An Information Extraction Tool for Microbial Characters

Jin Mao[1], Lisa Moore[2], Carrine Blank[3], Hong Cui[1]
[1]School of Information, University of Arizona
[2]Department of Biological Sciences, University of Southern Maine
[3]Department of Geosciences, University of Montana

Abstract
Automated extraction of phenotypic and metabolic characters from microbial taxonomic descriptions will benefit biology research and study. In this poster, we describe a Microbial Phenomics Information Extractor (MicroPIE) system, which takes taxonomic descriptions in XML files as input and extracts 57 types of microbial characters. The main extraction steps are :1) splitting paragraphs into sentences; 2) predicting the characters described in the sentences by using automated classifiers; 3) extracting character values from the sentences by applying a variety of methods, such as Regular Expression Rule, Term Matching, and Unsupervised Semantic Parsing. Parts of the system have been implemented and optimized for better performance. Results on optimizing the sentence classifiers show that SVM (Support Vector Machine) achieved better performance over the Naive Bayes classifier when the problem of unbalanced training instances was resolved by controlling the ratio of positive sentences over negative sentences.
Keywords: Phenotypic character extraction; information extraction; natural language processing; text classification
doi: 10.9776/16562

Acknowledgements: The work reported in this poster is part of the “Next Generation Phenomics” project funded by the National Science Foundation (NSF DEB-1208567).
Contact: maojin0@gmail.com

1 Introduction

Describing species has long been a basic and essential component to study biodiversity and evolution. Scientists have published detailed descriptions of organisms for centuries, amassing a rich legacy of taxonomic literature that includes descriptions of phenotypic characters (i.e., the traits of an organism, such as shape, size, and growth conditions) of millions of species. These character descriptions have become important materials for biology education and research. To better utilize the accumulating knowledge, the character information needs to be extracted and put in a structured format, such as taxon-by-character matrices, which are in tabular format with rows for taxa and columns for characters). However, it’s tedious and time-consuming to collect and annotate phenotypic character values manually. In the NSF-funded Next Generation Phenomics Project, we were charged to develop a software application (including the related information extraction algorithms) to extract character information from published taxonomic descriptions of micro organisms. In this poster, we describe our progress on the development of this software application called Microbial Phenomics Information Extractor (MicroPIE).

2 MicroPIE System Design

2.1 The format of taxonomic descriptions

The input of the system are microbial taxonomic descriptions, which can be obtained from published articles or books. To standardize the input format, the taxonomic descriptions are transformed into an XML format with elements such as taxon name, description, and source information.

2.2 Characters to be extracted

The characters to be extracted were defined by microbiologists based on their domain knowledge and by topic analysis on taxonomic descriptions using Latent Semantic Analysis (Wild, 2014). In the end, 9 categories holding 57 characters were identified as the extraction target.

2.3 Extraction process

The system takes the XML description text as input and generates a taxon-by-character matrix as output. The main steps of the extraction process are shown in Figure 1.

(1) Sentence splitting. Taxonomic description paragraphs are parsed from XML input files and then are split into sentences using Stanford Parser (Manning et al., 2014). Some sentences contain

multiple clauses describing different characters. To further split these complex sentences, a clause splitter, ClausIE (Del Corro & Gemulla, 2013), is applied to improve extraction performance.

(2) Sentence classification. For each character, a supervised text classifier, SVM (Cortes & Vapnik, 1995), was trained using sentences/clauses from microbiology taxonomic descriptions provided by microbiological experts and labelled with character names. The trained classifiers are stored in the system and used to predict the classifications of a new sentence. After the characters within a sentence are determined by the sentence classifier, the sentence will be passed into appropriate character extractors to extract the values.

(3) Character extraction. Each of the 57 characters is associated with a character extractor. Three techniques are designed to extract values. First, regular expressions are used to extract characters with clear lexical/syntax clues; for example, the pH value suitable for bacteria to grow can be extracted from the sentence "Grows at pH 8". Second, term matching utilizes extensive term lists created by a microbiologist to extract desired values. The last method used is Unsupervised Semantic Parsing (Poon & Domingos, 2009), which groups together varied syntactic expressions sharing a similar meaning. It can be used to extract character values associated with a set of seed expressions. It should be noted that different characters can share the same extractor.

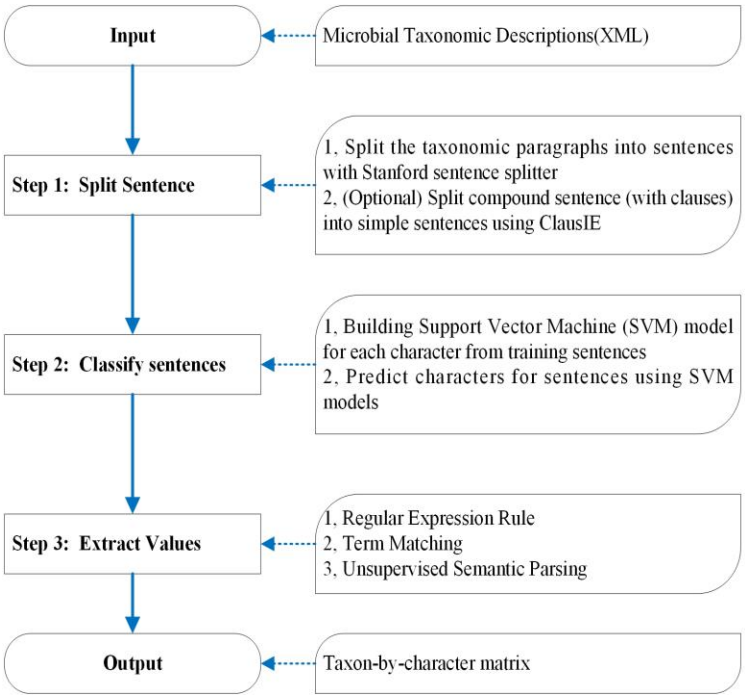

Figure 1. System work flow of MicroPIE

2.4 Taxon-by-character matrix

Taxon-by-character matrix has taxa as rows, characters as columns, and character values within each cell of the matrix. One such matrix is shown in Figure 2. Number signs (#) are used to separate multiple values for a character.

Taxon	%G+C	Cell shape	Pigment compounds	Cell wall	pH min	pH opt	pH max
Aquimarina intermedia	37.1		flexirubin-type pigments				
Bacteroides gallinarum	47	rods # singly or pairs		gram-negative			
Bacteroides stercoris	46	pairs # singly					
Bacteroides xylanisolvens	42.8	rods		gram-negative	6	6.8	7.2
Balneola		rod		gram-negative		8	
Bizionia algoritergicola	45	rod-like	not flexirubin				
Bizionia echini	34.4						
Bizionia gelidisalsuginis	39	rod-like	not flexirubin pigments				
Butyricimonas		rod-shaped		gram-negative			

Figure 2. Part of a simplified taxon-by-character matrix of microbial taxa

3 Optimizing Sentence Classifiers

We have so far implemented the main process of the system and are in the process of evaluating and optimizing each step of the system described in Figure 1. The split sentences look appropriate to support the subsequent steps well. Sentence classifiers have already been trained to predict the characters contained in the sentences. The majority of the character extractors have been implemented and more are being developed. We report how we train and optimize sentence classifiers here.

3.1 Training set

From 9642 randomly sampled sentences from 625 comprehensively gathered taxonomic descriptions, 20,126 training instances (one instance = one sentence + one label) were generated as one sentence may be labeled with zero or multiple characters. The number of training sentences ranges from less than ten for some characters to more than one thousand for other characters, because certain characters of microorganisms occur much less frequently than others. This presents an unbalanced training example issue (He & Garcia, 2009) for the classifiers.

The unbalanced training examples create a greater problem for binary classifiers such as SVM, because rare characters will have few, positive sentence examples (often contains 10-1000 examples) relative to a very large, negative sentence set (often contains close to 20,000 examples). This unbalanced problem can be resolved by controlling the ratio of positive sentences over negative sentences as *1:r* (where *r* is a tuning parameter taking values from 1.0 to 2.0 with steps of 0.1) by randomly sampling the negative sentences from the negative set in the training stage.

3.2 Evaluation

LibSVM classifiers (Chang & Lin, 2011) were compared against the baseline model, Weka NaiveBayes classifier (shorted as NB in this poster) (Hall et al., 2009). A 10-fold cross validation process was applied to evaluate the performance of classifiers in terms of accuracy and recall. The results are averaged over 52 characters that have more than 10 training sentences.

$$\text{Accuracy} = \frac{|correctly\ classified\ \text{sentences}\ |}{|all\ \text{sentences}\ |}$$

$$\text{Recall} = \frac{|correctly\ classified\ \text{positive sentences}|}{|all\ \text{positive sentences}\ |}$$

Accuracy is used to measure the overall accuracy of the classifier. However,a high importance is placed on recall because high recall of positive sentences will benefit the subsequent character extraction step whereas an incorrect classification can have multiple negative consequences. For instance, if one sentence that describes character C1 is predicted as describing another character C2, the sentence will be passed into a wrong character extractor, resulting in a missing value in the final matrix for the character C1, and possibly a wrong value for the other character C2. Therefore, we strive to train the classifiers to reach both high accuracy and high recall.

The classifiers that do not resolve the unbalanced training instances problem (marked as SVM+UB and NB+UB) are compared with the classifiers with positive/negative ratio controlled (marked as SVM and NB). The parameter r, taking values from 1.0 to 2.0 with a step of 0.1, was tested to find the optimal results for the SVM classifiers.

3.3 Results

The optimal results for the SVM classifiers is with r set to 1.0. Figure 3 illustrates the performances of the classifiers when r = 1.0. It's shown from the figure that the two groups of SVM classifiers had better performance than the two groups of NaïveBayes classifiers in terms of both accuracy and recall except for SVM+UB in terms of recall. The values of precision are nearly the same between SVM+UB and SVM, and between NB+UB and NB, showing controlled positive/negative ratio did not affect the overall accuracy of the classifiers. However, SVM+UB's performance improved dramatically when controlling the ratio of positive sentences over negative sentences as *1:r*, with r = 1.0, indicating the necessity of resolving the unbalanced training instance problem for the SVM classifier. The recall of SVM has reached up to 99.83%, ensuring the effectiveness of sentence classification to support the subsequent character extraction step.

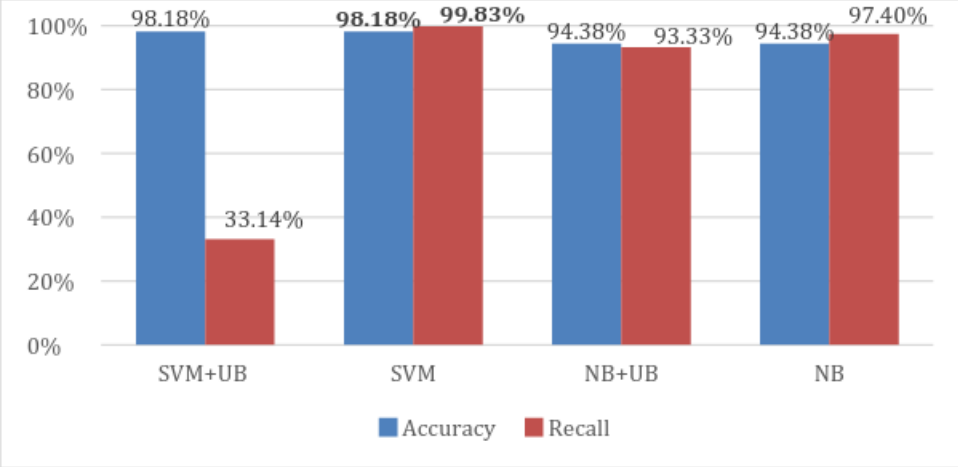

Figure 3. The performance of different classifiers

4 Conclusion

A comparison of SVM classifiers and NaiveBayes classifiers on the microbial description sentences suggests that NaïveBayes is the winner (33.14% vs. 93.33%). However, after applying the simple method of balancing training examples with 1/r, SVM outperforms NaiveBayes. Thus, we will use SVM with balanced training sets in the MicroPIE system and continue to evaluate and optimize the remaining components.

5 Acknowledgement

The work reported in this poster is part of the "Next Generation Phenomics" project funded by the National Science Foundation (NSF DEB-1208567).

6 References

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27:1-27:27. doi: 10.1145/1961189.1961199

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273-297. doi:10.1007/BF00994018

Del Corro, L., & Gemulla, R. (2013). ClausIE: Clause-based open information extraction. *Proceedings of the 22nd International Conference on World Wide Web*, 355-366.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10-18. doi: 10.1145/1656274.1656278

He, H., & Garcia, E. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263-1284. doi: 10.1109/TKDE.2008.239

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations:* 55-60.

Poon, H., & Domingos, P. (2009). Unsupervised semantic parsing. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing:* 1-10.

Wild, F. (2014). LSA: Latent semantic analysis. Retrieved from https://cran.r-project.org/web/packages/lsa/index.html.

7 Table of Figures