A COMBINING APPROACH TO FIND ALL TAXON NAMES (FAT) IN LEGACY BIOSYSTEMATICS LITERATURE

GUIDO SAUTTER[1,3], KLEMENS BÖHM[1], AND DONAT AGOSTI[2]

[1] *Department of Computer Science, Universität Karlsruhe (TH), 76128 Karlsruhe, Germany;*
[2] *Division of Invertebrate Zoology, American Museum of Natural History, New York NY 10024-5192, and Naturmuseum der Burgergemeinde Bern, 3005 Bern Switzerland;*
[3] sautter@ipd.uka.de

Abstract. Most of the literature on natural history is hidden in millions of pages stacked up in our libraries. Various initiatives aim now at making these publications digitally accessible and searchable, applying xml-mark up technologies. The unique biological names play a crucial role to link content related to a particular taxon. Thus discovering and marking them up is extremely important. Since their manual extraction and markup is cumbersome and time-intensive, it needs be automated. In this paper, we present computational linguistics techniques and evaluate how they can help to extract taxonomic names automatically. We build on an existing approach for extraction of such names (Koning et al. 2005) and combine it with several other learning techniques. We apply them to the texts sequentially so that each technique can use the results from the preceding ones. In particular, we use structural rules, dynamic lexica with fuzzy lookups, and word-level language recognition. We use legacy documents from different sources and times as test bed for our evaluation. The experimental results for our combining approach (FAT) show greater than 99% precision and recall. They reveal the potential of computational linguistics techniques towards an automated markup of biosystematics publications.

INTRODUCTION

The Mass Digitization of biosystematics literature is becoming a major issue (e.g., Biodiversity Heritage Library, www.bhl.si.edu; American Museum of Natural History Digital Library; antbase.org). This body of literature with well over 10 Million pages contains all the descriptions of the world's biological taxa, that is the names and formal descriptions of the estimated 1.7 Million species known today and their higher categories (Maze 2004). The scientific names, Latinized binomen composed of a generic and a specific name (ICZN 2000: Article 5.1), are important. This is because they are unique within animals, plants, bacteria, virus and fungi, and their applications are ruled by respective codes (e.g., International Code of Zoological Nomenclature for animals). Each of these names belongs in a specific position within the taxonomic hierarchy. Within the life sciences, these scientific names are used to report the identity of the organisms upon which a study has been conducted. This potentially allows finding and linking all information on a particular species. Thus, recognizing taxonomic names is highly relevant for the digitization process, since no complete list of all the names of living organisms exists yet. Manual extraction of these names is time-consuming, e.g., 80 hours of manual extraction versus 330 seconds automatic extraction (Koning et al. 2005), and thus expensive. Automated name recognition and extraction is the ultimate solution. This article describes a combining approach for taxonomic name extraction, i.e., it combines several existing techniques from machine learning etc. We have dubbed our approach *FAT*, which is short for 'Finds all taxonomic names'. By reducing the average error of the base techniques by over 90%, our technique comes close to meeting the claim behind its name.

NAME EXTRACTION (TECHNIQUES)

Taxonomic names have some basic structural commonalities. The combination of its elements

(see Table 1) is not very restrictive and includes many optional parts and combinations. Some of these are no longer used, such as quadrinomen, a variety of a subspecies of a species of a genus. But nevertheless it is part of the history of names (see ICZN 2000 for legalistic aspects).

Part	Example 1	Example 2
Genus	*Prenolepis*	*Dolichoderus*
(Subgenus)	*(Nylanderia)*	
Species	*vividula*	*decollatus*
(Author)	*Nylander*	
(Subspecies)	*subsp. guatemalensis*	
(Author)	*Forel*	
(Variety)	*var. itinerans*	
(Author)	*Forel*	

Table 1: The parts of taxonomic names

For example, both "*Prenolepis (Nylanderia) vividula Nylander subsp. guatemalensis Forel var. itinerans Forel*" and "*Dolichoderus decollatus*" are taxonomic names. There are only two mandatory parts in such a name: the genus and the species name. Table 1 shows the deconstruction of the two examples. The parts with their names in brackets are optional. Formally, the rules of the Linnaean (Binominal) nomenclature define the structure of taxonomic names as follows, exemplified using animal names:

- The **genus** is mandatory. It is a capitalized word, often abbreviated by its first one or two letters, followed by a dot. In enumerations of several species of the same genus, the genus tends to appear explicitly only with the first species in the sequence.
- The **subgenus** is optional. It is a capitalized word. In most cases, it is enclosed in brackets, but not always.
- The **species** is mandatory. It is a lower case word, often followed by the name of the scientist who first described the species.
- The **subspecies** is optional. It is a lower case word as well, preceded by an indicator word like ***subsp.*** or ***subspecies***. It is often followed by the name of the scientist who first described the subspecies. In newer publications, the species is often abbreviated if a subspecies is given. In this case, the author name of the species is omitted. In addition, the indicator word can be omitted as well.
- The **variety** is optional. It is a lower case word, preceded by an indicator word like ***var.*** or ***variety***. It is often followed by the name of the scientist who first described it. Since 1960, however, the indicator word ***var.*** or ***variety*** is not permitted anymore (ICZN 2000).

The main problem for the automated recognition of these names is to distinguish them from the surrounding text, including other Named Entities (NE). Named Entity Recognition (NER) techniques can be employed to automatically identify scientific names (Chieu & Ng 2002). NER uses a variety of methods. Most common are gazetteers, grammars, rules, and statistical methods like Support Vector Machine (Bikel et al. 1997; Cuerzan & Yarowsky 1999; Mikheev et. al., 1999; Isozaki et al. 2002; Koning et al. 2005). Tjong et al. (2003) introduce two typical NER tasks: The names of locations, persons, and organizations are to be extracted. One may perceive taxonomic names as a special case of NE. But their structure is more complex and more variable than the one of 'typical' NE, e.g., location names, despite some basic shared elements, such as a Latin binomen surrounded by text in another language, the Latin binomen often not being part of existing dictionaries. Hence, common NER techniques tend to be too general to recognize taxonomic names. Newer tasks like the one presented by Carreras et al. (2005) do not consider more complex entities, but start dealing with relationships and semantic roles. Therefore, we do not have the hope that general NER research will turn to the extraction of complex entity names in the near future. Another problem of existing NER techniques is that they usually require pre-annotated training data (several hundred thousand words) to achieve good results (about 97 % precision and recall). – Besides NER, the following techniques are used to extract taxonomic names.

List-based NER techniques. Palmer & Day (1997) perform a lookup to determine whether a word is a NE of the category sought. The sole use of a thesaurus as a positive list is not an option for taxonomic names. All existing thesauri are incomplete. Nevertheless, such a list allows recognizing known parts of taxonomic names.

The inverse approach would be a list-based exclusion technique, e.g., a common English thesaurus like WordNet serves as a list of known negatives. This in isolation is not an option either. It would not exclude proper names reliably. Next, it would exclude parts of taxonomic names that also happen to be used in common English. This was the reason for the majority of errors in the evaluation of TaxonGrab (Koning et al. 2005), which combines

list-based exclusion with some rules. However, exclusion of sure negatives, i.e., words that are never part of taxonomic names, simplifies the classification process.

Rule-based techniques do not require any training data. Instead, they try to find words or word sequences with a certain structure, e.g., regarding punctuation. Yoshida et al. (1999) presents a technique that extracts the names of proteins and their abbreviations based on regular expressions. It makes use of the very distinctive syntax of protein names, e.g., "*NG-monomethyl-L-arginine*".

The syntax of taxonomic names is subject to certain rules as well, but they are less restrictive. Due to the wide range of optional parts (see Tab. 1), it is impossible to find a regular expression that matches all taxonomic names and at the same time provides a satisfactory degree of precision. Koning et al. (2005) present an approach based on regular expressions and lexica. This technique (called TaxonGrab) performs satisfactorily compared to common NER approaches. But the conception of what is a positive is restricted. For instance, it simply leaves aside taxonomic names that do not specify a genus. However, the general idea of using rules to filter the phrases of documents is helpful.

Bootstrapping. Jones et al. (1999) describe an approach to training classifiers without large amounts of labeled training data. Some labeled seed data and a large unlabeled training corpus is taken as input. Learning from the seed data yields automatic labeling of the corpus. Jones et al. (1999) have shown that the performance of this approach is equal to the one of other techniques that require large amounts of labeled training data. Bootstrapping is not readily applicable to our particular problem, however. Niu et al. (2003) use an unlabeled corpus of 88,000,000 words to bootstrap a named entity recognizer. For our purpose, even unlabeled training data is not available in this order of magnitude, at least right now.

Active Learning. The intention behind Active Learning (Day et al. 1997) is to speed up the creation of large labeled training corpora from unlabeled documents. In particular, the system uses all of its knowledge during all phases of the processing. In this way, it can label many data items automatically, and the user has to label only pathologic cases. To increase data quality, such a user-interactive approach should be part of a taxonomic-name extractor as well. We make use of this approach in two ways: First, the output of each step serves as base data for the subsequent ones. Second, the user manually classifies the few remaining cases after these automated steps. Following the general idea of active learning, we feed these manual classifications back into the base data. The algorithm can then use them later when processing other documents. This improves the performance of the algorithm at runtime. In our evaluation, we will use a measure that quantifies the number of user interactions. This is to enable comparison to other components.

Word Language Recognition. Language Recognition is intended to determine the language a given text is written in. (Sautter & Böhm 2006) have shown that these techniques can be used to extract parts of taxonomic names from English text. In particular, modifications have been made to the standard techniques so that little training data is required and becomes applicable on word level. The technique is based on two statistics containing the N-Gram distribution of taxonomic names and of common English. Both statistics are built from examples from the respective languages. It applies Active Learning to reduce the need for annotated training data. The classifier is tunable towards precision or recall, as needed. In optimal configurations, both reach a level of 96%. This is the typical level of common up-to-date NER components. The Active Learning requires the user to classify about 3% of the words manually. Although this is relatively low, compared to manually annotating an entire text, the absolute number of user interactions is still high. In addition, some training data is needed. Thus, other techniques are used to (a) gather the required training examples and to (b) reduce the input to this classifier as far as possible. In particular, it should be used only to deal with word sequences that cannot be labeled safely with the other techniques.

Gene and Protein Name Extraction. The major focus of NER in biomedicine is the extraction of gene and protein names. Tanabe & Wilbur (2002) give a wide overview of the techniques used for this purpose. The most frequently used approaches are Hidden Markov Models, lexicon lookups and structural rules. Many of the techniques also include a Part-Of-Speech tagger and use its output as additional evidence. However, there are significant differences between gene and protein names on the one hand and taxonomic names on the other hand: First the nomenclature rules for the latter are by far less restrictive and include a wide range of optional parts. For instance, they may include the names of the discoverer/author of a given part. Second, there are parts of gene and protein names

which are easy to distinguish from the surrounding text because of their structure. For the extraction of taxonomic names, we cannot rely on this type of evidence. Consequently, the techniques for gene or protein name recognition are not feasible for the extraction of taxonomic names.

An individual technique in isolation thus might not be sufficient for taxonomic name extraction. Mikheev et al. (1999) have shown that a combining approach, i.e., one that integrates the results of several different techniques, is superior to the individual techniques for common NER. For this reason, we combine approaches for taxonomic name extraction.

Due to the active learning, the word-level language recognizer needs little training data. In addition, the manual effort induced by user interactions is high. Thus, other techniques need be applied beforehand, for the following two reasons: First, to find sufficient training examples for the word-level classifier. Second, to reduce the input to the classifier to as few words as possible. This last aspect is based on the idea to prevent as many words as possible from being prompted to the user to further reduce the manual effort.

Usage of the typical structure of taxonomic names allows achieving both goals. Syntax-based rules are used to extract training examples from the documents. This leads to a reduction of the number of words the classifier has to deal with. However, it is not possible to find rules that extract taxonomic names with both high precision and recall, as we will show later. But we have found rules that fulfill one of these requirements very well. In what follows, we refer to these as **precision rules** and **recall rules**, respectively.

MATERIAL AND METHODS

The Classification Process

The general idea of our approach is first to extract or exclude those parts of the text for which we can be sure about that they are either taxonomic names or not (Precision and Recall Rules in Fig. 1). We then use the parts already classified to build lexica and statistics, which we use to classify the rest of the text (Data Rules and Word Classifier in Fig. 1). If there are still uncertain parts left after this step, we present them to the user for manual classification (User Feedback in Fig. 1). In more detail, our approach works as follows:

1. In a first pass through the document, we apply the precision rules. Every word sequence from the document that matches such a rule is a *sure positive*.
2. In a second pass, we apply the recall rules to the phrases that are not sure positives. A phrase not matching one of these rules is a *sure negative*.
3. Third, we build lexica from the sure positives and sure negatives, and apply them in several ways to the phrases that are still uncertain. For instance, we filter out word sequences that contain at least one known negative word.
4. We collect a set of names from the set of sure positives. We then use these names to both include and exclude further word sequences.
5. We train the word-level language recognizer with the surely positive and surely negative words. We then use the language recognizer to classify word/phrases that are still uncertain.

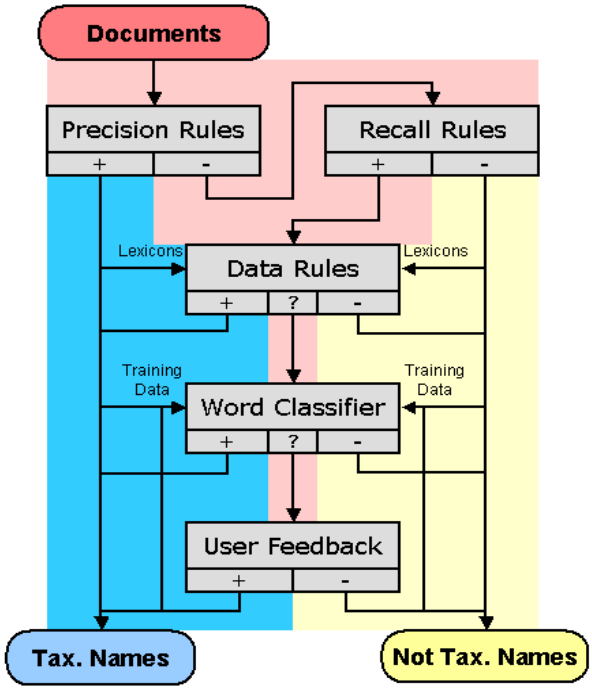

Figure 1: The Classification Process

Figure 1 visualizes the classification process. Red areas mark the flow of words/phrases which are uncertain at several stages, blue areas mark sure positives, while yellow areas mark sure negatives. The boxes with round corners represent sets of words/phrases, colored according to their state. Initially, all words in the text (Documents) are uncertain. After FAT has finished, all words/phrases are classified as sure positives (Tax. Names) or sure negatives (Not Tax. Names). The gray boxes represent the different steps of the FAT algorithm;

the arrows depict the data flow. More specifically, the meaning of the arrows also depends on where an incoming arrow meets a box: An arrow meeting the box at its top represents data the step has to process. Arrows going to the side of a box stand for words/phrases already classified and now serving as additional input. The state of the words/phrases after a step is visualized at the bottom of a box: + indicates that a word/phrase has matched the rule, - indicates the opposite, and ? indicates that the particular step could not classify the word or phrase with certainty. The background color of the area behind the outgoing arrows also emphasizes this. An arrow that splits indicates that data goes two ways. As data comes out of the User Feedback step and is finally classified, for instance, it goes into the sets of sure positives and sure negatives, respectively. Additionally, the Word Classifier receives it as additional training data. Two joining arrows signal that data comes from two sources. The training data for the Word Classifier, for example, comes from the sure positives and negatives as well as from the words/ phrases classified in the User Feedback step.

The order of application enables the different techniques to profit from each other: The Precision and Recall Rules extract the base data for the subsequent steps, so there is no need for training data at all. Using the Data Rules and the Word Classifier before them would require manual preparation of lexica and training data for the classifier. So inverting our proposed order of application is not feasible. When processing a document, the FAT algorithm does one pass applying the precision and recall rules. At the same time it collects the sure positives, candidates, and sure negatives. All further steps base on this initial trisection of the text. Only the recall rule using the set of scientist names (see below) requires one further pass over the document.

This approach is somewhat similar to the bootstrapping algorithm proposed by Jones et al. (1999). The difference is that this process works solely with the document it actually processes. In particular, it does not need any external data or a training phase. The 107 documents forming the A.M.N. (American Museum Novitates) part of our test bed count about 8,100 words on average, which is less than 0.02% of the data used by Niu et al. (2003). The entire test bed has less than 2,500,000 words, still less than 2% of the corpus used by Niu et al. (2003). On the other hand, with the classification process proposed here, the accuracy of the underlying classifier must be very high from the start. This is because we do not have a training phase, but start from scratch with the first document we process.

Rules for Structure of Taxonomic Names

In order to make use of the structure of taxonomic names, we use rules that refer to this structure, see Tab. 2. The syntax used here is the one of the JAVA programming language, documented in the JAVA online documentation (JAVA 1.4.2). We use regular expressions for the formal representation of the rules. In this section, we develop a regular expression matching any word sequence that conforms to the Linnaean rules of nomenclature (see 3.3).

_	one white space character
<LcW>	[a-z]$^{(3,)}$
<LcA>	[a-z]$^{(1,2)}$.
<CapW>	[A-Z][a-z]$^{(2,)}$
<CapA>	[A-Z]{[a-z]}?.
<Name>	{<CapA>_}$^{(0,2)}$<CapW>

Table 2: Abbreviations.

The taxonomic names are modeled as follows:

- The genus part of a taxonomic name is a capitalized word, often abbreviated by its first one or two letters, followed by a dot. We denote it as <genus>, which stands for {<CapW>|<CapA>}.
- The subgenus part of a taxonomic name is a capitalized word, optionally surrounded by brackets. We denote it as <subGenus>, which stands for <CapW>|(<CapW>).
- The species part of a taxonomic name is the name of the species, a lower case word, optionally followed by a name. In newer publications, on the other hand, the species is often abbreviated if a subspecies is given. In this case, the name is omitted. We denote this structure as <species>, which stands for {<LcW>{_<Name>}?|<LcA>}.
- The subspecies part of a taxonomic name is a lower case word, preceded by the indicator word ***subsp.*** or ***subspecies***, and optionally followed by a name. In newer publications, however, the subspecies is often abbreviated if a variety is given. In this case, the name is omitted. In addition, the indicator word ***subsp.*** or ***subspecies*** can be omitted as well. We denote this structure as <subSpecies>, standing for {{{{subsp.|subspecies}_}?<LcW>{_<Name>}?}|<LcA>}.

- The variety part of a taxonomic name is a lower case word, preceded by the indicator word ***var.*** or ***variety***, and optionally followed by a name. In newer publications, however, the indicator word ***var.*** or ***variety*** can be omitted. We denote this structure it as <variety>, which stands for {{{var.|variety}_}?<LcW> {_<Name>}?}.

A taxonomic name is now modeled as follows. We refer to the pattern as <taxName>:

<genus>{_<subGenus>}?
<species>{<subSpecies>}?
{_<variety>}?

Precision Rules

Because <taxName> matches any sequence of words that conforms to the Linnaean rules, it is not very precise. The simplest match is a capitalized word followed by one or more in lower case. Any two words at the beginning of a sentence are a match! Thus, to have less false positives, we need more precise regular expressions. To accomplish this, we rely on the optional parts of taxonomic names. In particular, we classify a sequence of words as a sure positive if it contains at least one of the optional parts <subGenus>, <subSpecies> and <variety>. For the last two, we additionally demand the subspecies or variety to be explicitly labeled or the part before them to be abbreviated. The second restriction is as secure as the first one. The reason is that normal text rarely continues in lower case after a dot. The hope is to exclude almost all phrases that are not taxonomic names. Even though our regular expressions may classify a sequence of words as a sure positive erroneously, our evaluation will show that this happens very rarely. Our set of precise regular expressions has three elements:

- <taxName> with subgenus in brackets, <subspecies> and <variety> optional:
  <genus>_(<CapW>)
  <species>{<subSpecies>}?
  {_<variety>}?
- <taxName> with <subspecies> given, <subGenus> and <variety> optional:
  <genus>{_<subGenus>}?
  <species><subSpecies>
  {_<variety>}?
- <taxName> with <variety> mandatory, <subGenus> and <subSpecies> optional:
  <genus>{_<subGenus>}?
  <species>{<subSpecies>}?
  {_<variety>}

To classify a word sequence as a sure positive if it matches *at least one* of these regular expressions, we combine them disjunctively and call the result <preciseTaxName>. It matches any sequence of words we can classify as a taxonomic name simply because of its structure. When applying the precision rules, we test phrases of up to 10 words, plus punctuation.

In many taxonomic publications, new genera, species, etc. are explicitly labeled. If *Dolichoderus decollatus* is described for the first time, for instance, it is likely to labeled as a new species somewhere. The title of the description would be *Dolichoderus decollatus, new species*. We use special forms of the precision rules to make use of these labels. In particular, we consider a match of <taxName> a sure positive if it directly precedes a label in the text. Because they rely on explicit labels, we refer to these special precision rules as *Label Rules*.

A notion related to that of a sure positive is the one of a *surely positive word*. A surely positive word is a part of a taxonomic name that is not part of a scientist's name. For instance, the taxonomic name *Prenolepis (Nylanderia) vividula Erin subsp. guatemalensis Forel var. itinerans Forel* contains the surely positive words *Prenolepis*, *Nylanderia*, *vividula*, *guatemalensis*, and *itinerans*. Further steps of our process assume that surely positive words exclusively appear as parts of taxonomic names.

Recall Rules

The recall rules basically consist of <taxName>, which matches any sequence of words that conforms to the Linnaean rules. When applying it, we again test phrases of up to 10 words, plus punctuation. But there is a further issue: Enumerations of several species of the same genus tend to contain the genus only once. For instance, in *Pseudomyrma (Minimyrma) arboris-sanctae Emery, latinoda Mayr and tachigalide Forel* we want to extract *latinoda Mayr* and *tachigalide Forel* as well. To address this, we make use of the surely positive words: We use them to extract parts of taxonomic names that lack the genus.

We also extract the names of the scientists from the sure positives and collect them in an extra list (name lexicon). We regard a capitalized word in a sure positive as a name if it comes after the second position. In the example, we would extract *Pseudomyrma*, *Minimyrma* and *arboris-sanctae* from

the sure positive *Pseudomyrma (Minimyrma) arboris-sanctae Emery*. We would also add *Emery* to the set of names.

We cannot be sure that the list of sure positive words suffices to find all species names in an enumeration. Hence, we additionally collect all lower-case words followed by a capitalized word contained in the set of names. In the example, we need to have *Mayr* and *Forel* in the set of names to extract *latinoda Mayr* and *tachigalide Forel*.

Data Rules

Because we want to achieve close to 100% in recall, the recall rules are minimally restrictive. Consequently, many word sequences that are not taxonomic names are considered uncertain. Before the word-level language recognizer deals with them, we explore some more ways to find negatives. Because the precision rules are very restrictive, they match only a fraction of the taxonomic names in a text. Making use of the sure positive words, we can also find additional sure positives.

Sure Negatives. As previously mentioned, <taxName> matches any capitalized word followed by a word in lower case. This includes the start of any sentence. But making use of the sure negatives, we can recognize these phrases. In particular, we classify any word sequence as negative that contains a word which is also in the set of sure negatives. For instance, in sentence "*Additional evidence results from …*", "*Additional evidence*" matches <taxName>. Another sentence contains "*… an additional advantage …*", which does not match <taxName>. Thus, the set of sure negatives contains "*an*", "*additional*", and "*advantage*". Knowing that "*additional*" is a sure negative, we exclude the phrase "*Additional evidence*".

Names of Scientists. Though the names of scientists are valid parts of taxonomic names, they may also cause false matches. A misclassification occurs when they are matched with the genus or subgenus part – <taxName> cannot exclude this. In addition, they might appear elsewhere in the text without belonging to a taxonomic name. Similarly to sure negatives, we exclude a word sequence if the first or second word is contained in the set of names. For instance, in "*…, and Forel further concludes …*", "*Forel further*" matches <taxName>. If the set of names contains "*Forel*", we can exclude "*Forel further*". This is because we know that "*Forel*" is not the name of a taxonomic genus.

Sure Positives. Making use of the sure positives we have extracted with the precision rules, we can find additional sure positives. In particular, we mark an uncertain word sequence as a sure positive if it consists of surely positive words or abbreviations. If the precision rules have extracted *Prenolepis (Nylanderia) vividula Erin subsp. guatemalensis Forel var. itinerans Forel*, for instance, we conclude that *Prenolepis vividula* is a sure positive as well.

Stemming Lookup Catch. Koning et al. (2005) used a common English dictionary to exclude negatives. As mentioned in the introduction, this leads to the exclusion of taxonomic names containing a common English word. For instance, this would exclude the taxonomic name *Formica minor* because of *minor*. On the other hand, such a dictionary-based exclusion can help to catch the (very few, but still existing) erroneous matches of the regular expressions in <preciseTaxName>. Our observation shows that if a common English word is part of a taxonomic name, it is always used in its base form. Thus, if we find a stemmed form of a word in a dictionary, we conclude that it is not part of a taxonomic name. Consider the following sentence from an essay on dangerous insects: … *In Chaguanas (Trinidad) another subspecies poisoned Forel.* Except for the word *In*, this sentence matches the regular expression from <preciseTaxName> where <subSpecies> is mandatory. But we can recognize it as a false match because of the conjugated verb *poisoned* in the subspecies position. Similar pathologic cases can occur for the variety part. But all these cases share a useful feature: They comprise *modified forms* of common English words. Thus, to exclude these errors, we combine the dictionary lookup with stemming: For all the words matched to a lower case part of a regular expression in <preciseTaxName> (<species>, <subSpecies> or <variety>), we check if it could be a conjugated verb given its ending (most common endings are *-s*, *-ed*, and *-ing*). If so, we apply stemming in order to obtain the infinitive. If contained in a common English dictionary, we exclude the match. In the example, the ending rule applies to *poisoned*. Porter's (1980) stemming algorithm produces *poison* as the word stem. Because this word is contained in the dictionary, we can exclude the erroneous match. In *In Chaguanas (Trinidad) another subspecies poisoned Forel* the stemming lookup catch is the only data rule that we also apply to the matches of the regular expression from <preciseTaxName>.

Name Completion

Making use of the scientists' names, we also extract taxonomic names that lack the genus, e.g., from enumerations, such as *Pheidole pallidula, orbula, xantra*. In addition, the rules allow genus abbreviations like *Ph.* for *Pheidole* in *Ph. cornutula*. In order to determine the meaning of a taxonomic name, we need to complete the names with their full parts.

If the genus part is missing, we have two options: First, we check if the species part appears elsewhere in the document, together with the genus it belongs to. If this is not the case, we use the last genus that we have extracted before the position of the name to complete. This is useful especially in case of enumerations: If several species of the same genus are enumerated, the genus is often given only with the first one. We then transfer the genus part to the subsequent taxon names.

If the genus is abbreviated, we also have two options: First, we again check if the species part appears elsewhere in the document, together with the full name of the genus it belongs to. If this fails, we check if we have recognized any genus name that starts with the given abbreviation. If there is exactly one such genus name, we insert it. If there is more than one, i.e., the abbreviation is ambiguous, we use the one which appears closest before the abbreviation.

Classification of Remaining Words

After applying the various rules, uncertain word sequences still remain. To deal with them, we use word-level language recognition (Sautter & Böhm 2006), a technique to classify words as parts of taxonomic names or as common English, respectively. It is based on two statistics containing the N-Gram distribution of taxonomic names and of common English, that is how often short sequences of letters occur in a group of words in a text (e.g., 4-Gram for Formica = {Form, ormi, rmic, mica}). Both statistics are built from examples from the respective languages. This technique achieves about 96% in precision and recall. It involves the user in the classification process to back up narrow decisions with the human expert knowledge. To train the classifier, we use the surely positive and surely negative words as the training data. Instead of classifying every word separately, we compute the word-level classification score of all words of a sequence and then classify the sequence as a whole. This has several advantages: First, if one word of a sequence is uncertain, this does not automatically incur a user-feedback request. Second, if a sequence of words is uncertain as a whole, the user gives feedback for the entire sequence. This results in several surely classified uncertain words at the cost of only one feedback request. In addition, a user can easier determine the meaning of a sequence of words than the one of a single word.

Experimental Setup and Test Bed

We run two series of experiments: We first process each document individually. We then process the documents incrementally, i.e., we do not clear the sets of known positives and negatives after each document. The same is true for the statistics of the word-level language recognizer. This is to measure the benefit of reusing data obtained from one document in the processing of subsequent ones. Finally, we take a closer look at the effects of the individual steps and heuristics.

The platform is completely implemented in **JAVA 1.4.2**, and we have used the **java.util.regex** package to represent the rules.

All tests presented here are based on three groups of annotated documents. First, we use 107 issues of the *American Museum Novitates* (A.M.N.), a natural science periodical published by the *American Museum of Natural History*. The second group is a recent publication representing a widely used standard in ant systematics (F.2000, Fisher 2000), and the third one is the Birds of Congo (C.1932: Chapin 1932, 1939, 1953, 1954), partitioned into four parts of similar size, which was used by Koning et al, 2005. Koning et al.'s test bed has been extended to include additional groups of publications with different ways of combining and abbreviating names. Table 3 contains the relevant numbers on our test bed (rounded), the numbers on the A.M.N. and F.2000 parts are the result of manual counting, those on C.1932 originate from (Koning et al. 2005).

	A. M. N.	**F. 2000**	**C. 1932**
Words	857,000	58,000	1,100,000
Taxonomic Names	12,000	175	21,000

Table 3: The Test Bed

Evaluation Measure

In NLP, the f-Measure is popular to quantify the performance of a word classifier, but we also need to measure the advantage the system gains from asking the user for feedback on narrow classifica-

tions. In particular, we use three measures to quantify our test results. As mentioned, the first one is the f-Measure:

```
P(P) := positives classified as positive
N(P) := positives classified as negative
P(N) := negatives classified as positive
N(N) := negatives classified as negative
```

$$\text{Precision}\quad p := \frac{P(P)}{P(P) + P(N)}$$

$$\text{Recall}\quad r := \frac{P(P)}{P(P) + N(P)}$$

$$\text{fMeasure} := \frac{2 \times p \times r}{p + r}$$

But our combined technique has three possible outputs. If the decision between positive or negative is narrow, a word is classified as uncertain, and the user is prompted. This prevents misclassifications and thus induces a considerable advantage over fully automated techniques. In order to enable comparison to fully automated techniques, we use two further measures:

```
U(P) := positives not classified (uncertain)
U(N) := negatives not classified (uncertain)
```

Given this, *Coverage C* is defined as the fraction of all classifications that are not uncertain:

$$C := \frac{P(P) + N(P) + P(N) + N(N)}{P(P) + N(P) + U(P) + P(N) + N(N) + U(N)}$$

To combine these two measures to a single measure for overall classification quality, we multiply f-Measure and coverage and define *Quality Q* as

$$Q := \text{fMeasure} \times C$$

This measure treats all uncertain votes as misclassifications, thus punishing every user interaction as if it was an error. This enables comparison to techniques that do not involve the user. It is very restrictive because a random guess might result in at least half of the uncertain words classified correctly. On the other hand, a correct vote from the user avoids misclassifications. In learning components, this keeps statistics clean because no errors are fed back.

EVALUATION AND DISCUSSION

A combining approach gives rise to many questions in the context of taxonomic name extraction, e.g.: How does a word-level classifier perform with training data automatically generated? How does rule-based filtering affect precision, recall, and coverage? What is the effect of extending the lexica dynamically? Which kinds of errors remain?

Tests with Individual Documents

First, we test the combined classifier with each document individually. We omit Fisher (2000) here because it consists of only one document. The results for this part of the test bed will be presented in the next section. Table 4 contains the average results for the A.M.N. and C.1932. The combination of rules, dynamic lexica, and word-level classification provides very high precision and recall. The former is 99.7% on average, the latter 98.2%. The need for manual intervention is very low: The average coverage is 99.7%.

	A. M. N.		**C.1932**	
	Doc	**Sum**	**Doc**	**Sum**
Words	857,000		1,100,000	
Taxonomic Names	12,600		22,500	
Sure Pos.	24	2,528	2,833	11,331
Uncertain	356	38,177	11,545	46,179
Data Rules SP	85	9148	4983	19933
Data Rules UC	42	4475	674	2,697
Scorings	17	1,836	181	723
Precision	93.1%	97.5%	92.7%	99.7%
Recall	55.2%	93.3%	97.8%	99.8%
f-Measure	56.8%	95.4%	95.0%	99.7%
Coverage	87.8%	99.0%	96.6%	99.8%
Quality	54.4%	94.4%	91.9%	99.6%

Table 4: Test with Individual Documents

The average results with individual issues of the A.M.N. (Column **Doc** in Table 4) are significantly worse than with the other two parts of the test bed. A more detailed look at the results reveals significant differences between the individual documents. For more than half of the documents, the results are equal to those of the other two parts of our test bed. The rest of the A.M.N. issues points out a weakness of our combined technique: If the precision rules do not extract a sufficient number of sure positives, we run into two problems. We neither have enough data to successfully apply the data rules, nor do we have enough positive examples to train the word level classifier. The **Sum** Column in Table 4 contains the summed up results for the A.M.N. It turns out that precision, recall and coverage are far better for the total numbers than the average per document. This is because the documents with few taxonomic names in them produce the poor results, while our combined technique performs better with the bigger documents. For C.1932, this effect is almost non-existent. The individual parts are big enough and contain sufficiently many sure positives for our technique to succeed.

Tests with Entire Corpus

In the first test the classifier did not transfer any experience from one document to later ones. We now process the documents one after another, de facto concatenating all the documents to one big super-document, which is then analyzed as a whole. Table 5 shows the results. As expected, the classifier performs better than with individual documents. This is true for both the A.M.N. and C.1932 test documents. The average recall increases to 99.9%, coverage improves to 99.8% on average. Precision increases to an average of 99.5%.

	A. M. N.	**F. 2000**	**C. 1932**
Words	857,000	58,000	1,100,000
Taxonomic Names	12,600	175	22,500
Sure Pos.	3,059	172	13,827
Uncertain	37,028	2,368	42,180
Data Rules SP	11,819	175	2,2084
Data Rules UC	1,132	2	583
Scorings	618	1	295
Precision	99.2%	100%	99.8%
Recall	99.9%	100%	99.9%
f-Measure	99.6%	100%	99.8%
Coverage	99.5%	100%	99.9%
Quality	99.1%	100%	99.7%

Table 5: Test with Corpora

The effect of the incremental learning is obvious, especially for the A.M.N. part of the test bed: The false positives are less than 2% of those in the first test shown by a comparison of the recall values in Tables 4 and 5. The effect on precision is significant as well: The number of false negatives is only a third of that in the first test. Finally, the number of words for which the technique has to ask for feedback is halved (compare coverage values).

The reason for the improvement is obvious from documents where the number of word sequences in <preciseTaxName> is low: data from other documents compensates the lack of positive examples. This reduces the number of false positives and false negatives as well as the user interactions.

The Data Rules

The Lines *Uncertain* in Tables 4 and 5 contain the number of uncertain phrases after the application of the regular expressions, the Lines *Data Rules UC* display how many phrases remain uncertain after the data rules were applied. The exclusion of word sequences containing a sure negative turns out to be effective to filter the matches of <taxName>. On average, this step reduces the number of uncertain word sequences by about 75%.

The Lines *Sure Pos.* and *Data Rules SP.*, in turn, provide the number of sure positives after the regular expressions and after the data rules, respectively. The data rules based on the sure positives are very effective as well: they reduce the uncertain word sequences by another 15 %, and at the same time enlarge the set of sure positives by 50% on average. In particular, they do not only reduce the uncertain sequences, but also obtain additional training data for the word level classifier.

The lines labeled *Scorings* display the number of distinct phrases that were classified by the statistical component. Our experiments show that the manual effort incurred by uncertain statistical classifications and subsequent user feedback decreases significantly. All four data rules decrease the number of words the language recognizer has to deal with. This is because they produce additional training data and reduce the number of words classified as uncertain.

Comparison to Word-Level Classifier and TaxonGrab

A word-level classifier (WLC) is the core component of FAT. We compare it in standalone use to the combining technique (Comb) and to the TaxonGrab (T-Grab) approach (Koning et al., 2005), which is based on a set of regular expressions and lists. The results for TaxonGrab were obtained from the C.1932 part of our test set used for this evaluation (see Tab. 6). FAT is superior to both TaxonGrab and standalone word-level classification. The improved precision and recall results are due to the usage of greater variety of evidence. The better coverage results from the lower number of words that the word-level classifier has to deal with. On average, it has to classify only 0.1% of the words in a document. This also significantly reduces the user feedback and the number of potential errors of the word-level classifier.

	Precision	Recall	f-Measure	Coverage
T-Grab	96%	94%	95%	-
WLC	97%	95%	96%	95%
Comb	99.7%	99.9%	99.8%	99.8%

Table 6: Comparison to Related Approaches

All these positive effects result in about 99.8% f-Measure and 99.8% coverage. This means the error is reduced by 90% compared to word-level classification, and by 93% compared to TaxonGrab.

Misclassifications

Although FAT achieves very high performance, some errors remain. In this section, we take a closer look at the latter and discuss how we can prevent them in the future.

False Negatives. False negatives can occur in the language recognition step. Most of them contain two words the first of which is a genus. *Xenomyrmex varies … ,* for instance, could induce such an error: The word level classifier (correctly) recognizes the first word as a part of a taxonomic name. The second word is not typical enough to change the overall classification of the sequence. To avoid this type of false negatives, one might use POS-tagging, which would label *varies* as a verb. We could exclude word sequences containing words with a meaning that cannot occur in taxonomic names. A related problem results from literature references in arbitrary languages. Chapin (1932, 1939, 1953, 1954), for instance, cites *Systema naturae* (Linnaeus 1758), a book written in Latin. The word level classifier correctly recognizes the language as Latin. The problem is that our assumption that the taxonomic names are the only parts of the text in Latin does not always hold. Other publications cite complete paragraphs from documents written in Italian. *Sulla posizione sistematica* is an excerpt from such a paragraph that happens to match <taxName>. Because Italian is closer to Latin than to English, the word level classifier recognizes this phrase as a taxonomic name. German citations raise yet another problem: Because the capitalization rules of this language differ from the English ones, the regular expressions happen to match such text parts. In particular, all nouns are capitalized in German. This lets them match the genus and subgenus part of our regular expressions.

False Positives. The <taxName> regular expression matches any word sequence that is a taxonomic name. But the subsequent exclusion mechanisms may misclassify a sequence of words. In particular, the word-level classifier does not always recognize taxonomic names if they have been formed from proper names of persons. This is because these words consist of N-Grams that are typical for common English. *Wheeleria rogersi Smith*, for instance, is a fictitious but valid taxonomic name. To overcome this problem, we can construct these genera and species from the names we have extracted from the sure positives.

CONCLUSIONS

This paper has shown how combined computer linguistic techniques can be applied to automatically extract taxonomic names from English text documents. FAT yields a precision of up to 99.7% as opposed to TaxonGrab (96%).

A promising future avenue is to study those names which have not been detected, and to start to integrate other languages. This is in fact necessary, since a large part of the heritage literature is written in languages different from English.

ACKNOWLEDGMENTS

The authors thank the members of the project team (National Science Foundation IIS-0241229, Deutsche Forschungsgemeinschaft BIB47) for their comments, and colleagues from the Marine Biological Laboratory at Woods Hole, Massachusetts for their discussions regarding issues of name structure and extraction.

LITERATURE CITED

Bikel, D. M., S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: a high-performance learning name-finder. Proceedings of ANLP-97, Washington, USA.

Carreras, X. and L. Marquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling.

Chapin, J. 1932. The birds of the Belgian Congo: Part 1. Bulletin of the American Museum of Natural History: 65. American Museum of Natural History, New York.

Chapin, J. 1939. The birds of the Belgian Congo: Part 2. Bulletin of the American Museum of Natural History: 75. American Museum of Natural History, New York.

Chapin, J. 1953. The birds of the Belgian Congo: Part 3. Bulletin of the American Museum of Natural History: 75A. American Museum of Natural History, New York.

Chapin, J. 1954. The birds of the Belgian Congo: Part 4. Bulletin of the American Museum of Natural History: 75B. American Museum of Natural History, New York.

Chieu, H. L., and H. T. Ng. 2002. Named Entity Recognition: A Maximum Entropy Approach Using Global Information. Proceedings of COLING-02, Taipei, Taiwan.

Cucerzan, S., and D. Yarowsky. 1999. Language independent named entity recognition combining morphological and contextual evidence. In Proceedings of SIGDAT-99, College Park, USA..

Day, D., J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson and M. Vilain. 1997. Mixed-Initiative Development of Language Processing Systems. Proceedings of the 5th Conference on Applied Natural Language Processing.

Fisher, B. 2000. The Malagasy Fauna of Strumigenys. Pp. 612–710. *in* Bolton, B., The ant tribe Dacetini. Memoirs of the American Entomological Insititute, 65 (2): 1-1028.

Isozaki, H., and H. Kazawa. 2002. Efficient Support Vector Classifiers for Named Entity Recognition. Proceedings of COLING-02, Taipei, Taiwan

Jones, R., A. McCallum, K. Nigam, and E. Riloff. 1999. Bootstrapping for Text Learning Tasks. Proceedings of IJCAI-99 Workshop on Text Mining.

ICZN. 2000. International Code of Zoological Nomenclature, Fourth Edition. International Commission of Zoological Nomenclature, London.

JAVA 1.4.2: JAVA online documentation, http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html.

Koning, D., N. Sarkar, and T. Moritz. 2005. TaxonGrab: Extracting Taxonomic Names from Text. Biodiversity Informatics 2: 79-82.

Linnaeus, C. 1758. Systema Naturae. Ipsiae, Lund.

Maze, G. 2004. The role of taxonomy in species conservation. Philosophical Transaction of the Royal Society London B 359: 711–719.

Mikheev, A., M. Moens, and C. Grover. 1999. Named Entity Recognition without Gazetteers. Proceedings of EACL-99, Bergen, Norway.

Niu, Ch., W. Li, J. Ding, and R. K. Srihari. 2003. A Bootstrapping Approach to Named Entity Classification Using Successive Learners. Proceedings of 41st Annual Meeting of the Association for Computational Linguistics.

Palmer, D. D., and D. S. Day. 1997. A Statistical Profile of the Named Entity Task. Proceedings of ANLP-97, Washington, USA..

Porter, M.F. 1980. An algorithm for suffix stripping. Program, 14: 130-137

Sautter, G., and K. Böhm. 2006. How Helpful Is Word-Level Language Recognition to Extract Taxonomic Names? Technical Report, http://www.ipd.uni-karlsruhe.de/~sautter/TaxonomicNameExtraction.pdf.

Tanabe, L., and W. J. Wilbur. 2002. Tagging Gene and Protein Names in Biomedical Text, Bioinformatics 18:1124-1132.

Tjong, E. F., K. Sang, and F. De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, Edmonton, Canada, 2003.

WordNet. A lexical database for the English language, http://wordnet.princeton.edu/.

Yoshida, M., K.-I. Fukada, and T. Takagi. 1999. PDAD-CSS: a workbench for constructing a protein name abbreviation dictionary. Proceedings of the 32nd Hawaii International Conference on System Sciences.