Enhancing a biomedical information extraction system with dictionary mining and context disambiguation

S. Mukherjea
L. V. Subramaniam
G. Chanda
S. Sankararaman
R. Kothari
V. Batra
D. Bhardwaj
B. Srivastava

Journals and conference proceedings represent the dominant mechanisms for reporting new biomedical results. The unstructured nature of such publications makes it difficult to utilize data mining or automated knowledge discovery techniques. Annotation (or markup) of these unstructured documents represents the first step in making these documents machine-analyzable. Often, however, the use of similar (or the same) labels for different entities and the use of different labels for the same entity makes entity extraction difficult in biomedical literature. In this paper we present a system called BioAnnotator for identifying and classifying biological terms in documents. BioAnnotator uses domain-based dictionary lookup for recognizing known terms and a rule engine for discovering new terms. We explain how the system uses a biomedical dictionary to learn extraction patterns for the rule engine and how it disambiguates biological terms that belong to multiple semantic classes.

1. Introduction

Biomedical information is growing explosively, and new and useful results are appearing daily in research publications. Many of these publications are available online—for example, in the PubMed MedLine database [1]. However, automatic extraction of useful information from these online sources remains a challenge because these documents are unstructured and expressed in a natural language form. To enable data mining and knowledge discovery from such documents, this data must be made available in a structured format. Because of the very large amounts of data being generated, it is difficult to have human curators extract all of the information and present it in a form usable by data mining and knowledge discovery tools.

Information extraction in the biomedical domain is a challenging task. A major problem is that because of inconsistent naming conventions, a term may be used to denote more than one semantic class. For example, *p53* is used to specify both a gene and a protein. Another problem is that new biological terms are continuously being created. Therefore, although several biomedical dictionaries and ontologies have been developed, none of them are up to date with the latest advances in the domain.

We have developed a system called *BioAnnotator* [2] for identifying biological terms in the scientific literature and annotating the terms with their semantic classes. BioAnnotator first identifies terms that are already known by doing a lookup on various publicly available biomedical dictionaries such as Unified Medical Language System (UMLS) [3] and Locus Link [4]. It then attempts to identify new and unknown terms by using character- and word-level properties of biological terms in addition to contextual clues.

In this paper, we discuss how BioAnnotator handles some of the challenges of biomedical information extraction. The next section cites related work, and Section 3 gives an overview of BioAnnotator. Section 4 describes how we use UMLS to discover some extraction patterns for the rule engine. Section 5 explains our technique of determining the semantic class of ambiguous

0018-8646/04/$5.00 © 2004 IBM

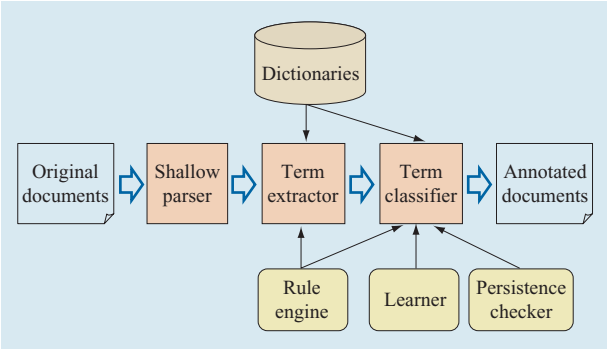

Figure 1

BioAnnotator: Component and flow details.

biological terms on the basis of the surrounding context. Finally, Section 6 is the conclusion.

2. Related work

The task of extracting biological terms from scientific documents can be considered similar to the *named entity* task in the Message Understanding Conference (MUC) evaluation exercises [5]. Many biomedical information extraction methods thus represent adaptations of methods originally proposed for MUC [6].

Biological term extraction systems can be broadly divided into two types: those with a rule base and those with a learning method. In [7], protein names are identified in biological papers using hand-coded rules. A rule-based approach combined with dictionary lookup for term recognition and classification is given in [8]. In [9], supervised learning methods based on hidden Markov models are used. In [10], statistical approaches based on word distributions in a large corpus are used to find biological terms. In [11], an entropy-based approach combined with morphological rules is used for finding terms. An excellent overview of the field is given in [12].

BioAnnotator uses a rule engine as well as biomedical dictionaries for identifying biological terms. Some of the previous rule-based systems have tuned their rules for identifying a small class of terms. For example, [7] has created rules for finding only proteins. On the other hand, BioAnnotator attempts to identify all possible biological terms. Instead of simply using hand-coded rules as in previous systems, we have also used UMLS to "learn" some patterns for extracting biological terms. Moreover, the system is designed so that the rules can easily be modified to identify a different class of entities. In contrast to most of the previous systems, BioAnnotator has also been evaluated using a publicly available corpus. As stated in [12], good evaluation of the existing systems is one of the main challenges in this domain.

Biological term classification involves assigning the correct class to a term. In the case of polysemous words, this becomes the word sense disambiguation problem. A technique for term classification that uses statistical classifiers and decision trees is given in [13]. An unsupervised method that exploits the constraints between different occurrences of a word in a document is presented in [14]. In [15], a machine learning approach is used, and [16] provides a good introduction to word sense disambiguation techniques and evaluates them. Unlike previous systems, BioAnnotator utilizes a hybrid approach, by combining rule-based and learning-based techniques, for term classification and context disambiguation.

3. BioAnnotator overview

BioAnnotator takes raw documents and annotates the biological terms present in them. This section gives a brief overview of the system.

Term extraction

Figure 1 shows the overall flow of the BioAnnotator document-annotation process. The system first uses a shallow parser to identify noun phrases[1] in the input document. The identified phrases are examined by a term extractor, which uses dictionaries and a rule engine to discover biological terms. The term classifier is used to determine the semantic classes of the biological terms. The details of the classifier are presented in Section 5.

Dictionaries

At present we are using three dictionaries:

- *Unified Medical Language System (UMLS)* [3]: UMLS is a consolidated repository of medical terms and their relationships. Each biological concept in UMLS is associated with semantic classes such as *gene or genome* and *amino acid, peptide, or protein.*
- *LocusLink* [4]: This is our primary knowledge source for gene names. It contains the list of genes of several organisms such as human, fruit fly, rat, and cow.
- *GeneAlias:* This is a locally compiled list of aliases for some of the gene names that are not present in LocusLink.

More dictionaries can be added by specifying them in a configuration file. The term extractor first looks up the entire noun phrase in the dictionaries. It then removes stop words (words such as *and*, *the*, *is*, and *was*) from the beginning and end of the phrase using a stop-word list.

[1] Although the parser can identify other types of phrases such as noun prepositional phrases and verb groups, our evaluation shows that using only noun phrases gives the best results.

This list contains standard English stop words as well as some additional words that occur frequently in the documents of interest, such as *abstracts*, *title*, and *pubmed*. The stripped phrase is then sent for lookup. If no match is found for the phrase, each single word from the phrase is looked up. Note that the order in which the dictionaries are accessed is specified in the configuration file. It can also be specified that a dictionary should be accessed only after the rule engine fails.

Rule engine

We have seen that many biological concepts are missing from UMLS and the other knowledge sources that we use. For identifying these terms we use a rule engine. Though the naming of biological concepts does not follow any convention, many biological terms have some specific patterns. The rules of the engine are encoded in a rule base, which is an XML file containing regular expressions specifying positive and negative examples of biological term patterns. Let us illustrate these with some examples[2]:

- Many biological terms contain uppercase letters, numerical figures, and non-alphabetical characters (for example, *PTEN*, *c-N-ras*, *CD4-Positive T-Lymphocytes*). To identify these terms, we use the following two regular expressions:

```
⟨RegularExpression name=
                    "DigitCapSpecialChar"⟩
   [\p{Upper}\d\p{Punct}]
⟨/RegularExpression⟩

⟨RegularExpression name=
                "NotDigitCapSpecialChar"⟩
  (^[\d\p{Punct}]+$)|(^\p{Upper}?
                        [\p{Lower}−]+$)
  |(^[\p{Alpha}\.−]+\.[\p{Alpha}\.−]+$)
⟨/RegularExpression⟩
```

The first regular expression matches any word that has uppercase alphabets, numbers, or special characters. The second regular expression can be used to filter out some non-biological words such as *10,000*, *H.D.Smith*, and proper nouns.

- In the documents many biological concept names are preceded or followed by keywords or signals that give an indication of their class (for example, *p16 tumor suppressor gene, pancreatic alpha cells, proteins Rac1 and Cdc42*). We have formed regular expressions for such signal words:

```
⟨RegularExpression name="CellSignal"⟩
   (?i)apoptoti(s|c)|cell(s)?|clone(d|s)?|
       culture(d|s)?|neuron(s)?|strain(s)?
⟨/RegularExpression⟩

⟨RegularExpression name="GeneSignal"⟩
   (?i)mutate(d|s)?|(onco)?gene(s)?
⟨/RegularExpression⟩

⟨RegularExpression name="ProteinSignal"⟩
   (?i)amino|amine(s)?|enzyme(s)?
                              |kinase(d|s)?|
       ligand(s)?|motif(s)?|peptid(e|ase)|
       protein(s|ase)?|protease
⟨/RegularExpression⟩
```

- Biological concept names often contain prefixes and suffixes that give an indication of their class [17]. For example, many proteins end with *ase* (for example, *amylase*). Therefore, the following regular expression is useful:

```
⟨RegularExpression name="ProteinSuffix"⟩
   .+ase$
⟨/RegularExpression⟩
```

As discussed in the next section, we try to learn these prefixes and suffixes from UMLS. The input to the rule engine is a noun phrase with all of the leading and trailing stop words removed. The input phrase is matched with the regular expressions. If there is no match, the individual words of the phrase are matched with the regular expressions. Each match is given a score based on the importance of the regular expression (also specified in the rule base). If the overall score is greater than a threshold specified in the configuration file, it is considered to be a biological term. For example, in the phrase *protein-kinase proto oncogene, protein-kinase* is a protein signal and *oncogene* is a gene signal. If the combined score of the two matches is greater than the threshold, the term is considered to be a biological term. (Note that BioAnnotator outputs one annotation per term, even if the subterms are themselves identified as biologically relevant.)

Evaluation

We performed a formal evaluation of BioAnnotator term extraction using the publicly available GENIA 1.1 corpus [18]. This corpus contains abstracts of 670 research papers as well as a list of the biological terms manually identified in them by human experts.

The BioAnnotator results are compared with the manual annotations. When a term from BioAnnotator is matched with a human-annotated term, one can look for an exact or approximate match. For an exact match, the

[2] Our regular expressions follow the *java.util.regex* convention.

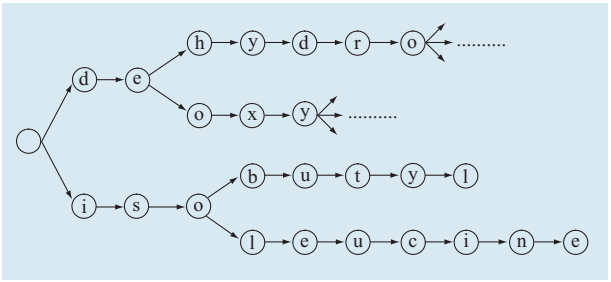

Figure 2

Example trie for the prefixes *dehydro* and *deoxy* as well as the words *isobutyl* and *isoleucine*.

annotations from BioAnnotator and experts should match exactly. For an approximate match, one of the annotations should be a substring of the other. **Table 1** summarizes the results, in which, for a system which finds m correct terms and n incorrect terms, the precision is $m/(m + n)$, and in a document containing p biological terms, the recall would be m/p. Note that we have also shown the *F-score,* which is the harmonic mean of precision and recall. It is calculated as $(2 * Precision * Recall)/(Precision + Recall)$. Further details about the evaluation are presented in [2].

4. Learning affixes from UMLS

Biological concept names often contain prefixes and suffixes that give an indication of their class [17]. For example, many proteins end with *ase*. Instead of simply trying to determine the prefixes and suffixes manually as in previous systems [7], we also try to "learn" these patterns by mining UMLS. The biological concepts in UMLS are grouped into semantic types such as *Amino Acid, Peptide or Protein,* and *Carbohydrates*. Our objective is to identify the common prefixes and suffixes for several semantic types of interest. The procedure is as follows:

- *Preprocessing UMLS biological terms:* We first extracted all of the strings in the UMLS database which belonged to a particular type. We then removed all punctuation marks, numbers, etc. from the strings, leaving only the normalized English. After processing, for example, the string *(1′R,2′S,3′R,4′S)-1-(2′,3′,4′-trihydroxycyclopent-1′-yl)-1H-uracil* was reduced to *rsrs trihydroxycyclopent yl H uracil*. We divided this processed string further, into individual words. We then removed all duplicate words, since we do not wish to identify prefixes and suffixes just because they are part of very common biological terms. After this preprocessing step, we were left with a unique list of words.

Table 1 Precision and recall of the BioAnnotator.

Match type	*Precision*	*Recall*	*F-score*
Approximate	0.8652	0.9425	0.9022
Exact	0.6029	0.6859	0.6401

Table 2 Prefixes and suffixes discovered for some UMLS semantic types.

Semantic type	*Prefix*	*Suffix*
Amino Acid, Peptide, or Protein	acetyl, hydro, methyl	ase, amide, nyl
Biologically Active Substance	dihydro, hydro, iso	ene, ium, rol
Carbohydrates	deoxy, hydro, poly	ethyl, mycin, oxy

- *Identifying prefixes:* We stored this list of words in a *trie* [19], which is a multiway tree structure useful for storing strings over an alphabet. In a trie all strings sharing a common stem or prefix hang off a common node. An example trie for the prefixes *dehydro* and *deoxy* as well as the words *isobutyl* and *isoleucine* is shown in **Figure 2**. From this trie we find the nodes N, where the number of leaf nodes in the subtree rooted at N is above a threshold value. By listing the strings corresponding to these nodes, we can determine the common prefixes for the given semantic type. We chose the threshold so that the ratio of the number of terms having the prefix to the total number of terms for the given semantic type is sufficiently high.
- *Identifying suffixes:* Suffixes can be discovered using a similar procedure by storing the strings in reverse order.
- *Removing common English patterns:* Many of the prefixes and suffixes that are determined from UMLS may be common to all English words and not biologically significant. Therefore, we determined the common prefixes and suffixes of English words (determined from a general English corpus) and filtered these from the discovered biological patterns.

Using the above technique, we could identify common prefixes and suffixes for several UMLS semantic types. However, for some types such as *Element, Ion,* or *Isotope,* no frequently occurring prefixes and suffixes were discovered. In certain cases, we could group together concepts for related semantic types and identify patterns for a more general concept (parent of the semantic types in the UMLS hierarchy). An example is *Biologically Active Substance,* which consists of semantic types such as

Hormone and *Enzyme*. **Table 2** shows some of the prefixes and suffixes that were identified for three UMLS semantic types by our technique. Terms having these prefixes and suffixes in the biomedical literature can be identified as biological terms by the rule engine. Moreover, if the prefixes and suffixes are not present in more than one semantic type, the engine can also classify them on the basis of the UMLS Semantic Network. However, many prefixes and suffixes are common among several semantic types (for example, *hydro*).

Evaluation

We used the Genia corpus to evaluate our technique of extracting prefixes and suffixes from UMLS. For each discovered prefix and suffix for a semantic type, we calculated the number of biological and nonbiological terms having that prefix and suffix. If the discovered prefix/suffix is biologically relevant, the number of biological terms having that pattern will be significantly higher than the number of nonbiological terms. On the basis of this observation, we evaluate the effectiveness of our pattern (prefix/suffix) extraction technique for a semantic type as follows:

- Let m be the total number of biological terms having the prefixes/suffixes discovered for a semantic type.
- Let n be the total number of nonbiological terms having the prefixes/suffixes discovered for a semantic type.
- The *precision* of a pattern (prefix/suffix) for a semantic type is then calculated as $m/(m + n)$.

Table 3 shows the results for several semantic types. In most cases the precision is more than 80%, indicating that biological terms can be identified using the discovered prefixes and suffixes with a high degree of accuracy. However, the table also indicates that for certain semantic types such as *Plant,* other techniques are needed to improve the precision.

Table 3 Precision of extracted patterns from UMLS mining.

Semantic type	*Prefix* (%)	*Suffix* (%)
Amino Acid, Peptide, or Protein	90.45	88.79
Biologically Active Substance	88.16	87.69
Carbohydrates	89.6	84.26
Organophosphorus Compound	88.10	85.70
Plant	74.6	83.17

5. Term classification using context disambiguation

BioAnnotator also attempts to determine the semantic class for each identified biological term. If a term is identified using UMLS, the UMLS semantic class is associated with the term. If a term is identified using LocusLink, its semantic class is *Gene or Genome: LocusLink*. Gene names identified by GeneAlias are classified similarly. If a phrase is identified using the rule engine, we try to guess its class.

Unfortunately, the large number of polysemous words in the biological domain makes it difficult to characterize an identified entity. To disambiguate the admissible possibilities, the standard approach is to rely on the context in which the entity appears to provide additional clues to the intended meaning [16]. Often such disambiguation is either based on rules or based on machine learning approaches. However, the lack of consistency in the nomenclature of biological terms makes a purely rule-based approach impractical. Similarly, the lack of a suitably large quantity of labeled data makes machine-learning-based approaches harder to use. We combine rule-based and learning-based approaches for disambiguation in an attempt to remove the individual drawbacks of each approach. The three classes we considered for disambiguation are *proteins, genes,* and *RNA*.

Rule-based disambiguation

The rule engine identifies contextual and surface clues for the protein and gene classes. For example, if a word of the phrase matches the regular expressions for *GeneSignal, CellRoot,* or *ProteinSuffix,* it can be classified respectively as *Gene, Cell,* or *Protein*. If two words in the phrase give different clues, the last matched regular expression receives preference. For example, the phrase *protein-kinase proto oncogene* has contextual clues for both protein and gene. Since the last match is for *GeneSignal,* it can be classified as a gene. (Note that if we identify a term because it contains only words with special characters, numbers, or common biological prefixes or suffixes, we cannot guess its class.)

Learning-based disambiguation

Labeled data in the biological domain is hard to obtain. Because manual labeling is laborious and time-consuming, we use author-disambiguated occurrences of terms as training instances, as in [15]. Authors often include disambiguating information by preceding or following a term with its class (for example, *p53 gene*). We collect all of the words and phrases in the context of the term along with positional information to form a context vector. We exclude the locally disambiguating term (in this case *protein, gene,* or *RNA*) from the context. The learning approach we used was Naive Bayes. We found that, compared with other learning techniques, Naive Bayes allows easier scaling and significantly faster training and classification. The learning algorithm learns the class

Table 4 Performance of the learning-based disambiguator on the locally disambiguated dataset.

Feature characteristics	*Accuracy (%)*	
	Two-way	*Three-way*
Bag of words[†]	80.56	77.95
Positional information	85.83	83.16
Case-sensitive	86.37	85.47
Fixed window	76.07	65.77
Stop-word removal	84.54	82.93
Retain phrases (flexible window)	93.88	92.56
Retain phrases (fixed window)	71.79	70.57

[†]A collection of words without their positional information.

assignment probabilities given different contexts. Given a term, we collect all of the words or phrases near the term along with their distance from the term. The method assigns a class c_i ($i \in 1, \cdots, N$), where N is the total number of possible classes) to a term given a context vector $\mathbf{K}$ ($= \{k_1, \cdots, k_m\}$) such that the probability $P(c_i/\mathbf{K})$ is maximized. Using the Bayes rule,

$$P(c_i/\mathbf{K}) = \frac{P(\mathbf{K}/c_i)}{P(\mathbf{K})} P(c_i), \tag{1}$$

where $P(c_i)$ is the *a priori* probability of class c_i. The *a priori* probability of the evidence $P(\mathbf{K})$ is a constant for all senses and hence does not influence the maximization of $P(c_i/\mathbf{K})$. Also, because of the naive Bayes independence assumption for the features, we can rewrite (1) as

$$P(c_i) \times \prod_{j=1}^{m} P(k_j/c_i), \tag{2}$$

where these probabilities are computed from the training data via maximum-likelihood estimation.

Persistence of sense

The constraints between different occurrences of a term in a document can be usefully exploited. In [14] the one-sense-per-discourse [20] hypothesis was applied to disambiguation, exploiting the fact that the sense of a term is highly consistent within a given document. In biological abstracts, however, we found that even though the document is short, a term can occur in multiple senses within it. Significantly, though, we noted that within the document, once the sense of a biological term is established, the authors continue to use the term in the same sense. A change in sense is often indicated by explicit inclusion of a disambiguating term. We looked at 20 MedLine abstracts and found that out of 113 occurrences of an entity following its earlier use in the document, 68 retained their sense (60.18%) and 34 of the occurrences were used in a different sense but were locally disambiguated. Only one occurrence had a change of sense without being disambiguated locally.

Combined rule-based and learning-based disambiguation

This disambiguation is based on combined evidence from the different sources. For each term to be classified, the overall evidence score β_i is computed as

$$\beta_i = \alpha_1 W_{\text{class}} + \alpha_2 W_{\text{persistentsense}} + P_i, \tag{3}$$

where P_i is the probability that the term belongs to sense i as predicted by the learning-based classifier, and W_{class} and $W_{\text{persistentsense}}$ respectively are the weights assigned to the evidence from the rule engine and persistent sense. At present these weights are heuristically determined; in the future, we plan to learn them automatically. The term $\alpha_1 = 1$ when the class determined by the rule engine is the sense i, and 0 otherwise; $\alpha_2 = 1$ when the persistent sense is the sense i, and 0 otherwise. The sense with the maximum overall evidence score is assigned to the term.

Evaluation

We evaluated the learning-based disambiguator on locally disambiguated terms using tenfold cross-validation. Our training and testing data was a set of 11,000 MedLine abstracts obtained by querying the PubMed database using a list of gene/protein names. Out of a total of 86,809 occurrences of gene/protein names specified in the list, 9,217 (10.62%) were author-disambiguated. The context window size[3] is fixed at 8. In **Table 4** we summarize the results of the classification using the learning-based disambiguator. The table shows that retaining phrases improves performance greatly. Such phrases were marked by BioAnnotator and were used as the context vector for a term without splitting them into the component words. We also noticed that not using the positional information resulted in a steep drop in performance. The classification accuracy in the two-way case of using two classes shows better results than the three-way case of using all three classes (proteins, genes, and RNA).

Evaluation of the combined evidence model using locally disambiguated occurrences is unfair, since the rules used in this model utilize this local disambiguation information and would hence yield accuracy figures close to 99%. In this case, the persistence of sense information is also not useful, since the disambiguating word allows classification without using persistence information. To test the combined disambiguator, we prepared a test bed of 19 hand-tagged documents containing 339 occurrences of gene/protein/RNAs.

[3] The term *context window size* refers to the number of words examined at either side of the phrase being disambiguated.

For the combined disambiguator, the weight W_{class} is set to 2.0 to give the rule-based classifier highest priority. The weight $W_{persistentsense}$ is set to 0.55. In **Table 5**, classification results for different configurations of the disambiguator are shown on the hand-tagged dataset. The performance of the disambiguator based on learning only is rather poor on this set compared with its performance on the earlier locally disambiguated set. The possible reasons for the drop could be the following:

- Because of the small size of the test set, it may not be representative.
- The locally disambiguated training instances may not be sufficient as examples for all cases.
- The system assigns senses to cases which are not truly classifiable (e.g., *tumor suppressor gene activity*).

However, the use of persistence of sense and rules enhances the performance greatly.

Table 5 Performance of different configurations of the disambiguator on the hand-tagged dataset.

Mode	*Accuracy (%)*	
	Two-way	*Three-way*
Learning-based	63.33	59.05
Learning-based with persistence of sense	69.33	65.08
Combined model (learning + rules + persistence)	81.00	76.19

6. Conclusion

In this paper we have presented a biological annotation system which uses a variety of knowledge sources along with syntactic information, term properties, and contextual clues to identify and classify known and new terms. BioAnnotator attempts to mine biomedical knowledge sources to learn extraction patterns in order to identify new and unknown biological terms. We have also developed a technique for context disambiguation. Our evaluation shows that the system has good precision and recall.

Systems such as BioAnnotator that extract and classify terms from biomedical literature can be utilized to support more complex text analysis tasks for researchers. We believe that the *semistructured* documents that result from BioAnnotator provide opportunities for additional types of knowledge discovery from biomedical document corpuses. For example, in the extraction of relations between biological entities (for example, protein–protein interactions), it is first necessary to recognize and classify the entities taking part in the interactions [21]. Term extraction is also useful for automatically updating biomedical databases such as SwissProt [22], which are at present largely hand-curated. We have also developed a Web-based application for a semantic search of online biomedical research publications based on the annotated documents. A traditional keyword-based search retrieves only publications that contain the specified keywords. Thus, searching on genes will not retrieve a very relevant document that discusses p53 (a type of gene). On the other hand, since we have annotated the documents with the semantic classes of the biological terms, our semantic search application will be able to retrieve all of the relevant documents. Another application of the annotations produced by the BioAnnotator to discover the biological significance of gene clusters is presented in [2].

Future work is planned along various lines:

- At present BioAnnotator learns only affixes of biological terms from UMLS. Besides prefixes and suffixes, biological concept names often contain root forms that give an indication of their class. For example, many cell names contain *blast, cyt,* or *phore* (for example, *leucocytes*). We must extend our algorithm to learn these root forms. We also must also develop techniques to learn other types of patterns that can be used to identify biological terms. For example, protein, cell, or gene signals can be determined from an annotated corpus.
- Our present evaluation of the UMLS mining technique determines only whether the discovered prefixes and suffixes are biologically relevant. We must also evaluate whether the discovered prefixes and suffixes are relevant for a particular semantic type.
- Our current technique of context disambiguation is restricted to proteins, genes, and RNA. We must extend the algorithm so that it can disambiguate among other semantic classes also. Moreover, since many of the gene names in LocusLink are common English words (for example, *we, high, star*), an effective method of disambiguating between two words is required. We plan to use the part of speech of the words for this purpose.
- Acquiring labeled data for learning is extremely difficult in this domain. In this work we have exploited the fact that explicitly disambiguated terms are often present in documents. In the future we plan to utilize metadata, such as MeSH descriptors assigned manually to each MedLine abstract, for disambiguation.

References

1. MedLine; see *http://www.ncbi.nlm.nih.gov/PubMed/*.
2. L. V. Subramaniam, S. Mukherjea, P. Kankar, B. Srivastava, V. Batra, P. Kamesam, and R. Kothari, "Information Extraction from Biomedical Literature: Methodology, Evaluation and an Application," *Proceedings*

of the ACM Conference on Information and Knowledge Management, New Orleans, 2003, pp. 410–417.
3. UMLS; see *http://umlsks.nlm.nih.gov/*.
4. LocusLink; see *http://www.ncbi.nlm.nih.gov/locuslink/*.
5. *Proceedings of the Sixth Message Understanding Conference (MUC),* Columbia, MD; Morgan Kaufmann Publishers, 1995.
6. K. Humphreys, G. Demetriou, and R. Gaizauskas, "Two Applications of Information Extraction to Biological Science: Enzyme Interactions and Protein Structures," *Proceedings of the Pacific Symposium on Biocomputing*, Hawaii, 2000, pp. 502–513.
7. K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi, "Toward Information Extraction: Identifying Protein Names from Biological Papers," *Proceedings of the Pacific Symposium on Biocomputing*, Hawaii, 1998, pp. 707–718.
8. R. Gaizauskas, G. Demetriou, and K. Humphreys, "Term Recognition and Classification in Biological Science Journal Articles," *Proceedings of the Computational Terminology for Medical and Biological Applications: Workshop of the 2nd International Conference on Natural Language Processing*, Patras, Greece, 1998, pp. 37–44.
9. N. Collier, C. Nobata, and J. Tsujii, "Extracting the Names of Genes and Gene Products with a Hidden Markov Model," *Proceedings of the 18th International Conference on Computational Linguistics*, Saarbrucken, Germany, 2000, pp. 201–207.
10. M. Andrade and A. Valencia, "Automatic Extraction of Keywords from Scientific Text: Application to the Knowledge Domain of Protein Families," *BioInform.* **4,** No. 7, 600–607 (1998).
11. Y. Park, "Identification of Probable Real Words: An Entropy-based Approach," *Proceedings of the Association for Computational Linguistics (ACL-02) Workshop on Unsupervised Lexical Acquisition*, Saarbrucken, Germany, 2002, pp. 1–8.
12. L. Hirshman, J. C. Park, J. Tsujii, L. Wong, and C. H. Wu, "Accomplishments and Challenges in Literature Data Mining for Biology," *BioInform. Rev.* **18,** No. 12, 1553–1561 (2002).
13. C. Nobata, N. Collier, and J. Tsujii, "Automatic Term Identification and Classification," *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium*, Beijing, China, 1999, pp. 369–374.
14. D. Yarowsky, "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods," *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA, 1995, pp. 189–196.
15. V. Hatzivassiloglou, P. A. Duboue, and A. Rzhetsky, "Disambiguating Proteins, Genes, and RNA in Text: A Machine Learning Approach," *Bioinform.* **1,** No. 1, 1–10 (2001).
16. P. Resnik and D. Yarowsky, "A Perspective on Word Sense Disambiguation Methods and Their Evaluation," *Proceedings of the Association for Computational Linguistics (ACL/SIGLEX) Workshop*, Washington, DC, 1997, pp. 79–87.
17. P. Zweigenbaum and N. Grabar, "A Contribution of Medical Terminology to Medical Language Processing Resources: Experiments in Morphological Knowledge Acquisition from Thesauri," *Proceedings of the International Medical Information Association (IMIA-WG6) Conference*, 1999, pp. 131–141.
18. Genia Corpus; see *http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/*.
19. E. Fredkin, "TRIE Memory," *Commun. ACM* **3,** 490–500 (1960).
20. A. W. Gale, K. W. Church, and D. Yarowsky, "One Sense Per Discourse," *Proceedings of the DARPA Speech and Natural Language Workshop*, Harriman, NY, 1992, pp. 233–237.
21. J. Thomas, D. Milnard, C. Ouzounis, S. Pulman, and M. Carroll, "Automatic Extraction of Protein Interactions from Scientific Abstracts," *Proceedings of the Pacific Symposium on Biocomputing*, Hawaii, 2000, pp. 541–551.
22. A. Bairoch and R. Apweiler, "The SWISS-PROT Protein Sequence Databank and Its New Supplement TrEMBL, *Nucleic Acids Res.* **25,** 31–36 (1997).

Received October 15, 2003; accepted for publication January 6, 2004; Internet publication September 1, 2004

Sougata Mukherjea *IBM Research Division, IBM India Research Laboratory, Block I, Indian Institute of Technology (IIT), Hauz Khas, New Delhi 110016 (smukherj@in.ibm.com).* Dr. Mukherjea is a Research Staff Member in the IBM India Research Laboratory. He received his bachelor's degree from Jadavpur University, Calcutta, his M.S. degree from Northeastern University, Boston, and his Ph.D. degree from the Georgia Institute of Technology, Atlanta (all in computer science). Before joining IBM, he held research and software architect positions in Silicon Valley companies including NEC USA, BEA Systems, and Verity. His research interests include information visualization and retrieval and applications of text mining in areas such as Web search and bioinformatics.

L. Venkata Subramaniam *IBM Research Division, IBM India Research Laboratory, Block I, Indian Institute of Technology (IIT), Hauz Khas, New Delhi 110016 (lvsubram@in.ibm.com).* Dr. Subramaniam has been a Research Staff Member in the IBM India Research Laboratory since 1998. He received his bachelor's degree in electronics and communication engineering from the P. E. S. College of Engineering, his M.S. degree in electrical engineering from Washington University, St. Louis, and his Ph.D. degree in electronics from the Indian Institute of Technology, Delhi. His research interests include unstructured information management, statistical natural language processing, machine learning, text mining, and the application of these technologies.

Gaurav Chanda *IBM Research Division, IBM India Research Laboratory, Block I, Indian Institute of Technology (IIT), Hauz Khas, New Delhi 110016 (ggaurav_chanda_81@yahoo.com).* Mr. Chanda has been a master's student in the College of Computing at the Georgia Institute of Technology since August 2003. He received his B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Delhi, in May 2003. From May to July 2002, he worked as a summer intern at l'Institut National de Recherche en Informatique et Automatique (INRIA), Rocquencourt, and in the summer of 2003 he worked as a trainee at the IBM India Research Laboratory. His research interests include computer vision, machine learning, and computer graphics.

Sriram Sankararaman *IBM Research Division, IBM India Research Laboratory, Block I, Indian Institute of Technology (IIT), Hauz Khas, New Delhi 110016 (rsrssr@yahoo.com).* Mr. Sankararaman is currently in the final year of the B.Tech. program in computer science and engineering at the Indian Institute of Technology, Madras. His research interests include networks, security, multimedia systems, bioinformatics, and quantum computing. He contributed to this paper while visiting the IBM India Research Laboratory as a summer trainee in 2003.

Ravi Kothari *IBM Research Division, IBM India Research Laboratory, Block I, Indian Institute of Technology (IIT), Hauz Khas, New Delhi 110016, India (rkothari@in.ibm.com).* Dr. Kothari received the B.E. degree (with distinction) from Birla Institute of Technology, India, the M.S. degree from Louisiana State University, and the Ph.D. degree from West Virginia University, all in electrical engineering. He joined the Department of Electrical and Computer Engineering (ECECS) and Computer Science of the University of Cincinnati (UC), Ohio, as an Assistant Professor; he later became a tenured Associate Professor and Director of the Artificial Neural Systems Laboratory. Since June of 2002, Dr. Kothari has worked in the IBM India Research Laboratory, New Delhi. His theoretical areas of interest include pattern recognition, machine learning, data mining, self-organization, and dimensionality reduction. He is currently applying these technologies in commerce, life sciences, and other domains. Dr. Kothari received the William E. Restemeyer Teaching Excellence Award from the Department of ECECS at UC in 1994 and the Eta Kappa Nu Outstanding Professor of the Year Award from the Department of ECECS at UC in 1995. He serves as an Associate Editor of the *IEEE Transactions on Knowledge and Data Engineering* and the *IEEE Transactions on Neural Networks*, as an Editorial Board member of the *Pattern Analysis and Applications Journal*, and as a member of the program committees of numerous conferences. Dr. Kothari is a member of Sigma Xi, Eta Kappa Nu, and Phi Kappa Phi; he is an IEEE Distinguished Visitor (2003–2005).

Vishal Batra *IBM Research Division, IBM India Research Laboratory, Block I, Indian Institute of Technology (IIT), Hauz Khas, New Delhi 110016 (bvishal@in.ibm.com).* Mr. Batra is a Technical Staff Member at the IBM India Research Laboratory. He received his bachelor's degree from the Indian Institute of Technology, Kharagpur, and has worked with IBM Research since June 2001. Prior to joining IBM, he worked on a variety of projects ranging from Internet banking to healthcare. His areas of interests are distributed and parallel systems. At IBM, Mr. Batra has worked on unstructured information management systems for the UIMA and WebFountain projects. His current focus is on WebFountain partner enablement and integration.

Deo Bhardwaj *IBM Research Division, IBM India Research Laboratory, Block I, Indian Institute of Technology (IIT), Hauz Khas, New Delhi 110016 (bhardwa@in.ibm.com).* Mr. Bhardwaj is a Technical Staff Member at the IBM India Research Laboratory. He received his bachelor's degree in computer science from the Regional Engineering College, Rourkela. Prior to joining IBM, he was a software engineer at Kanrad Technology (Bangalore), Rebbrigade (Johannesburg), and Avery India Limited (Faridabad), and contributed to the design and development of various projects. His areas of interest are text analysis, unstructured information management, and natural language processing for bioinformatics and Web search.

Biplav Srivastava *IBM Research Division, IBM India Research Laboratory, Block I, Indian Institute of Technology (IIT), Hauz Khas, New Delhi 110016 (sbiplav@in.ibm.com).* Dr. Srivastava is a Research Staff Member at the IBM India Research Laboratory. He received his Ph.D. and M.S. degrees (with an emphasis on planning, scheduling, and distributed computing) from Arizona State University and his B.Tech. degree from the Institute of Technology (IT-BHU), India, all in computer science. Before joining IBM, Dr. Srivastava held various positions at a Silicon Valley startup company, Philips Semiconductor, and TCS; his focus was on data integration and electronic design automation. At IBM, he has worked in data management for bioinformatics and multidevice computing applications. His current focus is on practical planning for business applications and dynamic process/data integration.