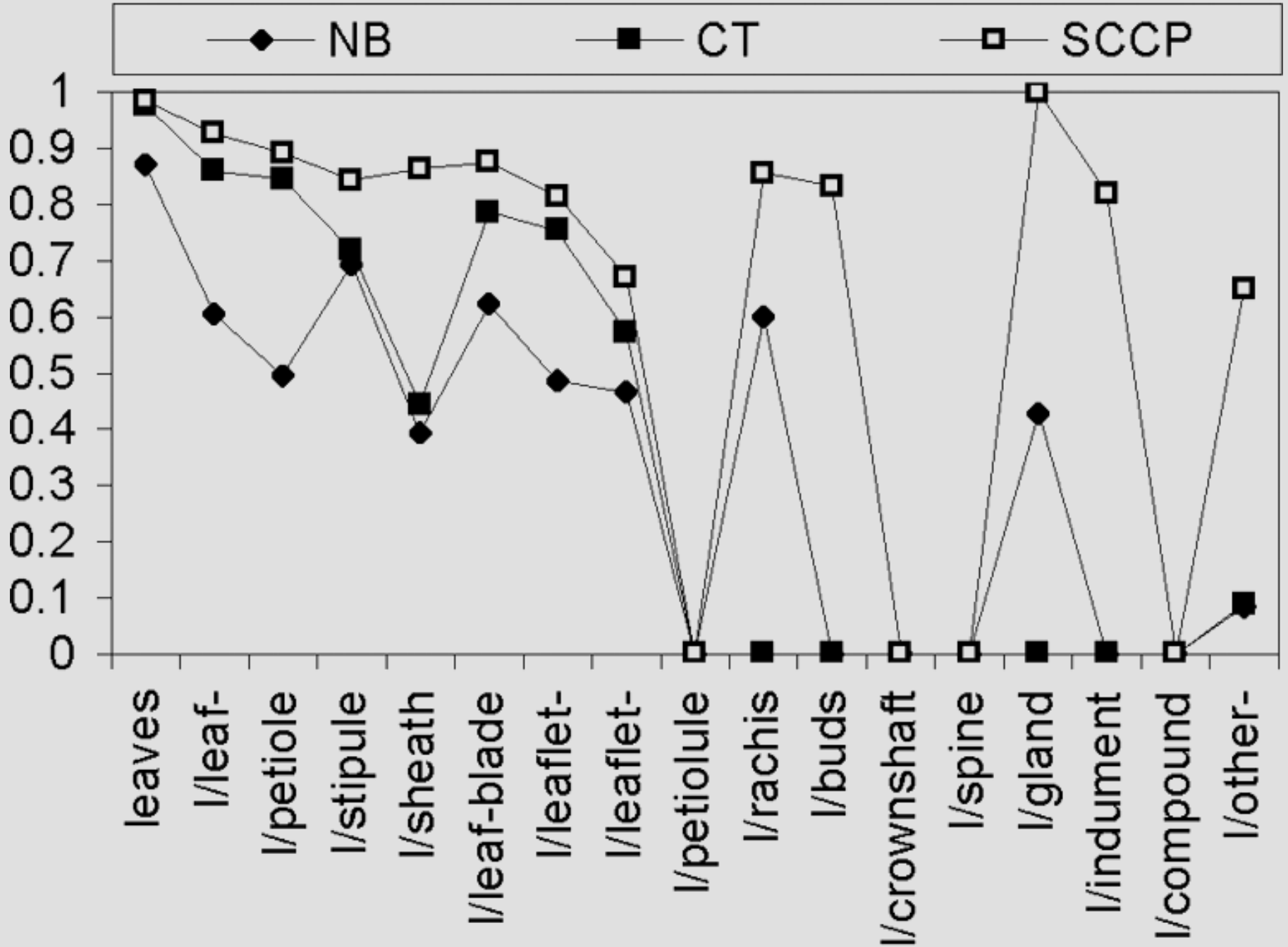

Figure 8: Performance Comparison of NB, CT and SCCP on Leaves Element in FNA Set

The performance plot also shows that SCCP was more robust when working with sub-elements with spare training examples. SCCP achieved very good performance for almost all elements but the extreme cases where less than 2 training examples were available. One exceptional element that drew our attention was *leaflet-blade* element. The difficulty with this element came from the fact that descriptive terms used for *leaf-blade* and *leaflet-blade* were extremely alike. To correctly identify the elements, MARTT needed to know the context: if a blade was described within the context of leaf description then it was a leaf blade; otherwise if the blade was described within the context of leaflet description then it was a leaflet blade. Unfortunately, current implementation of SCCP did not take the context into account.

Element *compound* was another difficult element. The tag *compound* was used to mark up a text segment containing descriptions of more than one organ, such as "scales and scars absent". The heterogeneous nature of the descriptions covered by this tag made it more difficult to learn. Making the difficult case even worse, the *compound* elements do not occur very often. Heterogeneous patterns with few training examples making this element very difficult to learn at both levels of markup.

Figure 9 shows the performance comparison in FoC data set. The plot shows some similar characteristics as Figure 8. SCCP had the best performance among the three algorithms and it was more robust in marking up elements with fewer training examples. Similarly, the results demonstrated that even though CT had slightly better performance (in F-measure) than SCCP on *leaves* elements (CT: precision=0.988 and recall=0.976; SCCP: precision=0.974 and recall=0.980), its performance was much worse on the sub-elements of *leaves*.

NB
CT
SCCP
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
leaves
l/leaf-
l/petiole
l/stipule
l/sheath
l/leaf-blade
l/leaflet-
l/leaflet-
l/petiolule
l/rachis
l/buds
l/cataphyll
l/tendril
l/ligule
l/abscissio
l/other-

Figure 9: Performance Comparison of NB, CT and SCCP on Leaves Element in FoC Set

The performance comparison on FNCT set demonstrated similar features as well (Figure 10). Differ from FNA and FoC set, fewer sub-elements of *leaves* were described in FNCT description. This was at least partially due to the editorial policy of FNCT as described earlier in this paper.

NB
CT
SCCP
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
leaves
l/leaf-general
l/petiole
l/stipule
l/sheath
l/leaf-blade
l/leaflet-general
l/spine
l/tendril
l/ligule
l/gland
l/compound

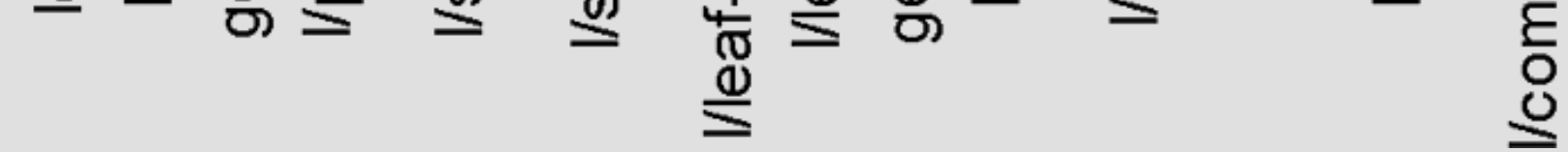

Figure 10: Performance Comparison of NB, CT and SCCP on Leaves Element in FNCT Set

In summary, the performance comparison at this level of markup was in large extent in agreement with the comparison at the higher level markup: SCCP outperformed NB and CT and was less sensitive to the lack of training examples. Comparing the performance of algorithms on different levels of markup, we noted a new trend: the small performance differences between SCCP and other algorithms at higher level markup were enlarged when the markup was performed at deeper level. The similar trends were observed in the deeper level markup in other elements such as *stems, flowers*, and *fruits* etc. as well.

The experiment results consistently suggested that SCCP was the best-performed algorithm of the four. In the subsequent experiments, MARTT used SCCP to mark up the available descriptions from FNA and FoC. It then mined the association rules and other potentially useful domain knowledge and conventions from the marked-up collections of FNA and FoC and populated the knowledge base. The experiments compared the markup performance of SCCP on FNCT set with and without the support from the knowledge base. The results suggested that the use of the knowledge base was effective in improving the markup performance on FNCT set. Details on these experiments will be reported in a later paper.

Discussion

The experiment results showed that SCCP worked relatively equally well across the three sets of plant descriptions and consistently outperformed NB and CT algorithms. In addition, SCCP was less sensitive to the lack of training examples. These findings demonstrated SCCP is a high performance markup algorithm with good portability and is robust to the limited training data.

Through the experiments, we also gained insights on two aspects of the automatic markup of plant descriptions. Firstly, despite the variations on element distributions across different data sets, we observed that the element distributions of different data sets are complementary to each other. While some collections are lacking certain elements, the elements occur quite frequently in other

collections. The complementary nature of the variations supports our hypothesis that domain knowledge learned from some collections is useful for the markup task of other collections.

Secondly, as the markup level goes deeper, the variations in element distribution become larger and more elements with few training examples appear. Manually preparing more training examples is not a good solution to this problem for three reasons: (1) it is tedious and time-consuming to prepare training examples. Having to provide training examples is a well-known weakness of supervised learning approach. It is more desirable to make supervised learning algorithm work with fewer training examples. (2) It is highly likely, in the domain of plant descriptions, providing more training examples would increase training examples for some elements, but at the same time introduce new elements with sparse training examples, due to the diversity of biological organisms. (3) It is highly likely, in the domain of plant descriptions, providing more training examples may solve the problem for the markup at certain level, but if the markup goes deeper, the problem would recur.

A more active approach to solve the problem of sparse training data is to invent algorithms that require less training examples. The experiment results have shown SCCP performed better than other algorithms on elements with spare training examples. However, the algorithm alone is not sufficient for solving the problem: as we have seen on many occasions in the experiments, an algorithm may need to mark up some descriptions for which it has never seen any training examples. The solution to this problem calls for external knowledge and support. The learned knowledge base seems to be useful in addressing this problem. Our further experiments showed that the domain knowledge and convention learned from the marked-up descriptions of FNA and FoC helped to improve the markup performance on FNCT descriptions, especially on elements with sparse or zero training examples.

Implications on Community Practice

Plant morphological descriptions contain information on the characters (e.g. leaf shape) and character states (e.g. “cordate”) of plant organs (e.g. leaves). Plant descriptions are the essential information source on which other significant work of taxonomy is built, for example, creating taxonomic keys for plant identification. A taxonomic key is a series of contrasting paired choices of character descriptions used to identify an unknown plant by the process of elimination. For the purpose of constructing taxonomic

keys, the descriptions must contain information for all the comparable characters. This parallelism is extremely difficult to achieve in practice. The authors may forget to mention certain characters/character states. Certain characters/character states may be implied: for example, for certain genera, it is common to assume certain character is non-existent unless it is explicitly described. It may be just infeasible for some genera to include all relevant information in a description because that would make the description excessively long. In addition, because plants in the same family or genus often share some common characters, the characters described in the family or genus descriptions typically are not repeated in individual species descriptions. The question is how to ensure the parallelism in plant descriptions given the situation. In other words, the problem is how to produce a complete description given the pieces of possibly implied information. Proposals had been made to use comprehensive forms or spreadsheets for the entry of descriptive data; however, the resistance from the taxonomist community was high. The resistance was due to, at least partially, the learning curve for adapting the new technology was high and the change of established day-to-day practice was never easy.

An automated system like MARTT can play a role in making this transition happen. One of the lessons we learned from the success stories of many software applications (for example, electronic spreadsheets) is to lower the learning curve by providing end-users a workspace similar to the one they are already familiar with. MARTT can be embedded in any work environment desired by taxonomists and invoked by a click of a button to convert a taxonomist-composed plant description to XML format. Once the descriptions are in XML format, it becomes easier to check for possible missing characters, to link family, genus, and species descriptions together, and to fill implied characters by comparing the structure of relevant descriptions in XML format. Much of the checking and comparison work can be done automatically. The author may choose to accept the XML document produced by MARTT, to edit certain portions to correct the few errors, or to simply ignore the document if there are too many errors. All these actions give feedback to the system and may be used to improve system performance. This is one of the directions for future research.

Notes

[1] http://www.fna.orgBack

[2] http://artemis.austincollege.edu/acad/bio/gdiggs/shinners.htmlBack

[3] http://flora.huh.harvard.edu/china/ Back

Acknowledgments

This project would not have been possible without the permissions of Hong Song, Barney Lipscomb, and George Diggs for the use of Flora of North America, Flora of China, and Illustrated Flora of North Central Taxes in this project. The author thanks Dr. David Boufford for his generosity in sharing his expertise in plant taxonomy and Professor P. Bryan Heidorn, Professor Linda Smith, and Professor Anhai Doan for informative discussions.

References

Abascal, R., & Sánchez, J.A. (1999). X-tract: structure extraction from botanical textual descriptions. In *Proceedings of the string processing & Information Retrieval Symposium and International Workshop on Groupware,* (pp. 2-7}.

Blum, S.D. (2000). An overview of biodiversity informatics. Retrieved April 1, 2004 from http://www.calacademy.org/research/informatics/sblum/pub/biodiv_informatics.html.

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

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

Han, J. & Kamber, M. (2001). *Data mining: concepts and techniques.* San Francisco: Morgan Kaufmann.

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

Lydon, S., Wood, M.M., Huxley, R., & Sutton, D. (2003). Data patterns in multiple botanical descriptions: Implications for automatic processing of legacy data.*Systematics and Biodiversity, 1(2),* 151-157.

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

Taylor, A. (1995). Extracting knowledge from biological descriptions. In *Proceedings of 2nd International Conference on Building and Sharing Very Large-Scale Knowledge Bases* (pp 114-119).

Thiele, K. (2003). SDD part 0: Introduction and primer to the SDD standard. Retrieved May 12, 2005 from http://160.45.63.11/Projects/TDWG-SDD/Primer/index.htm