The Difficulties of Taxonomic Name Extraction and a Solution

Guido Sautter **Klemens Böhm**

Dept. of Computer Science
Universität Karlsruhe (TH)
Germany

sautter@ipd.uka.de boehm@ipd.uka.de

Abstract

In modern biology, digitization of biosystematics publications is an important task. Extraction of taxonomic names from such documents is one of its major issues. This is because these names identify the various genera and species. This article reports on our experiences with learning techniques for this particular task. We say why established Named-Entity Recognition techniques are somewhat difficult to use in our context. One reason is that we have only very little training data available. Our experiments show that a combining approach that relies on regular expressions, heuristics, and word-level language recognition achieves very high precision and recall and allows to cope with those difficulties.

1 Introduction

Digitization of biosystematics publications currently is a major issue. They contain the names and descriptions of taxonomic genera and species. The names are important because they identify the various genera and species. They also position the species in the tree of life, which in turn is useful for a broad variety of biology tasks. Hence, recognition of taxonomic names is relevant. However, manual extraction of these names is time-consuming and expensive.

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) currently is a big research issue. However, conventional NER techniques are not readily applicable here for two reasons: First, the NE categories are rather high-level, e.g., names of organizations or persons (cf. common NER benchmarks such as (Carreras 2005)). Such a classification is too coarse for our context. The structure of taxonomic names varies widely and can be complex. Second, those recognizers require large bodies of training data. Since digitization of biosystematics documents has started only recently, such data is not yet available in biosystematics. On the other hand, it is important to demonstrate right away that text-learning technology is of help to biosystematics as well.

This paper reports on our experiences with learning techniques for the automated extraction of taxonomic names from documents. The various techniques are obviously useful in this context:

- Language recognition – taxonomic names are a combination of Latin or Latinized words, with surrounding text written in English,
- structure recognition – taxonomic names follow a certain structure,
- lexica support – certain words never are/may well be part of taxonomic names.

On the other hand, an individual technique in isolation is not sufficient for taxonomic name extraction. Mikheev (1999) has 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. Combining approaches are also promising for taxonomic name extraction. Having said this, the article will now proceed as follows:

First, we have conducted a thorough inspection of taxonomic names. An important observation is that one cannot model taxonomic names both concisely and precisely using regular expressions. As is done in bootstrapping, we use two kinds of regular expressions: *precision rules*, whose instances are taxonomic names with very high probability, and *recall rules*, whose instances are a superset of all taxonomic names. We propose a meaningful definition of precision rules and recall rules for taxonomic names.

Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL 06, pages 126–133,
New York City, June 2006.

Second, the essence of a combining approach is to arrange the individual specific approaches in the right order. We propose such a composition for taxonomic name extraction, and we say why it is superior to other compositions that may appear feasible as well at first sight.

Finally, to quantify the impact of the various alternatives described so far, we report on experimental results. The evaluation is based on a corpus of biosystematics documents marked up by hand. The best solution achieves about 99.2% in precision and recall. It prompts the user for only 0.2% of the words.

The remainder of the paper is as follows: Section 2 discusses related approaches. Section 3 introduces some preliminaries. Section 4 describes one specific combining approach in some detail. Section 5 features an evaluation. Section 6 concludes.

2 Related Work

This section reviews solutions to problems related to the extraction of taxonomic names.

2.1 Named Entity Recognition

Taxonomic names are a special case of named entity. In the recent past, NER has received much attention, which yielded a variety of methods. The most common ones are list lookups, grammars, rules, and statistical methods like SVMs (Bikel 1997). All these techniques have been developed for tasks like the one presented by Carreras (2005). Thus, their focus is the recognition of somewhat common NE like locations and persons. Consequently, they are not feasible for the complex and variable structure of taxonomic names (see Section 3.3). Another problem of common NER techniques is that they usually require several hundred thousand words of pre-annotated training data.

2.2 List-based Techniques

List-based NER techniques (Palmer 1997) make use of lists 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 list-based exclusion, using a common English dictionary. Koning (2005) combines such an approach with structural rules. In isolation, however, it is not an option either. First, it would not exclude proper names reliably. Second, it excludes parts of taxonomic names that are also used in common English. However, exclusion of sure negatives, i.e., words that are never part of taxonomic names, simplifies the classification.

2.3 Rule Based Techniques

Rule based techniques do not require pre-annotated training data. They extract words or word sequences based on their structure. Yoshida (1999) applies regular expressions to extract the names of proteins. He makes use of the syntax of protein names like *NG-monomethyl-L-arginine*, which is very distinctive.

There are also rules for the syntax of taxonomic names, but they are less restrictive. For instance, *Prenolepis (Nylanderia) vividula Erin subsp. guatemalensis Forel var. itinerans Forel* is a taxonomic name as well as *Dolichoderus decollatus*. Because of the wide range of optional parts, it is impossible to find a regular expression that matches all taxonomic names and at the same time provides satisfactory precision. Koning (2005) presents an approach based on regular expressions and static dictionaries. This technique performs satisfactorily compared to common NER approaches, but their conception of what is a positive is restricted. For instance, they leave aside taxonomic names that do not specify a genus. However, the idea of rule-based filters for the phrases of documents is helpful.

2.4 Bootstrapping

Instead of a large amount of labeled training data, Bootstrapping uses some labeled examples (“seeds”) and an even larger amount of unlabeled data for the training. Jones (1999) has shown that this approach performs equal to techniques requiring labeled training data. However, Bootstrapping is not readily applicable to our particular problem. Niu (2003) used an unlabeled corpus of 88.000.000 words for training a named entity recognizer. For our purpose, even unlabeled training data is not available in this order of magnitude, at least right now.

2.5 Active Learning

According to Day (1997), the original idea of Active Learning was to speed up the creation of large labeled training corpora from unlabeled documents. The system uses all of its knowledge during all phases of the learning. Thus, it labels most of the data items automatically and requires user interaction only in rare cases. In order to increase data quality, we include user-interaction in our taxonomic name extractor as well.

2.6 Gene and Protein Name Extraction

In the recent past, the major focus of biomedical NER has been the recognition of gene and protein names. Tanabe (2002) gives a good overview of various approaches to this task. Frequently used techniques are structural rules, dictionary lookups and Hidden Markov Models. Most of the approaches use the output of a part-of-speech tagger as additional evidence. Both gene and protein names differ from taxonomic names in that the nomenclature rules for them are by far stricter. For instance, they never include the names of the discoverer / author of a given part. In addition, there are parts which are easily distinguished from the surrounding text based on their structure, which is not true for taxonomic names. Consequently, the techniques for gene or protein name recognition are not feasible for the extraction of taxonomic names.

3 Preliminaries

This section introduces some preliminaries regarding word-level language recognition. We also describe a measure to quantify the user effort induced by interactions.

3.1 Measure for User Effort

In NLP, the f-Measure is popular to quantify the performance of a word classifier:

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

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

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

But components that use active learning have three possible outputs. If the decision between positive or negative is narrow, they may classify a word as uncertain and prompt the user. This prevents misclassifications, but induces intellectual effort. To quantify this effort as well, there are 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 obtain a single measure for overall classification quality, we multiply f-Measure and coverage and define **Quality Q** as

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

3.2 Word-Level Language Recognition for Taxonomic Name Extraction

In earlier work (Sautter 2006), we have presented 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. Both statistics are built from examples from the respective languages. It uses active learning to deal with the lack of training data. Precision and recall reach a level of 98%. This is satisfactory, compared to common NER components. At the same time, the user has to classify about 3% of the words manually. In a text of 10.000 words, this would be 300 manual classifications. We deem this relatively high.

3.3 Formal Structure of Taxonomic Names

The structure of taxonomic names is defined by the rules of Linnaean nomenclature (Ereshefsky 1997). They are not very restrictive and include many optional parts. For instance, both *Prenolepis (Nylanderia) vividula Erin 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. Table 1 shows the decomposition of the two examples. The parts with their names in brackets are optional. More formally, the rules of Linnaean nomenclature define the structure of taxonomic names as follows:

- The **genus** is mandatory. It is a capitalized word, often abbreviated by its first one or two letters, followed by a dot.

- The **subgenus** is optional. It is a capitalized word, often enclosed in brackets.
- The **species** is mandatory. It is a lower case word. It is often followed by the name of the scientist who first described the species.
- The **subspecies** is optional. It is a lower case word, often preceded by ***subsp.*** or ***subspecies*** as an indicator. It is often followed by the name of the scientist who first described it.
- The **variety** is optional. It is a lower case word, preceded by ***var.*** or ***variety*** as an indicator. It is often followed by the name of the scientist who first described it.

Part		
Genus	*Prenolepis*	*Dolichoderus*
(Subgenus)	*(Nylanderia)*	
Species	*vividula*	*decollatus*
(Discoverer)	*Erin*	
(Subspecies)	*subsp. guatemalensis*	
(Discoverer)	*Forel*	
(Variety)	*var. itinerans*	
(Discoverer)	*Forel*	

Table 1: The parts of taxonomic names

4 Combining Techniques for Taxonomic Name Extraction

Due to its capability of learning at runtime, the word-level language recognizer needs little training data, but it still does. In addition, the manual effort induced by uncertain classifications is high. Making use of the typical structure of taxonomic names, we can improve both aspects. First, we can use syntax-based rules to harvest training data directly from the documents. Second, we can use these rules to reduce 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.

4.1 The Classification Process

1. We apply the precision rules. Every word sequence from the document that matches such a rule is a *sure positive*.
2. 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. We make use of domain-specific vocabulary and filter out word sequences containing at least one known negative word.
4. We collect a set of names from the set of sure positives (see Subsection 4.5). 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 apply it to the remaining uncertain word sequences.

Figure 1 visualizes the classification process. At first sight, other orders seem to be possible as well, e.g., the language recognizer classifies each word first, and then we apply the rules. But this is not feasible: It would require external training data. In addition, the language recognizer would have to classify all the words of the document. This would incur more manual classifications.

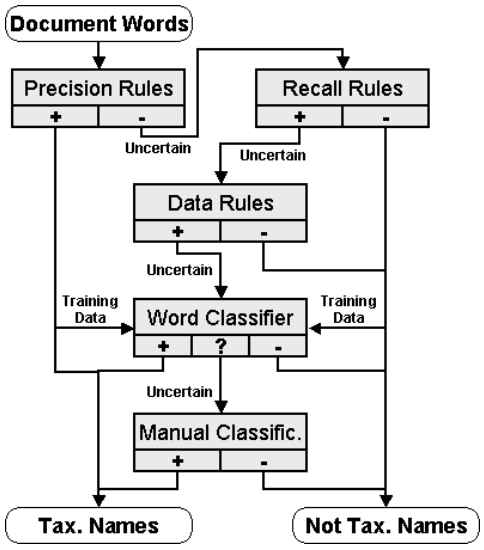

Figure 1: The Classification Process

This approach is similar to the bootstrapping algorithm proposed by Jones (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. Average biosystematics documents contain about 15.000 words, which is less than 0.02% of the data used by Niu (2003). On the other hand, with the classification process proposed here, the accuracy of the underlying classifier has to be very high from the start.

4.2 Structural Rules

In order to make use of the structure of taxonomic names, we use rules that refer to this structure. 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). Table 2 provides some abbreviations, to increase readability. We model taxonomic names as follows:

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

Table 2: Abbreviations

- The genus is a capitalized word, often abbreviated. We denote it as <genus>, which stands for {<CapW>|<CapA>}.
- The subgenus is a capitalized word, optionally surrounded by brackets. We denote it as <subGenus>, which stands for <CapW>|(<CapW>).
- The species is a lower case word, optionally followed by a name. We denote it as <species>, which stands for <LcW>{_<Name>}?.
- The subspecies is a lower case word, preceded by the indicator ***subsp.*** or ***subspecies***, and optionally followed by a name. We denote it as <subSpecies>, standing for {subsp.|subspecies}_<LcW>{_<Name>}?.
- The variety is a lower case word, preceded by the indicator ***var.*** or ***variety***, and optionally followed by a name. We denote 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>}?
```

4.3 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 in lower case. Any two words at the beginning of a sentence are a match!

To obtain more precise regular expressions, 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>. Even though these regular expressions may produce false negatives, 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>.

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*. We assume that surely positive words exclusively appear as parts of taxonomic names.

4.4 Recall Rules

<taxName> matches any sequence of words that conforms to the Linnaean rules, 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 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.

Our technique also extracts the names of the scientists from the sure positives and collects them in a name lexicon. Based on the structure described in Section 3.3, a capitalized word in a sure positive is a name if it comes after the second position. From the sure positive *Pseudomyrma (Minimyrma) arboris-sanctae Emery*, the technique extracts *Pseudomyrma*, *Minimyrma* and *arboris-sanctae*. In addition, it would add *Emery* to the name lexicon.

We cannot be sure that the list of sure positive words suffices to find all species names in an enumeration. Hence, our technique additionally collects all lower-case words followed by a word contained in the name lexicon. In the example, we extract *latinoda Mayr* and *tachigalide Forel* if *Mayr* and *Forel* are in the name lexicon.

4.5 Data Rules

Because we want to achieve close to 100% in recall, the recall rules are very weak. In consequence, many word sequences that are not taxonomic names are considered uncertain. Before the word-level language recognizer deals with them, we see some more ways to exclude negatives.

Sure Negatives. As mentioned in Subsection 4.3, `<taxName>` matches any capitalized word followed by a word in lower case. This includes the start of any sentence. Making use of the sure negatives, we can recognize these phrases. In particular, out technique classifies 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 also cause false matches. The reason is that they are capitalized. A misclassification occurs if 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 match of `<taxName>` if the first or second word is contained in the name lexicon. For instance, in "*…, and Forel further concludes*", *Forel further* matches `<taxName>`. If the name lexicon contains *Forel*, we know that it is not a genus, and thus exclude *Forel further*.

4.6 Classification of Remaining Words

After applying the rules, some word sequences still remain uncertain. To deal with them, we use word-level language recognition. We train the classifier with the sure positive and sure negative words. We do not classify every word separately, but compute the 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 feedback request. Second, if a word sequence 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, it is easier to determine the meaning of a word sequence than the one of a single word.

5 Evaluation

A combining approach gives rise to many questions, 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 to dynamic lexicons? Which kinds of errors remain?

We run two series of experiments: We first process individual documents. We then process the documents incrementally, i.e., we do neither clear the sets of known positives and negatives after each document, nor 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 from Section 4.

The platform is implemented in **`JAVA 1.4.2`**. We use the **`java.util.regex`** package to represent the rules. All tests are based on 20 issues of the *American Museum Novitates*, a natural science periodical published by the *American Museum of Natural History*. The documents contain about 260.000 words, including about 2.500 taxonomic names. The latter consist of about 8.400 words.

5.1 Tests with Individual Documents

First, we test the combined classifier with individual documents. The **Docs** column in Table 3 contains the results. The combination of rules and word-level classification provides very high precision and recall. The former is 99.7% on average, the latter 98.2%. The manual effort is very low: The average coverage is 99.7%.

5.2 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. The **Corp** column of Table 3 shows the results. As expected, the classifier performs better than with individual documents. The average recall is 99.2%, coverage is 99.8% on average. Only precision is a little less, 99.1% on average.

	Docs	Corp
<preciseTaxName>	22,6	
<taxName>	414,1	
SN excluded	78,5	
Names excluded	176,15	
Scorings	139,9	
User Feedbacks	19,6	10,35
False positives	4,25	1,5
False negatives	0,55	1,5
Precision	0,997	0,991
Recall	0,982	0,992
f-Measure	0,990	0,992
Coverage	0,997	0,998
Quality	0,987	0,990

Table 3: Test results

The effect of the incremental learning is obvious. The false positives are less than half of those in the first test. A comparison of Line `False Positives` in Table 3 shows this. The same is true for the number feedback requests (Line `User Feedbacks`). The slight decrease in precision (Line `False Negatives`) results from the propagation of misclassifications between documents. The reason for the improvement becomes clear for documents where the number of word sequences in `<preciseTaxName>` is low: experience from previous documents compensates the lack of positive examples. This reduces both false positives and manual classifications.

5.3 The Data Rules

The exclusion of word sequences containing a sure negative turns out to be effective to filter the matches of `<taxName>`. Lines `<taxName>` and `SN excluded` of Tables 3 show this. On average, this step excludes about 20% of the word sequences matching `<taxName>`. Lines `<taxName>` and `Names excluded` tell us that the rule based on the names of scientists is even more effective. On average, it excludes about 40% of the matches of `<taxName>`. Both data rules decrease the number of words the language recognizer has to deal with and eventually the manual effort. This is because they reduce the number of words classified uncertain.

5.4 Comparison to Word-Level Classifier and TaxonGrab

A word-level classifier (WLC) is the core component of the combining technique. We compare it in standalone use to the combining technique (Comb) and to the TaxonGrab (T-Grab) approach (Koning 2005). See Table 4. The combining technique is superior to both TaxonGrab and standalone word-level classification. The reason for better precision and recall is that it uses more different 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 2.5% of the words in a document. This reduces the classification effort, leading to less manual feedback. It also decreases the number of potential errors of the word-level classifier.

All these positive effects result in about 99% f-Measure and 99.7% coverage. This means the error is reduced by 75% compared to word-level classification, and by 80% compared to TaxonGrab. The manual effort decreases by 94% compared to the standalone word-level classifier.

	Precision	Recall	f-Measure	Coverage
T-Grab	96%	94%	95%	-
WLC	97%	95%	96%	95%
Comb	99.1%	99.2%	99%	99.7%

Table 4: Comparison to Related Approaches

5.5 Misclassified Words

Despite all improvements, there still are word sequences that are misclassified.

False Negatives. The regular expressions in `<preciseTaxName>` are intended to be 100% precise. There are, however, some (rare) exceptions. Consider the following phrase: *"… In Guadeloup (Mexico) another subspecies killed F. Smith."* Except for the word *In*, this sentence matches the

regular expression from `<preciseTaxName>` where `<subSpecies>` is mandatory. Similar pathologic cases could occur for the variety part. Another class of false negatives contains two word sequences, and the first one is the name of a genus. For instance, "*Xenomyrmex varies* …" falls into this category. The classifier (correctly) recognizes the first word as a part of a taxonomic name. The second one is not typical enough to change the overall classification of the sequence. To recognize these false negatives, one might use POS-tagging. We could exclude word sequences containing words whose meaning does not fit into a taxonomic name.

False Positives. Though `<taxName>` matches any taxonomic name, the subsequent exclusion mechanisms may misclassify a sequence of words. In particular, the word-level classifier has problems recognizing taxonomic names containing proper names of persons. The problem is that these words consist of N-Grams that are typical for common English. *"Wheeleria rogersi Smith"*, for instance, is a fictitious but valid taxonomic name. A solution to this problem might be to use the scientist names for constructing and recognizing the genus and species names derived from them.

6 Conclusions

This paper has reported on our experiences with the automatic extraction of taxonomic names from English text documents. This task is essential for modern biology. A peculiarity of taxonomic name extraction is a shortage of training data. This is one reason why deployment of established NER techniques has turned out to be infeasible, at least without adaptations. A taxonomic-name extractor must circumvent that shortage. Our experience has been that designing regular expressions that generate training data directly from the documents is feasible in the context of taxonomic name extraction. A combining approach where individual techniques are carefully tuned and assigned in the right order has turned out to be superior to other potential solutions with regard to precision, recall, and number of user interactions. – Finally, is seems promising to use document and term frequencies as additional evidence. The ides is that both are low for taxonomic names.

7 References

(Bikel 1997) Daniel M. Bikel, Scott Miller, Richard Schwartz, Ralph Weischedel: *Nymble: a high-performance learning name-finder*, In Proceedings of ANLP-97, Washington, USA, 1997

(Carreras 2005) Xavier Carreras, Lluis Marquez: *Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling*, 2005

(Chieu 2002) Hai Leong Chieu, Hwee Tou Ng: *Named Entity Recognition: A Maximum Entropy Approach Using Global Information*, In Proceedings of COLING-02, Taipei, Taiwan, 2002

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

(Day) David Day, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson, Marc Vilain: *Mixed-Initiative Development of Language Processing Systems*, In Proceedings of ANLP-97, Washington, USA, 1997

(Ereshefsky 1997) Marc Ereshefsky: *The Evolution of the Linnaean Hierarchy*, Springer Science & Business Media B.V., 1997

(Jones 1999) Rosie Jones, Andrew McCallum, Kamal Nigam, Ellen Riloff: *Bootstrapping for Text Learning Tasks*, In Proceedings of IJCAI-99 Workshop on Text Mining, 1999

(Koning 2005) Drew Koning, Neil Sarkar, Thomas Moritz: *TaxonGrab: Extractin Taxonomic Names from Text*

(Niu 2003) Cheng Niu, Wei Li, Jihong Ding, Rohini K. Srihari: *A Bootstrapping Approach to Named Entity Classification Using Successive Learners*, In Proceedings of 41st Annual Meeting of the ACL, 2003

(Palmer 1997) David D. Palmer, David S. Day: *A Statistical Profile of the Named Entity Task*, In Proceedings of ANLP-97, Washington, USA, 1997.

(Sautter 2006) G. Sautter, K. Böhm, K. Csorba: *How Helpful Is Word-Level Language Recognition to Extract Taxonomic Names?*, submitted to DILS, 2006

(Tanabe 2002) Lorraine Tanabe, W. John Wilbur: *Tagging Gene and Protein Names in Biomedical Text*, Bioinformatics, Vol. 18, 2002, pp. 1124-1132

(Yoshida 1999) Mikio Yoshida, Ken-ichiro Fukada and Toshihisa Takagi: *PDAD-CSS: a workbench for constructing a protein name abbreviation dictionary*, In Proceedings of the 32nd HICSS, 1999