Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Biomedical Informatics 37 (2004) 396–410

Journal of Biomedical Informatics

www.elsevier.com/locate/yjbin

Gene name identification and normalization using a model organism database☆

Alexander A. Morgan[a,b,*], Lynette Hirschman[a], Marc Colosimo[a], Alexander S. Yeh[a], Jeff B. Colombe[a]

[a] *MITRE Corporation, 202 Burlington Road, Mail Stop K325, Bedford, MA 01730-1420, USA*
[b] *Department of Biology, Tufts University, Medford, MA 02155, USA*

Received 22 July 2004
Available online 8 October 2004

Abstract

Biology has now become an information science, and researchers are increasingly dependent on expert-curated biological databases to organize the findings from the published literature. We report here on a series of experiments related to the application of natural language processing to aid in the curation process for FlyBase. We focused on listing the normalized form of genes and gene products discussed in an article. We broke this into two steps: gene mention tagging in text, followed by normalization of gene names. For gene mention tagging, we adopted a statistical approach. To provide training data, we were able to reverse engineer the gene lists from the associated articles and abstracts, to generate text labeled (imperfectly) with gene mentions. We then evaluated the quality of the noisy training data (precision of 78%, recall 88%) and the quality of the HMM tagger output trained on this noisy data (precision 78%, recall 71%). In order to generate normalized gene lists, we explored two approaches. First, we explored simple pattern matching based on synonym lists to obtain a high recall/low precision system (recall 95%, precision 2%). Using a series of filters, we were able to improve precision to 50% with a recall of 72% (balanced *F*-measure of 0.59). Our second approach combined the HMM gene mention tagger with various filters to remove ambiguous mentions; this approach achieved an *F*-measure of 0.72 (precision 88%, recall 61%). These experiments indicate that the lexical resources provided by FlyBase are complete enough to achieve high recall on the gene list task, and that normalization requires accurate disambiguation; different strategies for tagging and normalization trade off recall for precision.

Keywords: Gene name finding; FlyBase; Named entity extraction; Text mining; Natural language processing; bioNLP

1. Introduction

The field of biology has undergone a profound revolution over the past 15 years, as an outgrowth of the successful sequencing of many organisms. Biology is now an information science, as well as a laboratory science; a major bottleneck for biology is the management of growing quantities of genomic and proteomic data reported in the biomedical literature.

The explosion of literature makes it nearly impossible for a working biologist to keep up with developments in one field, let alone with relevant work across organisms or on related genes or proteins. Biologists have responded by developing specialized databases to organize information. There are now hundreds of databases that are maintained and made accessible to the research community. These include organism-specific databases, such as the fly (Flybase [1]), mouse (MGI [2]), yeast (SGD [3]), rat [4], and worm [5] "model organism" databases, databases for proteins (PIR [6], SWISS-PROT [7]),

☆ This work has been funded in part by National Science Foundation Grant Number EIA-0326404.
* Corresponding author. Fax: +781 271 2780.
E-mail address: amorgan@mitre.org (A.A. Morgan).

1532-0464/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2004.08.010

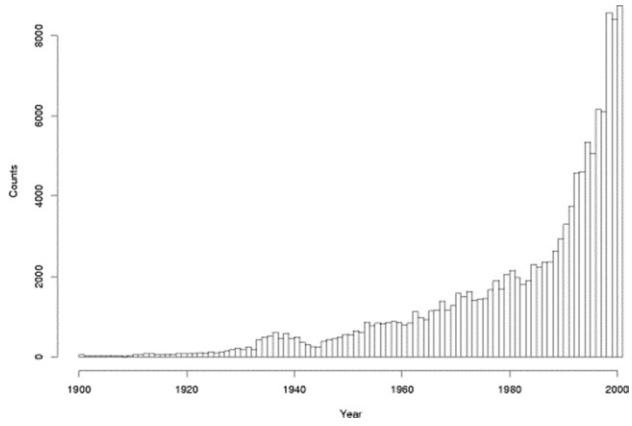

Fig. 1. FlyBase references, 1900–2000.

protein domains (Pfam [8], PROSITE [9]), protein–protein interactions (BIND [10]), pathogens (PRINTS Virulence Factor Database [11]), etc. Fig. 1 illustrates the exponential growth of literature references in the biological database FlyBase [1] organized by date of publication over a hundred year span.[1]

These databases have become critical to managing the flood of new information; at the same time, they are becoming increasingly expensive and difficult to maintain. Each model organism database, for example, is maintained by a team of specialized Ph.D. biologists (''curators'') who track the literature and transfer relevant new findings into appropriate database entries, in a process called ''data curation''. Often, these databases lag behind the literature because the curators have difficulty keeping up with the literature. The curators need interactive tools to help in the timely and consistent transfer of information from the literature into the databases. Curation typically involves a series of steps: identifying and prioritizing new articles to be curated; identifying the genes and/or proteins that have experimental findings associated with them; and, where possible, associating functional and expression information with each gene and protein. There has been growing interest in the application of entity extraction and text classification techniques to the problem of biological database curation [12].

The focus of our recent work has been to build text mining tools to aid curators in the identification and capture of information for biological databases. Our primary goal is to provide tools that can improve the currency, consistency, and completeness of biological databases.

A secondary goal has been to explore text mining strategies in a ''resource rich'' domain. Computer science researchers have had success in creating text mining and information extraction tools for other application domains, such as newswire – but these applications have required significant investment in infrastructure, such as lexical resources and hand annotated corpora. We wanted to explore the hypothesis that expert-curated biological databases provide sufficient resources for the creation of high quality text mining tools that can be applied to specific curation tasks.

Finally, a third goal has been to understand the complexities of the nomenclature problem for genes and gene products. There are three sub problems:

1. *Synonymy*. How to capture the different ways a single gene can be referred to in the literature (essential for searching the literature).
2. *Ambiguity*. How to know which gene a particular term refers to, since a given string, such as ''rhodopsin'', can refer to a number of different genes; this becomes particularly important if a word is ambiguous between ''regular language'' and a gene name or symbol (such as ''not'', the abbreviation of the gene *non-stop*). A recent study by Tuason et al. [13] showed that gene names are highly ambiguous, ranging from 2.4% to 32.9%.
3. *Normalization*. How to map from the name appearing in text to the underlying unique identifier (ultimately associated with a sequence of DNA on a specific chromosome).

We explore these issues in the remainder of the paper. The structure of the paper is as follows: Section 2 describes the applications of text mining to database curation, including a discussion of current approaches to entity extraction. Section 3 describes the resources that we created for running our experiments on extraction and gene list normalization. Section 4 describes a series of three experiments on gene extraction and the automatic generation of normalized gene lists. Section 5 concludes with discussion of the results and directions for future research.

2. Applying text mining to biological database curation

This section describes the curation process and possible applications for text mining, including entity tagging and the generation of normalized gene lists. We then review approaches to entity tagging across a range of applications (newswire in multiple languages and biological applications).

2.1. The curation process

The curation process [14] can be divided into a handful of steps, starting with a selection process for papers

[1] FlyBase is a database that focuses on research in the genetics and molecular biology of the fruit fly (*Drosophila melanogaster*), a model organism for genetics research. Of course, most of the early references in FlyBase are not in electronic form.

about a particular entity of interest. For example, curators may be interested in a particular organism or a particular gene. A variety of information retrieval and text mining techniques can be used for this step; the KDD Challenge Cup 2002 [15] focused on this task for FlyBase curation of gene expression data. One of the problems at this stage is that gene names can be highly ambiguous, especially across organisms [13], and the retrieval of relevant documents could be improved by disambiguating the names of genes of interest.

For FlyBase,[2] as articles are selected, there is an initial phase of annotation that generates the list of genes (in the form of FlyBase gene identifiers) that have "curatable information" in the article. These normalized lists of genes are the focus of our current work.

The next step is for the curator to read the full text of the selected papers and enter the relevant information into the database for the listed genes, citing the source document. This is the most time consuming step in curation, and our long-term goal is to speed up this process by nominating database entries for further manual examination. This would assist the curators with prioritization and would offload many bookkeeping tasks onto the automated system.

There are already several automated systems in use. The BIND [10] database uses PreBIND, a support vector machine classifier, to extract protein–protein interactions based on text features. It provides a 70% savings in task duration [16]. The Textpresso [17] system uses an ontology and extensive lexical tools to provide improved search and retrieval for WormBase.

2.2. Entity tagging: current approaches

Entity tagging is a foundational step for more sophisticated extraction. This has been an area of active research for computer scientists and computational linguists for the past 20 years. There are two approaches in general use for entity extraction. The first requires manual or heuristic creation of rules to identify the names mentioned (occurrences) 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. For negative examples, 'closed world' assumption generally is taken to apply: if an entity is not tagged, it is assumed to be a negative example.

Fig. 2 shows results from the IdentiFinder system for English and Spanish newswire [18], 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.

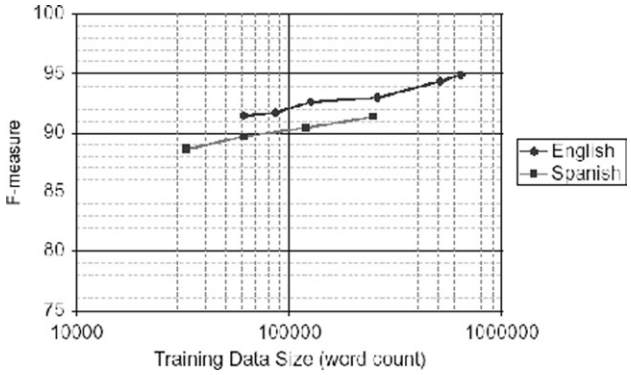

Fig. 2. Performance of BBN's IdentiFinder named entity recognition system relative to the amount of training data, from [Bikel99].

2.3. Entity tagging and extraction systems in biology

Overall, hand-crafted systems have tended to outperform learning-based systems for biology. However, it is clear that the quantities of training data have been small, relative to the results reported for entity extraction in, e.g., newswire [19]. There are several published sets of performance results for automatic named biological entity extraction systems.

One of the earliest results reported in biological named entity tagging were from Fukuda et al. [20] which gave a precision of 0.95 and a recall of 0.99, using a heuristic system, on a very small data set focused on SH3 (Src homology 3) domain proteins. These results are based on a relaxation of their original, strict evaluation rules so that cell names and phage names were considered acceptable when reporting these final numbers. Franzén et al. [21] compared their heuristically based system, Yapex, to the freely available KeX[3] system based on the Fukuda heuristics (PROPER – PROtein proper noun Extraction Rules) and reported an *F*-measure of 0.83 for both systems using a "sloppy" metric on their evaluation set.[4] Interestingly, when using a "strict" boundary metric requiring exact matching of the tagged term boundaries to the goldstandard (similar to the one we describe in Section 4.2.1), the Yapex system had an *F*-measure of 0.67 and the KeX system dropped to 0.41.

The system of Collier et al. [22] used a hidden Markov model to achieve a balanced *F*-measure[5] of 0.73

[2] Special thanks to William Gelbart, David Emmert, Beverly Matthews, Leyla Bayraktaroglu, and Don Gilbert for their expert assistance in understanding and accessing FlyBase.

[3] http://www.hgc.ims.u-tokyo.ac.jp/service/tooldoc/KeX/intro.html.

[4] http://www.sics.se/humle/projects/prothalt.

[5] Balanced *F*-measure is the harmonic mean of precision and recall, weighted equally: $F = (2 * \text{precision} * \text{recall})/(\text{precision} + \text{recall})$.

when trained on a corpus of 29,940 words of text from 100 MEDLINE abstracts. Contrast this with results on newswire in Fig. 2, which achieved a balanced *F*-measure of 0.95 for English entity extraction, when trained on over 600,000 words of training data; *F*-measure for Spanish was 0.91 on 250,000 words of training data.

Krauthammer et al. [23] took a somewhat different approach that encoded characters as 4-tuples of DNA bases; they then used BLAST together with a lexicon of gene names to search for 'gene name homologies'. They reported an *F*-measure of 0.75 without the use for a large set of rules or annotated training data.

A hybrid system, Abgene, developed by Lorraine Tanabe and John Wilbur [24] trained the Brill tagger[6] to tag gene mentions as an additional part of speech, using a tagged corpus of 10,000 sentences[7], and then a series of heuristically derived rules to correct errors in the tagging. Rather than reporting a single set of results, they showed what their results would be at a variety of sentence score thresholds. The best *F*-measure for a subset of the sentences reported in this way was 0.89.

Another hybrid system, PASTA [25], used a combination of heuristic and machine-learned rules to achieve an *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).

This contrasts with the results described by Kazama et al. [26] which compared support vector machines to maximum entropy methods to automatically identify and assign terms to one of the 24 categories from the GENIA ontology using the GENIA corpus.[8] The overall results for the SVM's and the max-entropy method were comparable with an *F*-measure of 0.73 for finding named entities in general, but only 0.54 for both correctly identifying and classifying the terms.

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 [27] 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 algorithm, they identified single sentences which mentioned both the protein and its localization. They then matched these sentences 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. Resources

As noted earlier, much of our effort has been focused on how to make the most of the rich information in biological databases. We describe the resources available in FlyBase and the use of abstracts from MEDLINE. We then provide a detailed example of a FlyBase entry (gene list and abstract) and discuss the training and test corpora that we used for our experiments.

3.1. FlyBase

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 (DNA sequence locus). The definition of what constitutes a gene is complex,[9] and unfortunately, the naming of genes is also not straightforward. For example, proteins are often described in terms of their function; this description then becomes used as the name of the protein; and the protein name, in turn, is used to refer to the gene that codes for that protein. For example, *suppressor of sable* (FBgn0003575) is the name of a gene which suppresses expression of another gene *sable* (FBgn0003309), defects of which cause visible changes in color and darkening of the fly. This name can then be abbreviated as *su(s)* or *su-s* (these are just two of five synonyms).

[6] http://www.cs.jhu.edu/~brill/RBT1_14.tar.Z.

[7] Tanabe and Wilbur have made this corpus available as training data for the BioCreAtIvE evaluation Task 1A, described in Section 3.4. http://www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/.

[8] http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/.

[9] There can be considerable confusion in defining a gene. A chromosomal sequence which is translated into RNA and then transcribed into a polypeptide is the basic definition, but there can be many repeats of the same sequence on different chromosomes which can be identical or functionally identical (code for the same protein). Particular chromosomal locations or sequences of interest (e.g., portions of the centromere) have unique identifiers in the database, as do pieces of mitochondrial DNA that are not part of the chromosomes. There are many other complexities, such as the multitude of sequence locations which look like polypeptide coding regions based on sequence analysis but which have not been linked to known proteins. However, fortunately, we can take the set FlyBase gene identifiers as a given, so for our purposes, it is not necessary to worry about the exact definition of a gene.

If there were a one-to-one relationship between gene name and unique identifier, the gene identification task would be straightforward. However, both ambiguity 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 that often overlap with English words. For example, *18 wheeler*, *batman*, and *rutabaga* are all Drosophila gene names. Likewise, the word "not" is the symbol for the *non-stop* gene (FBgn0013717). In some cases, a single entity (as represented by a unique identifier) may have a number of names, such as *ATPα*, which has 38 synonyms listed in FlyBase, or the gene *toll* with 14 synonyms (*toll* is also listed as a synonym for *18 wheeler*).

FlyBase provides synonym lists for each gene, along with the gene symbol and its unique identifier in FlyBase. Using these synonym lists, we 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 a unique FlyBase gene identifier.

Fig. 3 shows a part of the FlyBase entry for the gene *Ultrabithorax* (FBgn0003944). Under the headings **Molecular Function** and **Biological Process**, we see that this gene is responsible for encoding a DNA binding and transcription regulating protein. We see further that *Ubx* is synonymous for *Ultrabithorax* and is the short form of the name (top of the entry next to **Symbol**). The link **Synonyms** leads to a long synonym list which includes *DmUbx*, *bithorax*, *DUbx*, *Ultraabdominal*, *Ubx1b*, bx^{D}, bxd^{D}, *bxl*, *l(3)89Eb*, *bx*, *pbx*, *Cbx*, *abx*, *Haltere mimic*, *Hm*, *postbithorax*, *Contrabithorax*, *anterobithorax* and *bithoraxoid*.

Most of the recorded facts about *Ultrabithorax* are linked to a particular literature reference in the database. For example, following the link for **Attributed Data** (not shown on this screen shot) leads to a page linked to the abstract of a paper by Larsen [28], which reports on the phenotypic effect of modifications to the gene. Fig. 4 shows this abstract and the normalized list of entities for that paper, including *Ultrabithorax* (*Ubx*) and its allele $Ubx^{bx\text{-}1}$.

3.2. MEDLINE abstracts

In order to build and evaluate our text mining systems, we needed data, specifically text associated with gene lists. Using the BioPython [29] modules, we were able to obtain MEDLINE abstracts for 22,739 of the

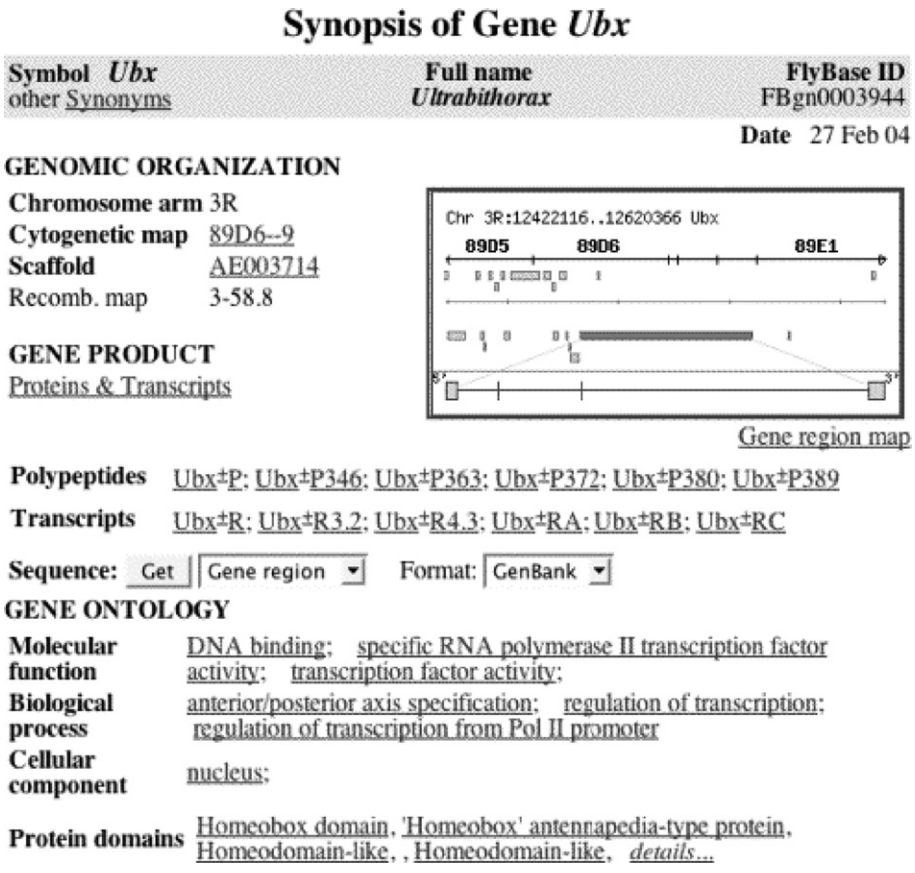

Fig. 3. FlyBase entry for *Ultrabithorax*.

A

Abstract from PubMed

Genetic analysis of modifiers affecting sexual dimorphism and temperature sensitivity of bx1 in Drosophila melanogaster.

Larsen E.

Department of Zoology, University of Toronto, Ontario, Canada.

Two modifiers of bithorax1 phenotypic expression are described. An X-chromosome region is associated with sexual dimorphism in bx1 penetrance. It is hypothesized that sexual dimorphism is in part due to a lack of dosage compensation of the modifier, in males. A third chromosome region that segregates with the pink peach allele is implicated in mediating temperature sensitivity. By appropriate combinations of modifiers, both sexual dimorphism and temperature sensitivity can be greatly reduced.

B

Symbol	Class	ID
Ubx^{bx-1}	Allele	FBal0017516
Ubx	Gene	FBgn0003944

Fig. 4. Abstract for Larsen, 1989 and associated gene list from FlyBase.

papers referenced in FlyBase. Of these, 15,144 had gene lists (the absence of a gene list typically indicated that the paper dealt with fly molecular biology rather than genetics). We excluded the 1111 articles used to make up the blind test set for the KDD Cup Challenge [15] so that we would not inadvertently "contaminate" this data set; this left a pool of 14,033 abstracts for our experiments.

We used journal abstracts because of their availability through MEDLINE. From our earlier work, we know 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 [19]. For example in one sample, only about 25% of the genes listed on the gene list appeared in the abstract.[10] However, due to copyright restrictions, there is a paucity of freely available full text for journal articles. The articles that are available in electronic form vary in their formatting, which can cause difficulty in automatic processing. By contrast, MEDLINE abstracts have a uniform format and are readily available. Many other experiments have been performed on MEDLINE abstracts for similar reasons. In the long term, however, it will be critical to develop tools that can work on full text articles, since this is where the full spectrum of interesting information resides.

3.3. Evaluating gene name mentions

We divided our experiments into two phases, separating the tagging phase from the normalization phase. To evaluate the gene name tagging, we created a small doubly annotated test corpus. We selected a sample of 86 abstracts and had two annotators mark these abstracts for *D. melanogaster* gene name mentions. 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).[11] 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 *D. melanogaster* 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).

One annotator was a professional researcher in biology with experience as a model organism genome database curator (Colosimo). This set of annotations was

[10] These results were derived from the blind test data for BioCreAtIvE Task 1B for fly, described in Section 3.4. We determined that of 1516 genes listed for 250 FlyBase abstracts, only 368 (24%) were found in the abstracts. See http://www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/.

[11] In our manual annotation, we did create separate tags for complexes and families, since we believe that these will be important for future tasks.

Table 1
Training data quality and inter-annotator agreement for 86 gene mention tagged abstracts

	F-measure	Precision	Recall
Training data quality	0.83	0.78	0.88
Inter-annotator agreement	0.87	0.83	0.91

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 by comparing the second set of annotations to the gold standard; these showed agreement that gave a balanced *F*-measure of 0.87 (see Table 1). Fig. 5 shows the previously mentioned Larsen, 1989 article annotated for the gene mention task. The gene mentions are bracketed by XML tags "⟨gn⟩" and "⟨/gn⟩" (in bold).

3.4. BioCreAtIvE evaluation datasets

The second set of experiments focused on gene list generation, which required a separate set of annotated data. For this, we drew on the FlyBase data sets that we were preparing for BioCreAtIvE (Critical Assessment of Information Extraction in Biology). BioCreAtIvE is a CASP-like evaluation [30] for systems working in text mining for biomedical literature, funded by EMBO and NSF. It is intended to provide a common evaluation corpus, evaluation metrics, and a forum to discuss results of natural language processing tasks applied to biomedical literature. In this way it is similar to the TREC [31] and MUC [32] evaluations. It consists of two gene extraction tasks (described below) and a functional annotation task, prepared by Christian Blaschke and Alfonso Valencia (Protein Design Group, Centro Nacional de Biotecnologia, Autonomous University, Madrid) in collaboration with Rolf Apweiler and Evelyn Camon (SWISS-PROT).

We were responsible for the development of BioCreAtIvE Task 1B. This track focused on the normalized gene list task using data from three model organism databases (mouse, fly, and yeast). For each organism, the task was to generate the normalized list of organism-specific genes mentioned in abstract (since we did not have the full text articles available). Task 1B was structured as an entity identification and normalization task. Participants were evaluated on the list of unique identifiers, and not on tagging gene names in running text, although another BioCreAtIvE task, Task 1A, using annotated data provided by Lorraine Tanabe and John Wilbur at NCBI, focused on tagging individual mentions in sentences drawn from MEDLINE abstracts. We used the Task 1B fly corpus and scoring script to evaluate our ability to normalize gene names, i.e., to link mentions to unique identifiers.

The BioCreAtIvE task 1B fly corpus was drawn from references not used in the KDD Cup Challenge [15], consisting of 5000 abstracts with noisy annotation (described in Section 4.1.2 below) for training data. Another 1000 were set aside for hand annotation; of these, 108 abstracts were annotated (Colombe, Morgan) for development testing and 250 were annotated (Colombe, Morgan, Colosimo) to create the evaluation gold standard. This also left a set of 8033 abstracts not used as part of the KDD Cup Challenge and not part of BioCreAtIvE. We used both the BioCreAtIvE training data and this additional data set to train our tagger in the experiments described in Section 4.

The annotations consisted solely of a list of the unique identifiers of the *Drosophila melangaster* genes mentioned in each abstract, as described in the annotation guidelines for BioCreAtIvE Task 1B [33]. A mention had to be specific to a *D. melanogaster* gene, not just to a related species (e.g., *Drosophila virilis*), even if the gene was of the same name. When no explicit fly species was mentioned, *D. melanogaster* was assumed. This was a valid assumption for two reasons. First, *D. melanogaster* is far more studied than other species, and second, when general properties of a gene shared are described, the *D. melanogaster* gene is implicitly included. A mention also had to be uniquely associated to a single gene or be part of an enumerated set. For example, the text "G2-specific (CLB1-CLB4) cyclins" would map to the identifiers for *CLB1*, *CLB2*, *CLB3* and *CLB4*, whereas a mention of *actin* would not constitute a mention of a specific gene, because there are six different genes that code for actin in the fly, and it is not clear which one of them is meant.

```
<DOC pubmedid="2499435">
<TXT>
Two modifiers of <gn>bithorax1</gn> phenotypic expression are described.
An X-chromosome region is associated with sexual dimorphism in
<gn>bx1</gn> penetrance. It is hypothesized that sexual dimorphism is in
part due to a lack of dosage compensation of the modifier, in males. A
third chromosome region that segregates with the <gn>pink peach</gn>
allele is implicated in mediating temperature sensitivity. By appropriate
combinations of modifiers, both sexual dimorphism and temperature
sensitivity can be greatly reduced.
</TXT>
</DOC>
```

Fig. 5. Larsen, 1989 annotated for gene mentions.

Abstract	FlyBaseGeneID	AutoFound	HandFound
fly_00004_devtest	FBgn0003944	N	Y
fly_00004_devtest	FBgn0003029	X	Y

Fig. 6. Gene list annotations for Larsen, 1989 (fly_00004_devtest.gene_list).

```
<DBGENELIST>
<DBGENE DBID="FBgn0003944" AF="N" HF="Y"/><!--Ubx, Ultrabithorax, CG10388, bx: bithorax,
Cbx: Contrabithorax, abx: anterobithorax, pbx: postbithorax, UBx, Hm: Haltere mimic, bxd,
ubx, DmUbx, bithorax, bx&lt;up&gt;D&lt;/up&gt;, bxd&lt;up&gt;D&lt;/up&gt;, bxl, l(3)89Eb,
bx, pbx, Cbx, abx, Haltere mimic, Hm, postbithorax, Contrabithorax, anterobithorax, bitho-
raxoid-->
<DBGENE DBID="FBgn0003029" AF="X" HF="Y"/><!--pink peach-->
</DBGENELIST>
```

Fig. 7. Gene list annotations used by curators for Larsen, 1989 (fly_00004_devtest.gene_list).

The annotations for the Larsen, 1989 abstract (referenced as `fly_00004_devtest` in the BioCreAtIvE corpus) are shown in Fig. 6. There are four columns in the training data. The first column identifies the abstract (using BioCreAtIvE internal identifiers). The second column contains the gene identifiers for that article from FlyBase. The final two columns are used in the preparation of the training data to keep track of genes automatically found in the abstract (**AutoFound**) and any additional corrections done by hand (**HandFound**). To create the **AutoFound** entry, a pattern matching program looks for mentions of the gene and its synonyms in the abstract.[12] If the gene mention is found, a "Y" is entered in the **AutoFound** column, otherwise a "N" is entered. Then these entries are manually checked (the **HandFound** column). In the case where a gene is found that is not on the gene list, a new entry is created, with an "X" in the **AutoFound** column, and a "Y" in the **HandFound** column. In the Larsen, 1989 article, this occurred for FBgn0003029, *pink peach*, mentioned in the fourth sentence of the abstract ("A third chromosome region that segregates with the pink peach allele...") in Fig. 5. To facilitate in the correction process, the annotators were provided with an enhanced version of the lists shown in Fig. 7. These included the full synonym lists for each gene, contained in the comment field enclosed by "<!–" and "–>".

[12] It was necessary to edit the gene list because we were using abstracts rather than full text. We therefore knew that there would be many genes on the list that did not appear in the abstract. However, we also made a modification to the gene list task, to include any organism-specific gene mentioned in the abstract, even when the gene did not appear on the gene list. Since different databases use different criteria for the inclusion or exclusion of genes from the gene list, this served to simplify the task for the participants (although it complicated the hand correction process for the development training and test materials considerably).

4. Gene extraction and normalization experiments

We performed a series of three experiments related to identifying and normalizing gene mentions in FlyBase abstracts. The first experiment used lexicon-based pattern matching to generate a high recall, low precision gene list (since many extraneous genes were included in this process). In the second experiment we ignored normalization and used pattern matching to create a large set of abstracts, imperfectly tagged with all gene mentions; this was then used to train a statistical gene mention tagger. In the third experiment, we combined this tagger with some very simple disambiguation approaches to return a list of normalized genes. These experiments allowed us to explore different precision-recall trade-offs for the gene normalization problem.

4.1. Experiment A: using the lexicon to create a normalized list of fly gene mentions

The rich lexical resources available in FlyBase formed the basis for a simple experiment in gene normalization. This approach used lexical-based pattern matching to identify genes and to associate the genes with the unique gene identifiers. We knew, from earlier work, that we would "overgenerate", since many gene names are ambiguous either with English words ("not") or among alternate genes ("Clock"), or both.

Previously, we had looked at different matching schemes to reproduce the gene list associated with paper references in FlyBase on both abstracts and the available full text articles [19]. The results were predictably poor: we reported a recall of 84% and precision of 2% on the full papers; and recall of 31% with 7% precision on abstracts, due to the impoverished information content of the abstracts. However, we decided to redo this experiment using the BioCreAtIvE Task 1B development test set, because this data set had been hand corrected to correspond to the genes mentioned in each abstract, rather than just the list of genes curated for the full text.

Table 2
Lexicon-based gene name tagging and normalization, with multiple filtering methods

	Method	True positive	False positive	False negative	Precision	Recall	Balanced *F*-measure
1	**Full lexicon**	**200**	**6785**	**11**	**2.9**	**95**	**5.6**
2	No common English	196	2214	15	8	93	15
3	>2 Characters only	195	1204	16	14	92	24
4	No ambiguous	161	952	50	15	76	25
5	>3 Characters only	173	536	38	24	82	37
6	No common English >2 characters	187	355	24	35	89	50
7	No common English, no ambiguous, >2 characters	151	154	60	50	72	59

For the evaluation in the normalization experiments, the system provides a list of unique identifiers and an excerpted mention for a given document, and that list is compared to the gold standard. The evaluation program[13] returns true positive, false negative and false positive counts, along with precision, recall and balanced *F*-measure scores. The actual excerpts are not evaluated automatically, but are used for analysis and comparison.

A major advantage of the normalized gene list evaluation is that evaluation is very simple: the system's list of unique identifiers is compared against the list in the gold standard. This allows us to ignore issues of mappings to normalized forms, tag boundaries and variant tokenization. The gene list task differs from the gene mention task in that it weights a gene mentioned once equally with a gene mentioned many times in a given abstract, favoring systems with high sensitivity (recall) for infrequently appearing genes.

4.1.1. Methodology

We first compiled a series of regular expressions for all the terms in the lexicon, linking them with a list of the associated unique identifiers. We treated all white space and hyphens inside of the gene term as equivalent, ignored case differences, and matched at any non-alphanumeric at the right and left of the term. For example, the term "suppressor of sable" would be compiled in Python, using the **re** regular expression module, as:

```
regexterm = re.compile(r"\Wsuppressor[\-\s]
                of[\-\]sable\W",re.I)
```

For each regular expression used, we did a search for any matches, and if a match occurred, the unique identifiers associated with that term would be added to the gene list for the abstract.

4.1.2. Analysis

We used the 108 abstracts in the BioCreAtIvE development test set and compared different filtering schemes to reduce the large number of spurious matches. The results are shown in Table 2. The baseline system (Full Lexicon, line 1, Table 2) with no filtering matched all terms and compiled a list of the unique identifiers for all those matches. We then explored a number of filters to improve precision by removing false positives. For identifying and filtering out common English words, we used a list of the 5000 most frequent terms in the Brown corpus [34] (lines 2, 6, 7 in Table 2). We also tried excluding any terms which were ambiguous (polysemous gene names—shown as "No Ambiguous" in lines 4 and 7, Table 2), and removing all very short identifiers (3 characters and smaller—line 5, Table 2; 2 characters and smaller, lines 3 and 7, Table 2). We also tried combining these filtering techniques, for example, excluding common English words and words 2 characters and smaller (line 6, Table 2) and excluding all ambiguous terms, common English words and all terms 2 characters or less (line 7, Table 2).

Out of the 108 abstracts, there were 211 annotations (unique identifiers associated with an abstract). Perhaps the most interesting row is the top one ("Full Lexicon"), with the lowest *F*-measure. Using all possible matches, 95% of all gene names were found, setting an upper bound for recall. This gives us a measure of the completeness of the lexicon in FlyBase. Some of the mentions that were missed were due to ellipsis under conjunction in the phrase "host yolk polypeptide (1 and 2)" which corresponds to *yolk protein 1* (*yp1*) and *yolk protein 2* (*yp2*); in a similar case, "JPDs" was used as a plural to discuss differences in *Drosophila* and mouse versions of *jdp*, and was missed. A description of the *flare* (*flr*) gene in one of its allelic forms, "flr3" was missed. Some substitutions of similar words also caused problems in description-like names. For example, the phrase "insulin-like peptide" was missed, because the lexicon had "insulin related peptide", and "93D region" was missed as a mention of *Hsrω* that has "93D locus" in its synonym list. What looks like an interesting transcription error in the synonym list was found in our example abstract, Larsen 1989. The synonym list for *Ubx* has "bxl" in it (third character is an ell), whereas the text had "bx1" (third character is the numeral one, which appears to be the correct form; a formatted form of the abstract text actually had a superscript, "bx^{1}").

The low levels of precision for the "Full Lexicon" run can be attributed to the extensive ambiguity of gene

[13] Available for download at http://www.mitre.org/public/biocreative/task1Bscorer.pl.

names. We identified three types of ambiguity in Drosophila gene names. 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 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.[14] The most problematic type of ambiguity 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*, *to*, *a*, *not* and even *and* each occur as gene names. For example, the gene symbol (short form) of the gene *takeout* is *to*, and the symbol for the gene *wee* is *we*. Tuason et al. [13] report that they found a 2.4% ambiguity for all FlyBase names with their English dictionary. This was the motivation for filtering out common English words (**No Common English** lines in Table 2). It may be that more sophisticated handling of abbreviations can address some of these issues.

Interestingly, the best performance with respect to *F*-measure (0.59) and precision (50%), but with the lowest recall (72%) was achieved by a union of the filtering techniques, seen in the last line of Table 2: **No Common, No Ambiguous**, **>2 Characters**. However, even these rather draconian filters achieved only 50% precision—clearly better disambiguation (and more selective tagging of genes) is required to produce acceptable performance.

4.2. Experiment B: machine learning to find gene names using noisy training data

Our initial experiment demonstrated that exact match using rich but highly ambiguous lexical resources was not useful on its own. We realized, however, that to develop training data, 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. The data appears to be too sparse to learn a separate classifier for each gene, but we decided to separate the problems of tagging and normalization. By first focusing on gene name tagging, we hoped to increase precision by learning statistics based on local context and actual usage that would outperform the simple pattern matching approach.

We did not have the time or resources to tag and normalize a large test set of abstracts with all gene mentions, so we used our small set of 86 abstracts hand-tagged for gene mentions. This allowed us to measure system performance for tagging gene names in running text.

4.2.1. Methodology

To create the noisy training data, we needed to tokenize the texts (divide the text into a sequence of words), and then perform longest-first pattern matching (to create the tags), based on looking for synonyms of all "licensed" genes for each abstract.

To divide the text of abstracts into words, we used the MITRE 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 [36] for an interesting discussion of tokenization issues, and [37] for a protein name finding system which avoids tokenization entirely. 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. We processed the 14,033 abstracts not used for the KDD Cup Challenge to generate a large quantity of noisy training data.

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 many 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. To evaluate how noisy the training data were, we used our licensed gene pattern matching technique to tag the 86 abstract evaluation set. We then evaluated the automatically tagged data against the hand-annotated gold standard. The results, (Table 1, row 2), showed a recall of 88%, precision of 78% and *F*-measure of 0.83. This was lower than the 0.87 *F*-measure for inter-annotator agreement, but we believed it would be good enough to train a statistical tagger.

Fig. 8 shows our noisy training data methodology. We chose the HMM-based trainable entity tagger ***phrag***[15] [38] to extract the names in text. We trained

[14] Hazivassloglou [35] 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. Fortunately, the FlyBase gene list focuses on both genes and gene products, so it was not important to make this distinction in this application.

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

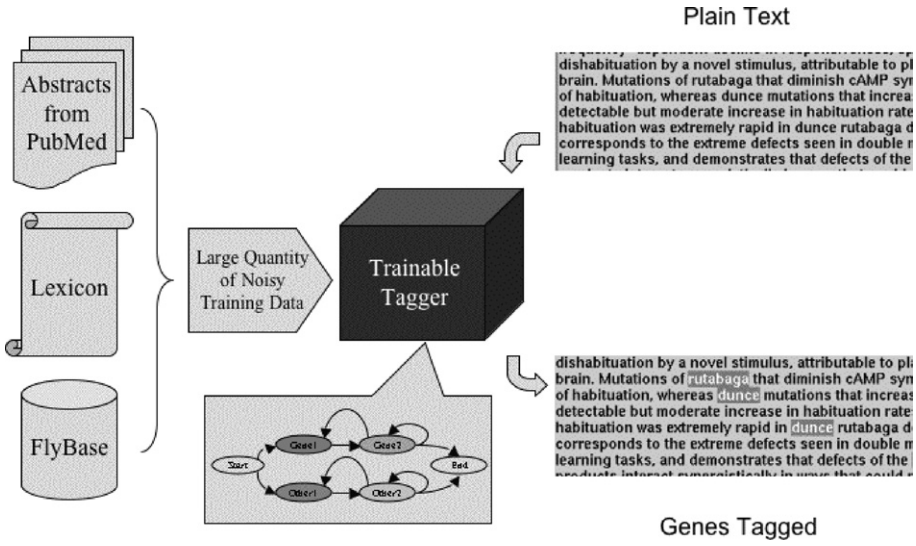

Fig. 8. Schematic of methodology.

phrag on different amounts of noisy training data and measured performance. Our evaluation metric was the standard metric used in named entity 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 was tagged for "fas" but not for "receptor" would constitute a miss (recall error) and a false alarm (precision error).

4.2.2. Analysis

Table 3a shows the performance of the basic system as a function of the amount of training data. As in Fig. 2, performance improved with the amount training data. At 2.6 million words of training data, ***phrag*** achieved an entity identification *F*-measure of 0.73. 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 3b, to an *F*-measure of 0.75.

Table 3a
Performance as a function of training data

No orthographic correction			
Training data	*F*-measure	Precision	Recall
531,522	0.62	0.73	0.54
529,760	0.64	0.75	0.56
1,342,039	0.72	0.80	0.65
2,664,324	0.73	0.79	0.67

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

Orthographic correction			
Training data	*F*-measure	Precision	Recall
531,522	0.65	0.76	0.56
529,760	0.66	0.74	0.59
522,825	0.67	0.76	0.59
1,322,285	0.72	0.77	0.67
1,342,039	0.75	0.80	0.70
2,664,324	0.75	0.78	0.71

Fig. 9 shows these results in a graphical form. Two things are apparent from this graph. Based on the results in Fig. 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 0.88. The other result which stands out is that there is considerable variation in the performance when

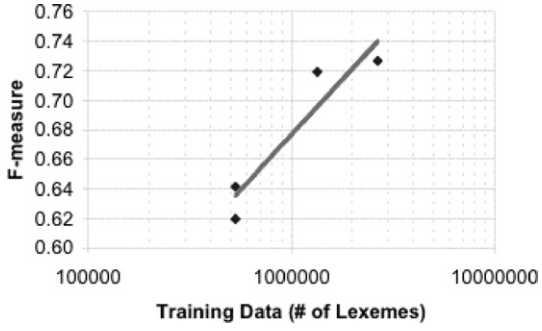

Fig. 9. Performance as a function of the amount of training data.

trained on different training sets of the same size. We believe that this is due to the very limited amount of testing data.

An error analysis of 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 tagging some, but not all, of the occurrences of term. This suggests an opportunity to use document level context for some sort of post processing step: e.g., if the term "rutabaga" is identified in one place in an article, this provides strong evidence that all occurrences of "rutabaga" should be tagged in the article.

The pattern matching process might also be improved by using a morphological analyzer trained for biological text. This would eliminate some of the tokenization errors and perhaps capture underlying regularities, such as addition of Greek letters or numbers (with or without preceding hyphen) to specify sub-types within a gene family. There is also 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" as a synonym of the *Toll* gene [39]. Gene names can also be explicit descriptions of their chromosomal location or even function (e.g., *Dopamine receptor*). Of course this type of analysis needs to be extended to other organisms with biological databases, as in the work of Tuason et al. [13].

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 somewhat encouraging, compared with the 3.3% precision and 4.4% recall for novel gene names reported by Krauthammer et al. [23]. 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. However, for the gene list task, this was not relevant, since we were dealing with previously curated articles, where all gene names had already been entered into the lexicon.

4.3. Experiment C: using the HMM based tagger and lexicon to create a normalized list of gene mentions

Our final experiment was to combine the statistical tagger with a normalization procedure, to generate normalized gene lists using the BioCreAtIvE Task 1B data. The results were evaluated as described in Section 4.1.

4.3.1. Methodology

We trained ***phrag***, our basic HMM-based named entity tagger, on "noisily tagged" versions of the 5000 abstracts that were the BioCreAtIvE training set, and then added the remaining 8033 abstracts to train ***phrag*** as previously described. Then we simply used our lexicon of FlyBase synonyms to look up the mentions tagged by ***phrag***. Due to the polysemy in the synonym lists, this lookup can often yield multiple unique identifiers for a given name, so we examined two methods of dealing with this. The maximal version retrieved all unique identifiers possible, whereas the minimal case ignored any tagged mention that was ambiguous (mapped itself to multiple unique identifiers).

4.3.2. Analysis

The performance of the lookup on the BioCreAtIvE Task 1B fly development test set is shown in Table 4. We focus here on the results using the larger set of training data. The maximal version (line 3, Table 4) had a recall of 77%, (slightly better than the recall level of our named entity tagger), and a precision of 53%. The minimal version that excluded ambiguous forms (line 4) had a relatively high precision of 0.88, with a recall of 61% and *F*-measure of 0.72. The many false positives in the maximal version (147) were almost entirely due to polysemous gene names. Perhaps more interestingly, there were 17 false positives reported in the minimal case; 7 of these were linked to mentions of identically named genes in related species (e.g., *Drosophila lebanonensis*, *Drosophila simulans*, *Drosophila yakuba* and in one case, yeast).

There were also a few examples of misinterpreted acronyms. The term "FOUT" was tagged, and rather than being a reference to the gene *fade out*, it was a reference to one of a pair of promoter regions for the *F-element*,

Table 4
Performance of normalization using lexical lookup of tagged mentions

	Training set	True positive	False positive	False negative	Precision	Recall	Balanced *F*-measure
Maximal	BioCreAtIve	158	152	53	51	75	61
Minimal (no ambiguous)	BioCreAtIve	123	17	88	88	58	70
Maximal	BioCreAtIve + 8033	163	147	48	53	77	63
Minimal (no ambiguous)	BioCreAtIve + 8033	128	17	83	88	61	72

along with "FIN". "RACE analysis" was used in an experimental process, and confused with a gene that had "RACE" as a synonym. In disambiguation, we noticed problems with misinterpreted abbreviations. The problem of matching abbreviations has been tackled by a number of researchers; see Liu and Friedman [40] and Pustejovsky et al. [41]. It may be that ambiguity for "short forms" of gene names could be partially resolved by detecting local definitions for abbreviations.

Ambiguous terms that could be both English words and gene names caused problems. The mistagging of "brown" was the most obvious example, but also a discussion of a transfected gene taken from a castor bean was confused with a mention of the gene *castor* (*Cas*). Finally, "Type I" and "Type II" occur on the synonym lists for the ribosomal elements *R1* and *R2*, but the text was describing "Type I and Type II DNA-topoisomerases". It might be possible to apply part of speech tagging, noun phrase chunking and corpus statistics to avoid mis-tagging these common words.

The results of these experiments are shown as a scatter plot of precision vs. recall in Fig. 10. The 7 methods listed in Table 2 are plotted, along with the results from Table 4. The graph shows a variety of different trade-offs between precision and recall. Our initial experiments used only very simple kinds of filtering, such as removing ambiguous terms, removing short words or removing terms that also occurred in English. We clearly need more sophisticated approaches to disambiguation. It will be interesting to revisit these results in light of the BioCreAtIvE evaluation results that are now available along with datasets [42,43].

5. Discussion

Our goals in this work were to create text mining tools useful to biologists, particularly to database curators; to explore text mining strategies in the resource-rich domain of biological databases; and to understand the dimensions of the problems in identifying and normalizing gene mentions in biological text.

For text mining, our major accomplishment has been to demonstrate 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 described in Section 2. While these results are still considerably below the results from Gaizauskas et al. [25], we believe we can improve both the quality of the training data (through better tokenization and pattern matching) and the performance of the tagger (through the use of non-local context and morphological features).

We also demonstrated that it can be productive to separate tagging from normalization. In one experiment, simple pattern matching produced a system with recall of 95% in the gene list task. This provides a good measure of the completeness of the lexicon for FlyBase. The missing cases (from our very small test sample) seem to involve degenerate forms due to conjunctions or plurals, paraphrases of names, or to persistent ambiguity between gene names and other technical terms, including gene names from other species.

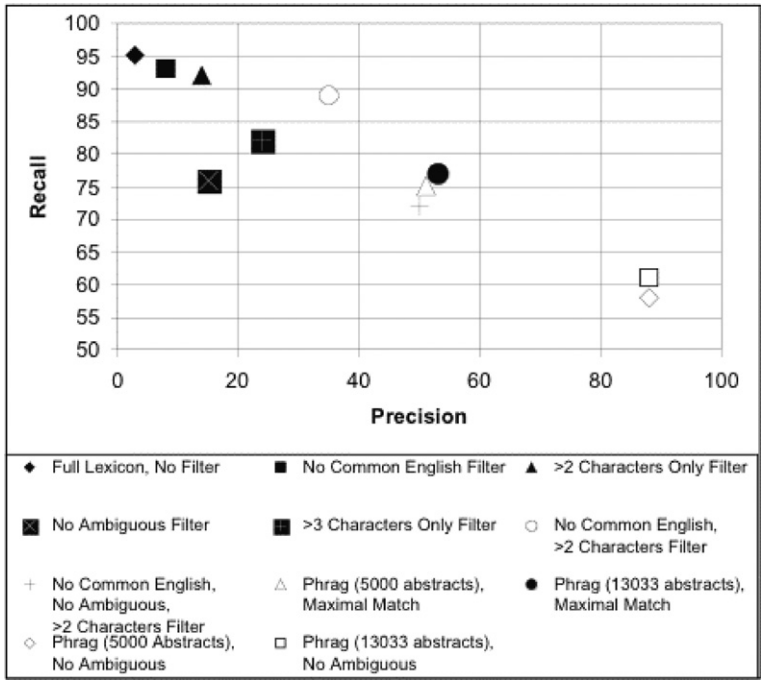

Fig. 10. Precision and Recall for different methods of normalized gene list generation.

Our best *F*-measure for the gene list task (0.72) was achieved using a statistical tagger, coupled with removal of all ambiguous terms (relying on redundant mentions of terms to preserve recall). In this experiment, we believe that recall for the gene list results (0.77 at best) may have been limited by the low recall of the underlying tagger (0.71). This suggests that use of an improved tagger would yield significantly better results on the gene normalization task.

Additionally, our analysis suggests an alternative approach to the gene list task, namely treating disambiguation as an evidence combining task: given all tagged gene names in a text, it may be possible to combine evidence from adjacent words, from multiple occurrences of words and synonyms, and from statistics on the observed frequency of gene occurrence, to determine which genes are really mentioned and which are false positives.

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., to other model organism genome databases, as is being done in BioCreAtIvE, and to other biological entities). Entity tagging also provides the foundation for more complex tasks, such as relation extraction (e.g., using the BIND database), functional annotation (e.g., using GO terms, as in BioCreAtIvE Task 2) or attribute extraction (e.g., using FlyBase to identify attributes such as RNA transcript length, associated with protein coding genes).

References

[1] Available from: <http://www.flybase.org>.
[2] Available from: <http://www.informatics.jax.org>.
[3] Available from: <http://genome-www.stanford.edu/Saccharomyces/>.
[4] Available from: <http://rgd.mcw.edu/>.
[5] Available from: <http://www.wormbase.org/>.
[6] Available from: <http://pir.georgetown.edu/pirwww/pirhome3.shtml>.
[7] Available from: <http://us.expasy.org/sprot/>.
[8] Available from: <http://www.sanger.ac.uk/Software/Pfam/index.shtml>.
[9] Available from: <http://us.expasy.org/prosite/>.
[10] Available from: <http://www.bind.ca/>.
[11] Available from: <http://www.jenner.ac.uk/BacBix3/PPprints.htm>.
[12] Hirschman L, Park J, Tsujii J, Wong L, Wu C. Accomplishments and challenges in literature data mining for biology. Bioinformatics 2002;17:1553–61.
[13] Tuason O, Chen L, Liu H, Blake J, Friedman C. Biological nomenclatures: a source of lexical knowledge and ambiguity. In: Proceedings of the pacific symposium for biocomputing; 2004.
[14] Pierre S, Designing an XML-based curation environment. Available from: <http://people.type-z.org/seb/pro/ST40-swissprot.pdf>.
[15] Yeh A, Hirschman L, Morgan A. Evaluation of text data mining for database curation: lessons learned from the KDD challenge cup, intelligent systems in molecular biology, June 2003. Bioinformatics 2003;19(Suppl. 1):i331–9.
[16] Donaldson I, Martin J, de Bruijn B, Wolting C, Lay V, Tuekam B, et al. PreBIND and Textomy—mining the biomedical literature for protein–protein interactions using a support vector machine. BMC Bioinform 2003;4:11.
[17] Available from: <http://www.textpresso.org/>.
[18] Bikel D, Schwartz R, Weischedel R. An algorithm that learns what's in a name. Mach Learning 1999;34:211–31. Special Issue on Natural Language Learning.
[19] Hirschman L, Morgan A, Yeh A. Rutabaga by any other name: extracting biological names. J Biomed Inform 2002: 247–259.
[20] Fukuda K, Tamura A, Tsunoda T, Takagi T. Toward information extraction: identifying protein names from biological papers. In: Pacific symposium for biocomputing; 1998. p. 707–18.
[21] Franzen K, Eriksson G, Olsson F, Asker L, Liden P, Coster J. Protein names and how to find them. Int J Med Inform 2002;67(1–3):49–61.
[22] Collier N, Nobata C, Tsujii J. Extracting the names of genes and gene products with a Hidden Markov model. In: Proceedings of COLING '2000; 2000. p. 201–7.
[23] Krauthammer M, Rzhetsky A, Morosov P, Friedman C. Using BLAST for identifying gene and protein names in journal articles. Gene 2000;259:245–52.
[24] Tanabe L, Wilbur WJ. Tagging gene and protein names in biomedical text. Bioinformatics 2002;18(8):1124–32.
[25] Gaizauskas R, Demetriou G, Artymiuk PJ, Willett P. Protein structures and information extraction from biological texts: the PASTA system. Bioinformatics 2003;19:135–43.
[26] Kazama, Jun'ichi, Takaki Makino, Yoshihiro Ohta, Jun'ichi Tsujii. Tuning support vector machines for biomedical named entity recognition. In: Proceedings of the workshop on natural language processing in the biomedical domain (ACL 2002). Philadelphia, PA: USA; 2002.
[27] Craven M, Kumlien J. Constructing biological knowledge bases by extracting information from text sources. In: Proceedings of the seventh international conference on intelligent systems for molecular biology; 1999. p. 77–86.
[28] Larsen EW. Genetic analysis of modifiers affecting sexual dimorphism and temperature sensitivity of bx1 in Drosophila melanogaster. Dev Genet 1989:106–11.
[29] Available from: <http://www.biopython.org>.
[30] Available from: <http://predictioncenter.llnl.gov/>.
[31] Available from: <http://trec.nist.gov/>.
[32] Available from: <http://www.itl.nist.gov/iaui/894.02/related_projects/muc/info/muc_eval.hw>.
[33] Available from: <http://www.mitre.org/public/biocreative/Task1BGuidelines.pdf>.
[34] Available from: <http://www.edict.com.hk/TextAnalyser/wordlists.htm>.
[35] Hatzivassiloglou V, Duboue P, Rzhetsky A. Disambiguating proteins, genes, and RNA in text: a machine learning approach. Bioinformatics 2001:97–106.
[36] Cohen KB, Dolbey A, Hunter L. Contrast and variability in gene names. In: Proceedings of the workshop on natural language processing in the biomedical domain. Association for Computational Linguistics; 2002.
[37] Yamamoto K, Kudo T, Konagaya A, Matsumoto Y. Protein name tagging for biomedical annotation in text. In: Proceedings of the workshop on natural language processing in the biomedical domain. Association for Computational Linguistics; 2003.
[38] Palmer D, Burger J, Ostendorf M. Information extraction from broadcast news speech data. In: Proceedings of the DARPA broadcast news and understanding workshop; 1999.
[39] Available from: <http://www.flybase.org/bin/fbidq.html?FBgn0003717>.

[40] Liu H, Friedman C. Mining terminological knowledge in large biomedical corpora. In: Proceedings of the pacific symposium on biocomputing; 2003.
[41] Pustejovsky J, Castaño J, Saurí R, Rumshisky A, Zhang J, Luo W. Medstract: creating large-scale information servers for biomedical libraries. In: Proceedings of the ACL 2002 workshop on natural language processing in the biomedical domain; 2002.
[42] Available from: http://www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/.
[43] Available from: http://www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/results/.