Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, pp. 1-8.

Gene Name Extraction Using FlyBase Resources

Alex Morgan
amorgan@mitre.org
Lynette Hirschman
lynette@mitre.org

The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420

Alexander Yeh
asy@mitre.org
Marc Colosimo
mcolosim@brandeis.edu

Abstract

Machine-learning based entity extraction requires a large corpus of annotated training to achieve acceptable results. However, the cost of expert annotation of relevant data, coupled with issues of inter-annotator variability, makes it expensive and time-consuming to create the necessary corpora. We report here on a simple method for the automatic creation of large quantities of imperfect training data for a biological entity (gene or protein) extraction system. We used resources available in the FlyBase model organism database; these resources include a curated lists of genes and the articles from which the entries were drawn, together a synonym lexicon. We applied simple pattern matching to identify gene names in the associated abstracts and filtered these entities using the list of curated entries for the article. This process created a data set that could be used to train a simple Hidden Markov Model (HMM) entity tagger. The results from the HMM tagger were comparable to those reported by other groups (F-measure of 0.75). This method has the advantage of being rapidly transferable to new domains that have similar existing resources.

1 Introduction: Biological Databases

There is currently an information explosion in biomedical research. The growth of literature is roughly exponential, as can be seen in Figure 1 which shows the number of literature references in FlyBase[1] organized by date of publication over a hundred year span.[2] This growth of literature makes it daunting for researchers to keep track of the information, even in very small subfields of biology.

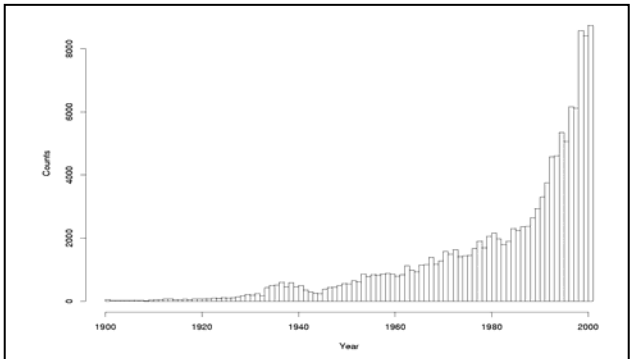

Figure 1: FlyBase References, 1900-2000

Increasingly, biological databases serve to collect and organize published experimental results. A wide range of biological databases exist, including model organism databases (e.g., for mouse[3] and yeast[4]) as well as various protein databases (e.g., Protein Information Resource[5] (PIR) or SWISS-PROT[6] and interaction databases such as the Biomolecular Interaction Network Database[7] (BIND). These databases are created by a process of curation, which is done by Ph.D. biologists who read the published literature to cull experimental findings and relations. These facts are organized into a set of structured fields of a database and

[1] FlyBase is a database that focuses on research in the genetics and molecular biology of the fruit fly (*Drosophila melangastor*), a model organism for genetics research: http://www.flybase.org.

[2] Of course most of these early references in FlyBase are not in electronic form. The FlyBase database has been in existence since 1993.

[3] http://www.informatics.jax.org/

[4] http://genome-www.stanford.edu/Saccharomyces/

[5] http://pir.georgetown.edu/pirwww/pirhome3.shtml

[6] http://us.expasy.org/sprot/

[7] http://www.bind.ca/

linked to the source of information (the journal article). As a result, curation is a time-consuming and expensive process; database curators are increasingly eager to adopt text mining and natural language processing techniques to make curation faster and more consistent. As a result, there has been growing interest in the application of entity extraction and text classification techniques to the problem of biological database curation [Hirschman02].

2 Entity Extraction Methods

There are two approaches to entity extraction. The first requires manual or heuristic creation of rules to identify the names mentioned in text; the second uses machine learning to create the rules that drive the entity tagging. Heuristic systems require expert developers to create the rules, and these rules must be manually changed to handle new domains. Machine-learning based systems are dependent on large quantities of tagged data, consisting of both positive and negative examples.[8] Figure 2 shows results from the IdentiFinder system [Bikel99] illustrating that performance increases roughly with the log of quantity of training data. Given the expense of manual annotation of large quantities of data, the challenge for the machine learning approach is to find ways of creating sufficient quantities of training data cheaply.

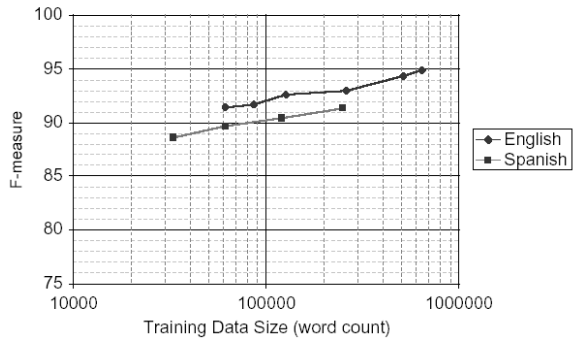

Figure 2: Performance of BBN's IdentiFinder named entity recognition system relative to the amount of training data, from [Bikel99]

Overall, hand-crafted systems seem to outperform learning-based systems for biology. However, it is clear that the quantities of training have been small, relative to the results reported for entity extraction in e.g., newswire [Hirschman03]. There are several published sets of performance results for automatic named biological entity extraction systems. The system of Collier et al. [Collier00] uses a hidden Markov model to achieve an F-measure[9] of 0.73 when trained on a corpus of 29,940 words of text from 100 MEDLINE abstracts. Contrast this with Figure 2, which reports results using over 600,000 words of training data, and an F-measure of 0.95 for English newswire entity extraction (and 0.91 for Spanish).

Krauthammer et al. [Krauthammer00] have taken a somewhat different approach which encodes characters as 4-tuples of DNA bases; they then use BLAST together with a lexicon of gene names to search for 'gene name homologies'. They report an F-measure of 0.75 without the use of a large set of rules or annotated training data.

The PASTA system [Gaizauskas03] uses a combination of heuristic and machine-learned rules to achieve a higher F-measure over a larger number of classes: F-measure of 0.83 for the task of identifying 12 classes of entities involved in the description of roles of residues in protein molecules. Because they used heuristic rules, they were able to get these results with a relatively small training corpus of 52 MEDLINE abstracts (roughly 12,000 words).

These results suggest that machine learning methods will not be able to compete with heuristic rules until there is a way to generate large quantities of annotated training data. Biology has the advantage that there are rich resources available, such as lexicons, ontologies and hand-curated databases. What is missing is a way to convert these into training corpora for text mining and natural language processing. Craven and Kumlien [Craven99] developed an innovative approach that used fields in a biological database to locate abstracts which mention physiological localization of proteins. Then via a simple pattern matching algo-

[8] For negative examples, the "closed world" assumption generally is taken to apply: if an entity is not tagged, it is assumed to be a negative example.

[9] $F = (2 \cdot \text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall})$

Manning D, Schutze H. *Foundations of Statistical Natural Language Processing*, 2002: p 269.

rithm, they identified those sentences where the relation was mentioned and matched these with entries in the Yeast Protein Database (YPD). In this way, they were able to automatically create an annotated gold standard, consisting of sentences paired with the curated relations derived from those sentences. They then used these for training and testing a machine-learning based system. This approach inspired our interest in using existing resources to create an annotated corpus automatically.

3 FlyBase: Organization and Resources

We focused on FlyBase because we had access to FlyBase resources from our work in the creation of the KDD 2002 Cup Challenge Task 1 [Yeh03]. Through this work, we had become familiar with the multi-stage process of curation. An early task in the curation pipeline is to determine, for a given article, whether there are experimental results that need to be added to the database. This was the task used as the basis for the KDD text data mining "challenge evaluation". A later task in the pipeline creates a list of the *Drosophila* genes discussed in each curated article. This is the task we focus on in this paper.

An example of a FlyBase entry can be seen in Figure 3 which shows part of the record for the gene *Toll*. Under **Molecular Function** and **Biological Process** we see that the gene is responsible for encoding a transmembrane receptor protein involved in antimicrobial humoral response (part of the innate immune system of the fly). We see further that "Tl" and "CG5490" are synonymous for *Toll* (top of the entry next to **Symbol**), and the link **Synonyms** leads to a long synonym list which includes: "Fs(1)Tl", "dToll", "CT17414", "Toll-1", "Fs(3)Tl", "mat(3)9", "mel(3)10", and "mel(3)9". Many of these facts about *Toll* are linked to a particular literature reference in the database. For example, following the link for **Transcripts** will lead to a page with links to the abstract of a paper by Tauszig et al. [Tauszig00] which reports on experiments which measured the lengths of RNA transcribed from the *Toll* gene.

For FlyBase, *Drosophila* genes are the key biological entities; each entity (e.g., gene) is associated with a unique identifier for the underlying physical entity. If there were a one-to-one relationship between gene name and unique identifier, the gene identification task would be straightforward. However, both polysemy and synonymy occur frequently in the naming of biological entities, and the gene names of *Drosophila* are considered to be particularly problematic because of creative naming conventions[10]. For example, "18 wheeler", "batman", and "rutabaga" are all *Drosophila* gene names. A single entity (as represented by a unique identifier) may have a number of names like *Toll* or even *ATPα*, which has 38 synonyms listed in FlyBase.

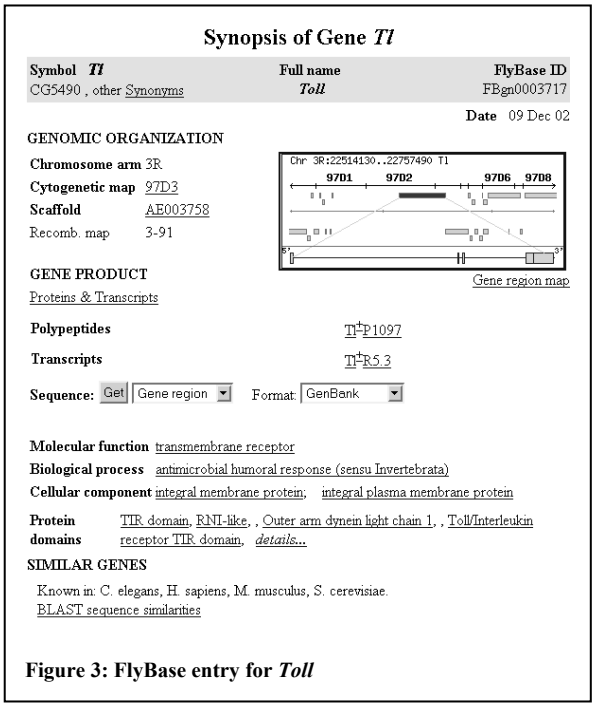

Figure 3: FlyBase entry for *Toll*

3.1 Resources

We obtained a copy of part the FlyBase database,[11] including the lists of genes discussed in each paper examined by the curators. Using the BioPython[12] modules, we were able to obtain MEDLINE abstracts for 15,144 for these papers. We decided to

[10] At the other end of the spectrum is the yeast nomenclature which is strictly controlled – see <http://genome-www.stanford.edu/Saccharomyces/gene_guidelines.shtml> for nomenclature conventions.

[11] Special thanks to William Gelbart, David Emmert, Beverly Matthews, Leyla Bayraktaroglu, and Don Gilbert.

[12] http://www.biopython.org/

set aside the same articles used in the KDD Cup Challenge [Yeh03] for evaluation purposes. This left a training set of 14,033 abstracts, consisting of a total of 2,664,324 lexemes identified by our tokenizer.

It was only with some reluctance that we decided to focus on journal abstracts. From our earlier work, we recognized that the majority of the information entered into FlyBase is missing from the abstracts and can be found only in the full text of the article [Hirschman03]. However, due to copyright restrictions, there is a paucity of freely available full text for journal articles. What articles are available in electronic form vary in their formatting, which can cause considerable difficulty in automatic processing. MEDLINE abstracts have a uniform format and are readily available. Many other experiments have been performed on MEDLINE abstracts for similar reasons.

We also created a synonym lexicon from FlyBase. We found 35,971 genes with associated 'gene symbols' (e.g. *Tl* is the gene symbol for *Toll*) and 48,434 synonyms; therefore, each gene has an average of 2.3 alternate naming forms, including the gene symbol. The lexicon also allowed us to associate each gene with one a unique FlyBase gene identifier, providing "term normalization."

4 Experiments

For purposes of evaluation, our task was the identification of mentions of *Drosophila* genes in the text of abstracts. We also included mentions of protein or transcript where the associated gene shared the same name. This occurs when, for example, the gene name appears as a pre-nominal modifier, as in "the zygotic *Toll* protein". We did **not** include mentions of protein complexes because these are created out of multiple polypeptide chains with multiple genes (e.g., *hemoglobin*). We also did not include families of proteins or genes (e.g. *lectin*), particular alleles of a gene, genes which are not part of the natural *Drosophila* genome such as reporter genes (e.g. *LacZ*), and the names of genes from other organisms (e.g. *sonic hedgehog*, the mammalian gene homologous to the *Drosophila hedgehog* gene).[13]

4.1 Background

Our initial experiment [Hirschman03] had looked at creating a gene name finder by simple pattern matching, using the extensive FlyBase list of genes and their synonyms and identifying each mention which occurred in the lexicon with the appropriate unique identifier. This yielded spectacularly poor results: recall[14] on the full papers was quite high (84%), but precision was 2%! For abstracts, the recall was predictably lower (31%) and precision remained low at 7%. Our analysis showed that polysemy (described in Section 5) and the large intersection of gene names with common English words caused most of the performance problems. In the initial run, where a name was ambiguous, we recorded all gene identifiers; this raised recall but lowered precision. After removing all the names which were ambiguous for a gene, precision climbed to 5% for full papers and 17% in abstracts, with a corresponding drop in recall (77% for full papers, 28% for abstracts). We also tried a few simple filters, such as ignoring all terms three characters or less in length, but the best precision we could achieve was 29% in abstracts, certainly unacceptable.

We were, however, encouraged by the relatively high recall in full papers. Analysis showed that many of the missing names were contained only in figures or tables that had not been downloaded. While these were counted as recall errors when compared to the FlyBase curation, there were, in fact, no mentions of these genes in the text that had been downloaded for this experiment. Similarly, for abstracts, while the recall appeared low compared to the complete set of genes discussed in the full paper, these genes were simply not mentioned in the abstract. So from an information extraction

[13] There are no curated lists of complexes or families in FlyBase, so we did not train a tagger for these tasks. In our manual curation, we did create separate tags for complexes and families, since we believe that these will be important for future tasks.

[14] Note that these measures of recall and precision are based on the list of unique Drosophila genes curated in a paper. This is quite different from recall and precision measuring the mentions of gene names in a paper. We used the measure of unique genes in a paper because this allowed us to take advantage of the existing FlyBase expert curated resources.

point of view, the simple pattern matching achieved a very high recall for genes mentioned in the text being processed.

4.2 Generating Noisy Training Data

The initial experiment demonstrated that exact match using rich lexical resources was not useful on its own. However, we realized that we could use the lists of curated genes from FlyBase to constrain the possible matches within an abstract – that is, to "license" the tagging of only those genes known to occur in the curated full article. Our hope was that this filtered data would provide large quantities of cheap but imperfect or noisy training data.

Our next experiment focused on generating this large but noisy training corpus. We used our internal tokenizer, *punctoker*, originally designed for use with newswire data. There were some errors in tokenization, since biological terms have a very different morphology from newswire– see [Cohen02] for an interesting discussion of tokenization issues. Among the problems in tokenization were uses of "-" instead of white space, or "/" to separate recombinant genes. However, an informal examination of errors did not show tokenization errors to be a significant contributor to the overall performance of the entity extraction system.

To perform the pattern matching, we created a suffix tree of all the synonyms known to FlyBase for those genes. This was important, since many biological entity names are multi-word terms. We then used longest-extent pattern matching to find candidate mentions in the abstract of the paper. The system tagged only terms licensed by the associated list of genes for the abstract, assigning the appropriate unique gene identifier. Even with the FlyBase filtering, this method resulted in some errors. For example, an examination of an abstract describing the gene *to* revealed the unsurprising result that all the uses of the word "to" did not refer to the gene. However, the aim was to create data of sufficient quantity to lessen the effects of this noise.

4.3 Evaluation

In order to measure performance, we created a small doubly annotated test corpus. We selected a sample of 86 abstracts and had two annotators mark these abstracts for gene name mentions as previously described. Mentions of families and foreign genes were also identified with different tags during this process, but not evaluated. One curator was a professional researcher in biology with experience as a model organism genome database curator (Colosimo). This set of annotations was taken as the "gold-standard". The second annotator was the system developer with no particular annotation experience (Morgan). With two annotators, we were able to measure inter-annotator agreement (F-measure of 0.87). We also measured the quality of the automatically created training data by using the lexical pattern matching procedure with filtering to generate annotations for 86 abstracts in the test set. The F-measure was 0.83, when compared against the gold standard, shown in Table 1 below.

	F-measure	Precision	Recall
Training Data Quality	0.83	0.78	0.88
Inter-annotator Agreement	0.87	0.83	0.91

Table 1: Training data quality and inter-annotator agreement

4.4 HMM Tagging With Noisy Training Data

We now had a large quantity of noisy training data that we could use to train a statistical tagger. This methodology is illustrated in Figure 4. We chose the HMM-based trainable entity tagger *phrag*[15] [Palmer99] to extract the names in text. We trained *phrag* on different amounts of training data and measured performance. Our evaluation metric was the standard metric used in named entity

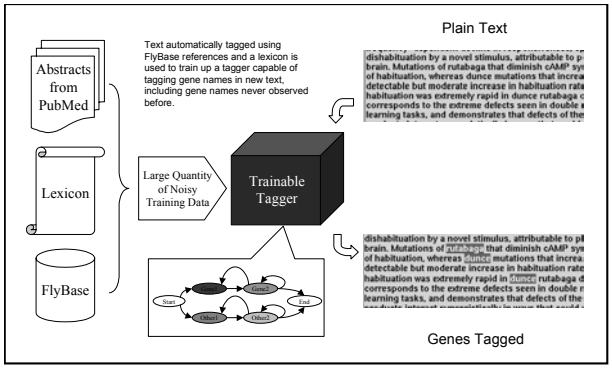

Figure 4: Schematic of Methodology

[15] *Phrag* is available for download at http://www.openchannelfoundation.org/projects/Qanda

evaluation, requiring the matching of a name's extent and tag (except that for our experiment, we were only concerned with one tag, Drosophila gene). Extent matching meant exact matching of gene name boundaries at the level of tokens: Exactly matching boundaries were considered a hit. Inexact answers are considered a miss. For example, a multiword gene name such as "fas receptor", which has been tagged for "fas" but not for "receptor" would constitute a miss (recall error) and a false alarm (precision error).

Table 2 shows the performance of the basic system as a function of the amount of training data. As with Figure 2, we see there is a diminishing return as the amount of training data is increased. At 2.6 million words or training data, *phrag* achieved an entity identification F-measure of 73%. We then made a simple modification of the algorithm to correct for variations in orthography due to capitalization and representation of Greek letters: we simply expanded the search for letters such as "Δ" to include "Delta" and "delta". By expanding the matching of terms using the orthographical and case variants, performance of *phrag* improved slightly, shown in Table 3, improving our best performance to an F-measure of 75%.

Figure 5 shows these results in a graphical form. Two things are apparent from this graph. Based on the results shown in Figure 2, we might expect the performance to be linear with the logarithm of the amount of training data, and in this case there is a rough fit with a correlation coefficient of .88. The other result which stands out is that there is considerable variation in the performance when trained on different training sets of the same size. We believe that this is due to the very limited amount of testing data.

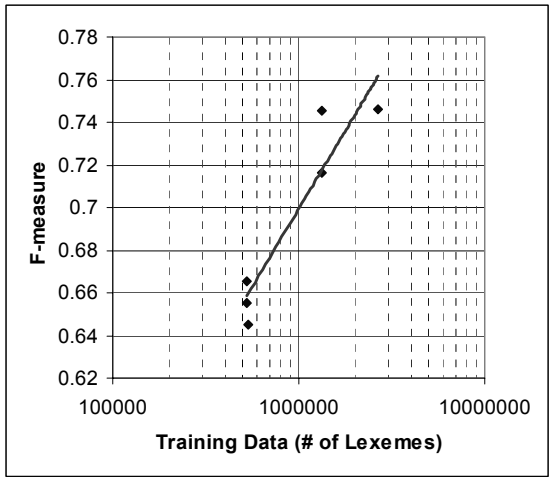

Figure 5: Performance as a function of the amount of training data. The line is a least-squares logarithmic fit with an R^2 value of .8814.

No Orthographic Correction			
Training Data	F-measure	Precision	Recall
531522	0.62	0.73	0.54
529760	0.64	0.75	0.56
1342039	0.72	0.80	0.65
2664324	0.73	0.79	0.67

Table 2: Performance as a function of training data

Orthographic Correction			
Training Data	F-measure	Precision	Recall
531522	0.65	0.76	0.56
529760	0.66	0.74	0.59
522825	0.67	0.76	0.59
1322285	0.72	0.77	0.67
1342039	0.75	0.80	0.70
2664324	0.75	0.78	0.71

Table 3: Improved performance with orthographical correction for Greek letters and case folding for term matching in training data

5 Error Analysis

We have identified three types of polysemy in *Drosophila* gene names in FlyBase. In some cases, one name (e.g., "Clock") can refer to two distinct genes: *period* or *Clock*. The term with the most polysemy is "P450" which is a family of genes and is listed as a synonym for 20 different genes in FlyBase. In addition, the same term is often used interchangeably to refer to the gene, RNA transcript, or the protein. [Hazivassloglou01] presents interesting results that demonstrate that experts only agree 78% of the time on whether a particular mention refers to a gene or a protein.[16] The most problematic type of polysemy occurs because many Drosophila gene names are also regular English words such as "white", "cycle", and "bizarre". There are some particularly troublesome examples that occur because of frequent use of short forms (abbreviations) of gene names, e.g., "we", "a", "not", and even "and" each occur as gene names. These short forms are often abbreviations for the full gene name. For example, the gene symbol of the gene *takeout* is "to", and the symbol for the

[16] The entity tagging task for FlyBase was defined to extract gene-or-protein names; however, in cases where the article talks only about the protein and not about the gene, the protein name may not appear on the list of curated genes for the article, leading to apparent false positives in tagging.

gene *wee* is "we". It may be that more sophisticated handling of abbreviations can address some of these issues.
An error analysis looking at the results of our statistical tagger demonstrated some unusual behavior. Because our gene name tagger *phrag* uses a first order Markov model, it relies on local context and occasionally makes errors such as not tagging all of the occurrences of the term "rutabaga" in an abstract about *rutabaga* as gene names. This certainly opens up the opportunity for some sort of post processing step to resolve these problems.

The fact that *phrag* uses this local context can sometimes be a strength, enabling it to identify gene names it has never seen. We estimated the ability of the system to identify new terms as gene names by substituting strings unknown to *phrag* in place of all the occurrences of gene names in the evaluation data. The performance of the system at correctly identifying terms it had never observed gave a precision of 68%, a recall of 22% and an F-measure of 33%. This result is relatively encouraging, compared with the 3.3% precision and 4.4% recall for novel gene names reported by Krauthammer. Recognizing novel names is important because the nomenclature of biological entities is constantly changing and entity tagging systems should to be able to rapidly adapt and recognize new terms.

6 Conclusion and Future Directions

We have demonstrated that we can automatically produce large quantities of relatively high quality training data; these data were good enough to train an HMM-based tagger to identify gene mentions with an F-measure of 75% (precision of 78% and recall of 71%), evaluated on our small development test set of 86 abstracts. This compares favorably with other reported results as described in Section 2, and as discussed below, we believe that we can improve upon these results in various ways. These results are still considerably below the results from [Gaizauskas03] and may be too low to be useful as a building block for further automated processing, such as relation extraction. However, in the absence of any shared benchmark evaluation sets, cross-system performance cannot be evaluated since the task definition and evaluation corpora differ from system to system.

We plan to take this work in several directions. First, we believe that we can improve the quality of the underlying automatically generated data, and with this, the quality of the entity tagging. There are several things that could be improved.

A morphological analyzer trained for biological text would eliminate some of the tokenization errors and perhaps capture some of the underlying regularities, such as addition of Greek letters or numbers (with or without preceding hyphen) to specify sub-types within a gene family. There can also be considerable semantic content in gene names and their formatting. For example, many *Drosophila* genes are differentiated from the genes of other organisms by prepending a "d" or "D", such as "dToll". Gene names can also be explicit descriptions of their chromosomal location or even function (e.g. *Dopamine receptor*).

The problem of matching abbreviations has been tackled by a number of researchers [e.g. Pustejovsky02 and Liu03]. As was mentioned above, it seems that ambiguity for "short forms" of gene names could be partially resolved by detecting local definitions for abbreviations. It should also be possible to apply part of speech tagging and corpus statistics to avoid mis-tagging of common words, such as “to” or “and”.

In the longer term, this methodology provides an opportunity to go beyond gene name tagging for *Drosophila*. It can be extended to other domains that have comparable resources (e.g. other model organism genome databases, other biological entities), and entity tagging itself provides the foundation for more complex tasks, such as relation extraction (e.g. using the BIND database) or attribute extraction (e.g. using FlyBase to identify attributes such as RNA transcript length, associated with protein coding genes).

Second, the existence of a synonym lexicon with unique identifiers provides data for term normalization, a task of potentially greater utility to biologists than the tagging of every mention in an article. There are currently few corpora with annotated term normalization; using the methodology outlined here makes it possible to produce large quantities of normalized data. The identification

and characterization of abbreviations and other transformations would be particularly important in normalization.
By exploiting the rich set of biological resources that already exist, it should be possible to generate many kinds of corpora useful for training high-quality information extraction and text mining components.

References

Bikel D, Schwartz R, Weischedel R. An Algorithm that Learns What's in a Name. *Machine Learning, Special Issue on Natural Language Learning* 34 (1999):211-31.

Cohen KB, Dolbey A, Hunter L. "Contrast and variability in gene names." *Proceedings of the workshop on natural language processing in the biomedical domain, Association for Computational Linguistics,* 2002

Collier N, Nobata C, Tsujii J. "Extracting the Names of Genes and Gene Products with a Hidden Markov Model." *Proceedings of COLING '2000* (2000): 201-07.

Craven M, Kumlien J. "Constructing Biological Knowledge Bases by Extracting Information from Text Sources." *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology* 1999: 77-86.

Gaizauskas R, Demetriou G, Artymiuk PJ, Willett P. "Protein Structures and Information Extraction from Biological Texts: The PASTA System." *Bioinformatics*. 19 (2003): 135-43.

Hatzivassiloglou V, Duboue P, Rzhetsky A. "Disambiguating Proteins, Genes, and RNA in Text: A Machine Learning Approach." *Bioinformatics* 2001: 97-106.

Hirschman L, Park J, Tsujii J, Wong L, Wu C. "Accomplishments and Challenges in Literature Data Mining for Biology," *Bioinformatics* 17 (2002):1553-61.

Hirschman L, Morgan A, Yeh A. "Rutabaga by Any Other Name: Extracting Biological Names**."** Accepted, *Journal of Biomedical Informatics,* Spring 2003.

Krauthammer M, Rzhetsky A, Morosov P, Friedman C. "Using BLAST for Identifying Gene and Protein Names in Journal Articles." *Gene* 259 (2000): 245-52.

Liu H, Friedman C. "Mining Terminological Knowledge in Large Biomedical Corpora." *Proceedings of the Pacific Symposium on Biocomputing.* 2003.

Palmer D, Burger J, and Ostendorf M. "Information Extraction from Broadcast News Speech Data." *Proceedings of the DARPA Broadcast News and Understanding Workshop*, 1999.

Pustejovsky J, Castaño J, Saurí R, Rumshisky A, Zhang J, Luo W. "Medstract: Creating Large-scale Information Servers for Biomedical Libraries." *Proceedings of the ACL 2002 Workshop on Natural Language Processing in the Biomedical Domain*. 2002.

Tauszig et al. "Toll-related receptors and the control of antimicrobial peptide expression in Drosophila." *Proceedings of the National Academy of Sciences* 97 (2000): 10520-5.

Yeh A., Hirschman L, Morgan A. "Evaluation of Text Data Mining for Database Curation: Lessons Learned from the KDD Challenge Cup." Accepted, *Intelligent Systems in Molecular Biology*, Brisbane, June 2003.