Developing Systems for Gene Normalisation

A thesis presented
by

Benjamin Goudey

to
The Department of Computer Science and Software Engineering
in partial fulfillment of the requirements
for the degree of
Bachelor of Computer Science (Honours)

University of Melbourne
Melbourne, Australia
October 2007

Thesis advisor(s)
Nicola Stokes
David Martinez

Author
Benjamin Goudey

Developing Systems for Gene Normalisation

Abstract

The rapid growth of biomedical literature has attracted interest from the text mining community to develop methods to help manage the ever-increasing amounts of data. Initiatives such as the BioCreative challenge (Hirschman *et al.* 2005b) have created standard corpora and tasks in which to evaluate a variety of systems in a common framework. One such task is gene normalisation, in which the problems of synonymy and polysemy in gene name identification are overcome by mapping each mention back to a unique identifier, unambiguously identifying that gene. This task is one of the foundations required for any kind of text mining system working with biomedical literature, where we must be very certain of which genes are being discussed in the text.

In this work, we present two systems for gene normalisation: a naive system performing no disambiguation and a machine learning-based approach which attempts to overcome limitations in the work of (Crim *et al.* 2005). These systems are evaluated on data taken from the first BioCreative challenge.

For each of these systems, a variety of methods are examined to assist gene name identification, either by adding new gene names or removing unlikely candidates. These techniques successfully improving the gene name identification of our system. We find that with data related to some organisms, filtering out unlikely gene name candidates allows the naive system to achieve high performance without the need for further disambiguation. After optimising the identification phase, we try to improve the machine learning approach by implementing a variety of novel features, expanding upon the small feature set used by Crim *et al.* (2005). Unfortunately, we find that a larger feature set has little impact on results. An analysis of the data and of the errors generated by our system reveals large difference between the data of each organism, indicating that better performance may be obtained by creating different solutions for different organisms.

Contents

Chapter 1

Introduction

Consider a biologist reading an article about the fruit fly, who comes across the *clock* gene, a gene which helps a fly's body to keep track of time. As our biologist is interested in this gene but knows little about it, he turns to Pubmed (Wheeler *et al.* 2004), a search engine across biomedical citations and research abstracts, and performs a search for the term *clock*. Below are three text fragments from documents which are associated with the *clock* gene.

- *...as a physiological **clock**, it appears...*
- *...The mouse **clock** gene encodes a ...*
- *...the **period** gene, a central component of...*

In the fragment taken from the first document, we see that the term *clock* has been found, but in this context it refers to a time-keeping device, rather than a specific gene. This demonstrates the first issue we must overcome when performing text mining in the biomedical domain; the names of genes and gene products are quite ambiguous. To perform a task effectively, there must be a method to determine when a reference refers to a gene name and when it does not.

While the second document contains a reference to a *clock* gene, it is not the gene we are after. Instead, it is a gene within the mouse, rather than the fly. Not only do genes have many different names, but a single gene name can refer to different kinds of genes. These genes can either occur within different species, as demonstrated in the example here, or alternatively there may be different genes within a single organism that go by the same name. In the fly, *clock* can refer to 6 different genes.

Finally, in the third document, we see that the term *clock* does not appear. However, the term *period* does appear and in this case, it is a synonym for the gene that we are searching for. Unfortunately, as it is labeled here with a different name, this document would not be returned by a standard search. This highlights the third issue; a single gene can have many synonyms. The *clock* gene in this example has 13 different names. If a text mining task is unable to take all of these names into account when looking for a gene, then many occurrences of it will be missed.

Any text mining system must not only be able to identify gene names, but must be able to determine exactly which gene is being referenced in a specific context. These two subtasks of gene name identification and disambiguation make up the task of *gene normalisation*, a task that is one of the foundations required for any kind of text mining system working with biomedical literature. To perform fact extraction procedures, such as protein-protein interactions or the generation of co-occurrence statistics, we must be very certain of which genes are being discussed in the text. It is also important for information retrieval tasks, where gene normalisation can facilitate query expansion, increasing the number of relevant documents available.

If such a system is applied to our example above, the query term is replaced with a unique identifier, *FBgn0003069*. This term also replaces *period* in the third document, while the occurrence in the second document is replaced with an identifier for a mouse gene, *MGI:99698*. The first fragment has no identifier as there is no gene being discussed. Once this has been performed, it can be clearly seen that only the third document is relevant.

The task of gene normalisation was recently investigated at the BioCreative Challenge (Hirschman *et al.* 2005b), a text mining evaluation focusing on the biomedical domain. Here, Crim *et al.* (2005) presented a gene normalisation system which uses a machine learning-based approach. Though one of the best performing systems at BioCreative, this system gives few details as to how identification of genes is performed, relies on an incomplete list of gene names, and a uses a quite limited feature set.

We present a system, based on a similar framework to Crim *et al.* (2005) , which attempts to overcome these limitations. This is contrasted with a dictionary-based lookup approach, which performs no disambiguation. For each of these systems, a variety of methods are examined to assist in the identification phase, either by adding new gene names or removing unlikely candidates.

We examine the performance of a state-of-the-art gene identification system, demonstrating that the rich information that can be obtained from biomedical databases allow a simple dictionary-based lookup system to achieve competitive identification performance for some organisms.

A number of novel features are implemented to extend the feature set applied by Crim *et al.* (2005), derived from external resources and lexical information. The use of additional features leads to only slight increases, revealing the difficulty of this task.

Finally, we present an analysis of each data set which identifies the complexities and ambiguities unique to each organism followed by an analysis of the errors generated by our system. The combination of these analyses shows each organism suffers from a quite different issues and creating separate solutions for different organisms may yield better results.

In Chapter 2, we give a detailed definition of the task, introduce the data used in this work and present previous work in this area. Chapter 3 outlines the architecture used by our system, with Chapter 4 presenting the results of our experiments. Chapter 5 contains a more detailed analysis of our results, with our final conclusions and possible future work in Chapter 6.

Chapter 2

Background

In this section, we provide an overview of biomedical text mining, followed by some basic definitions associated with gene normalisation. We then discuss the BioCreative challenge, with a focus on the gene normalisation tasks and examine the data produced from BioCreative for use in this task. A summary of previous work related to gene normalisation is also presented.

2.1 Biomedical text mining

Within the last decade, research efforts like The Human Genome Project (Lander *et al.* 2001) and recent technological advances, including microarray data generation, advances in DNA sequencing and in medical imaging, have lead to a dramatic increase in the rate at which scientific data can be produced (Buetow 2005). This increase in data has also meant the level of scientific literature generated by the biomedical community has grown rapidly.

One development which has been crucial to managing the flood of new publications has been the advent of online databases and repositories. Publication indices, such as MEDLINE, have given researchers access to millions of biomedical abstracts while gene databases, such as Entrez Gene (Maglott *et al.* 2005), provide up-to-date information on gene families, pathways and homologs can then be used to infer information about newly discovered genes. These kinds of repositories have allowed researchers to share their knowledge with others, facilitating new research(Collins *et al.* 2003).

However, the exponential growth of the literature has been so fast that it is impossible for even a large team of researchers to be able to comprehend a reasonable proportion of it (Cohen and Hunter 2004). This makes it difficult for online repositories to remain up-to-date while maintaining a high quality of data. The idiosyncrasies of biomedical literature also make it difficult for traditional data-management techniques to yield useful results (Buetow 2005). New methods are needed to cope with this information overload.

In an effort to improve current information retrieval processes, attempts are being made by the natural language processing (NLP) community to adapt techniques which have been traditionally used on domains such as news articles to the biomedical domain. Tasks like multi-document summarisation (Maglott *et al.* 2005), allow for better management of the data by making it easier to find and digest information, while new tools are being created to facilitate the automation of database curation (Alfarano *et al.* 2005).

More than just building a better search engine, the large amount of data available in the biomedical literature can also lead to a variety of new inferences about the way that biology works. Research is currently being conducted into using information found in the literature as secondary information to assist with analysis of microarray data (Chaussabel and Sher 2002). Methods are also under investigation which would use text mining techniques to discover new relationships between genes or to discover how sets of proteins interact with each other (Cohen and Hunter 2004). The success of these types of tasks would allow for a reduction in the work done in a lab by narrowing down the search space, allowing biologists to experiment with only combinations that have shown up as likely pairs within the literature. A similar technique can also use the co-occurrence statistics of genes within the literature to build hypotheses of how the underlying biology operates (Bekhuis 2006).

Before any of these tasks can be performed accurately, gene normalisation must be performed to unambiguously identify all genes which are referred to within the literature. It is foundation of many NLP processes in the biomedical domain.

2.2 Gene Normalisation

A gene mention is a textual entity which refers to any gene or its products, such as proteins, RNA, binding sites, promoters etc. Though it is possible to create a more fine-grained definition, whereby each gene product is classified into separate groups, in practice this is quite difficult. Even experts within the biomedical domain are only able to agree 77% of the time on whether a specific mention refers to a gene, RNA or protein (Tanabe *et al.* 2005).

Gene mentions have a variety of idiosyncrasies not encountered in common proper nouns. While the issue of gene mention ambiguity with relation to text mining systems has only become an issue in recent times, researchers have long identified that ambiguity within gene nomenclature has made it more difficult to share information across different biological fields. To overcome this obstacle, naming conventions were created for human genes in the 1950's and 60's setting the way for similar standards in other organisms. However, many in the scientific community have tended to use existing aliases rather than adopting official names set out in nomenclature guidelines, with influential scientific papers having more impact on the choice of gene name than the standards themselves (Tamames and Valencia 2006). Newly discovered genes tend to follow official guidelines, but after a time, often acquire unofficial synonyms.

The enforcement (or lack thereof) of naming conventions for a specific organism have a significant impact on the complexity of different NLP tasks on the related literature. Just because a text mining system may be able to perform well on documents related to one organisms does not imply that it will perform well over all organisms. For example, the nomenclature of yeast tends to follow the related naming conventions quite strictly, resulting in a vocabulary with little ambiguity and a general consistency between names. On the other hand, the official gene names of the fruit fly are often ignored in favour of unofficial aliases and synonyms, resulting in a nomenclature with a significant amount of overlap with English terms, multiple names per gene and many genes sharing the same name.

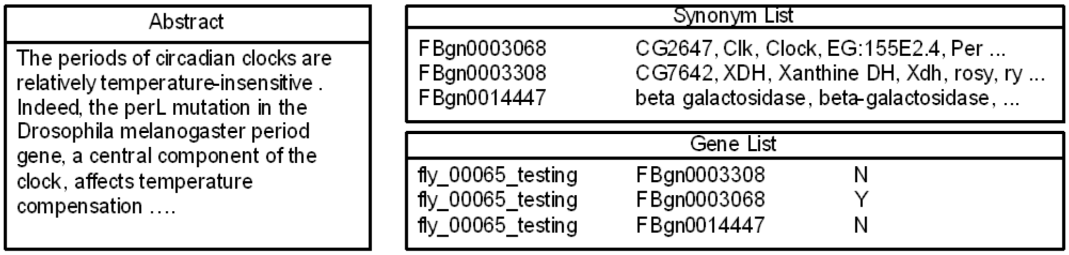
Gene normalisation attempts to normalise gene mentions by mapping each one back to a unique identifier, unambiguously showing the exact gene being referenced. The problem of gene normalisation can be broken down into two subtasks: identification and disambiguation. Identification of gene mentions is similar to named entity detection in a newswire domain, but the idiosyncrasies of biomedical literature add to the complexity of the task. Not only do we retain the problems associated with detecting word boundaries and issues as to what constitutes an important entity, but there are many issues with the naming conventions. New gene and proteins are also constantly being discovered, renamed or are found to be invalid. The model database related to the common mouse shows between 50 and 100 alterations every week to the nomenclature section(Dickman 2003). This dynamic vocabulary means that any identification system created must be flexible enough to keep up future additions.

The inclusion of symbols and non-alphabetic characters in many gene names also causes problems with identification. Tokenisation becomes much more difficult as we can no longer make assumptions that tokens are strings of alphanumeric characters. With mentions such as ”alpha 2/delta subunit 2” or ”PBSF/SDF-1”, it can be difficult to determine whether we are referring to two separate genes or a component of a more complex gene product.

The names of genes often originate from their descriptions, causing some organisms to have a large portion of gene names which are common English words or phrases. Examples from the fruit fly include *period*, *fused*, *for* and *in*. This means we must be able to accurately determine when terms are being used as English words or as gene mentions. A variety of genes have related locations, pathways and domains, and these related entities are often named after their associated gene. This also adds to the complexity of identification as these related entities can be difficult to distinguish from the actual gene itself.

For most tasks, we must also disambiguate the gene name to determine exactly which gene is currently being referenced. Quite often, different genes will share the same name and therefore identifying the name will not be enough to inform us which of these genes is being discussed. Normalisation resolves this issue by mapping each gene to an associated unique identifier.

There are two possible sources of ambiguity. A gene mention may require disambiguation between multiple genes within the same organism. Alternatively, a gene mention may refer to the equivalent genes within different organisms. In some cases, a gene mention may conform to both of these cases, requiring us to identify the correct gene in the correct organism. In this work, we are only concerned with disambiguation within a single organism. Even ignoring cross-species ambiguity, this is still non-trivial problem, with terms in the fruit fly having up to 108 unique genes associated with them.



Abstract

The periods of circadian clocks are relatively temperature-insensitive . Indeed, the perL mutation in the Drosophila melanogaster period gene, a central component of the clock, affects temperature compensation ….

Synonym List

FBgn0003068	CG2647, Clk, Clock, EG:155E2.4, Per ...
FBgn0003308	CG7642, XDH, Xanthine DH, Xdh, rosy, ry ...
FBgn0014447	beta galactosidase, beta-galactosidase, ...

Gene List

fly_00065_testing	FBgn0003308	N
fly_00065_testing	FBgn0003068	Y
fly_00065_testing	FBgn0014447	N

Figure 2.1: An example of an abstract, related gene list and the synonym list from the *fly* dataset

2.3 BioCreative and Data

In recent years, there has been a marked attempt to increase the amount of research into text mining on the biomedical domain. Much of this interest has been generated using a staple of the text mining community; by creating a challenge, with related data and guidelines to allow the development of systems and metrics that allow for the creation of new automatic techniques and accurate comparison between systems (Yeh *et al.* 2005). Shared tasks such as the KDD Cup (Yeh *et al.* 2003) and the genomic track of TREC (Hersh and Bhupatiraju 2003) have all helped to generate interest in text processing for the biomedical domain.

A similar evaluation, the BioCreative challenge (Hirschman *et al.* 2005b) was created with the intent of providing a “systematic assessment” of biomedical text mining systems. It ran in 2004 and again in 2007, with the first having over 27 groups from 10 different countries participate.

Both years focused on 3 main tasks: gene name identification, gene normalisation, and extracting protein-protein interactions. For each of these tasks, a corpus of data was generated, as well as associated scoring software. While the gene identification and protein-protein interaction tasks consisted of data that was created manually, similar to that of the MUC workshops (Chinchor 1998), the gene normalisation data was generated automatically in an attempt to ”explore the hypothesis that expert-curated biological databases provide sufficient resource for the creation of high quality text mining tools” (Hirschman *et al.* 2005a). The details of this generation are described in the following section. The nature of the automatic process meant it was not possible to annotate each gene within an abstract. Instead, a list of the genes mentioned in each abstract are created.

Gene normalisation was the focus of task 1B of BioCreative, and it is from here that we take our task definition and data. The aim of this task is to take a set of abstracts for a given organism and produce a list of gene identifiers indicating the genes that are mentioned within each abstract. The data consists of separate datasets for three different organisms: *yeast*, *mouse* and *fly*. These organisms were chosen as each has a related model database, containing a wide variety of information specific to each organism. Each dataset consists of three components: a set of abstracts, a list of gene synonyms specific to each organism and a list of genes contained in each abstract. An example of each of these data components is shown in Figure 2.1.

2.3.1 Abstracts

The abstracts are taken from MEDLINE and consists of 5000 abstracts per organism for use as training data as well as 108, 110 and 250 abstracts in the development test data for yeast, mouse and fly respectively. The final testing data consists of a further 250 abstracts for each organism. Each abstract contains only the text with further details, such as the document's title or publication information, stripped from the document.

2.3.2 Synonym Lists

Each organism also has a related synonym list, containing all known gene identifiers and the gene mentions that have been associated with each of them. These list have been manually created by each model organism database. The list are incomplete due to the variety of lexical variations in the literature and the limits of the original database from which the list is derived.

2.3.3 Gene Lists

For each abstract, a list of gene identifiers has been created, containing all genes referenced in the abstract. These gene lists were derived from manually created gene lists that apply to the whole document, created by curators of each model organism database. These lists therefore had to be filtered to only contain genes mentioned in the abstract. This filtering was performed using an automatic and noisy process. Possible gene mentions were identified using a organism's synonym list. If one of these mentions occurred in the abstract and had an associated gene identifier in the gene list, then the gene identifier remained in the list. All gene identifiers not identified in the abstract were marked as not present. The information of where each gene mention occurs in the abstract is not provided.

Although this method allows for the creation of a large corpus of training data due to its automatic and relatively efficient nature, there are many opportunities where noise could be introduced. Table 2.1 shows the estimated levels of recall for the training data. As previously mentioned, the synonym lists provided by each model organism database are incomplete. If a gene mention within a document contains a different spelling variation to its equivalent in the synonym list, then the gene will be marked as absent from the abstract.

Similarly to the completeness issues of the synonym list, the gene list provided by the model database may also not contain all genes within a given document. This occurs when a model database is specifically concerned with genes of a specific function or expression level, causing it to leave the more basic ”house-keeping” genes off its gene lists (Colosimo *et al.* 2005). If a gene occurs in an abstract, but is not marked on the gene list, it is as if that gene does not exist.

Evaluating on this noisy data would not provide a true measure of performance, as a system could correctly normalise all genes in a text, but would be penalised as many genes would

	# of abstracts in training data	# of abstracts in development data	# of abstracts in test data	Estimated precision of training data	Estimated recall of test data
Yeast	5000	108	250	98.5	86.0
Mouse	5000	110	250	99.0	55.2
Fly	5000	250	250	86.3	80.7

Table 2.1: Amount and quality of data provided in BioCreative

not appear in the associated noisy gene list. To resolve this issue, the test data had all noise removed by manually correcting the errors introduced by the filtering process. The different levels of noise in the training and test sets makes it difficult to generate to create a high accuracy, supervised machine learning approach.

Due to these issues, the data for this task is quite different to similar corpora. Whereas the data for the named entity recognition tasks in MUC consist of manually tagged entries with each entity being annotated in the document, in this task we only have a noisy lists of genes mentioned in each abstract and no information as to where each gene mention occurs in an abstract.

2.4 Previous Work

Though gene identification has been researched since the early 1990's, gene normalisation has only begun to be explored in recent years. Most systems have relied on the BioCreative data, as this appears to be the only widely used source of data avaliable for this task.

Morgan *et al.* (2004) split the problem into 3 stages: identifying genes in the abstract, matching these gene names to those in a synonym list and finally, disambiguating gene names. The data used was a superset of the fly dataset in BioCreative. To identify genes, noisy training data was generated from the abstracts and their associated gene lists and was used to train a gene identification tagger. Disambiguation was performed by a series of filters which would remove any ambiguity. Using this method, they were able to generate a final F-score of 72%. Though this system used similar data to the systems in BioCreative, it is not directly comparable to results from the evaluation as it used an extra 8033 abstracts to assist with training.

Two systems at BioCreative also utilised an automated approach. Crim *et al.* (2005) chose to use a machine learning approach for the disambiguation phase. An identification phase determines all possible gene mentions in an abstract by matching text to entries in the synonym list, generating a high recall identification system. In the training phase, each match generated by the identification phase was used to generate a series of positive and negative instances which were then passed to a machine learning algorithm. A maximum entropy classifier was used with a small feature set, achieving good results for fly and mouse and the highest results in the competition for yeast. The architecture for this system is described in greater detail in Section

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
Hachey *et al.* (2004)	96.9	75.4	84.8	77.0	59.6	67.2	59.2	74.8	66.1
Hanisch *et al.* (2005)	96.6	84.0	89.9	76.6	81.4	**79.0**	93.1	80.0	**81.6**
Fundel *et al.* (2005)	91.7	87.8	89.7	76.4	78.1	77.3	80.2	73.7	76.8
Crim *et al.* (2005)	95.6	88.1	**91.7**	78.7	73.2	75.8	70.4	78.3	74.2
Wellner (2006)	94.5	90.2	90.2	79.5	74.3	76.8	76.7	76.7	76.7

Table 2.2: The performance of previous systems over the BioCreative dataset. The best F-score per organism is shown in bold

3 as we use this framework as the basis for our machine learning-based approach for this work.

Hachey *et al.* (2004) took a similar approach to that of Morgan *et al.* (2004), but used only the BioCreative data. Noisy training data was generated from each organism's dataset which was then used to train a gene identification tagger that had been submitted in a separate BioCreative task. A series of further measures to improve recall of gene names in the synonym list and to improve disambiguation were then presented, utilising TF-IDF scores, co-occurrence and heuristic methods.

In general, the rule-based approaches from BioCreative achieved better performance than those based on machine learning techniques. Hanisch *et al.* (2005) created a system which relied on heavily modified synonym lists, with the addition of automatically generated lexical variations and filtering, as well as manual additions and removals of gene names from the provided synonym lists. An approximate search was used to identify genes, relying on similarity scores derived from the class of words being matched and the number of matching tokens. Disambiguation is performed by choosing the identifier with the highest number of occurrences within a document.

Fundel *et al.* (2005) chose to explore the performance of a simple dictionary-based lookup approach, using the combination of automatic and manual expansion used by Hanisch *et al.* (2005). Exact matching was used to locate candidate gene mentions, which were then filtered based on a list of words known to indicate the matched text does not refer to a gene. This simple system was able to achieve results close to the best within all BioCreative submission.

Wellner (2006) tried to improve results by improving the quality of the training data. This was achieved using weakly supervised methods and high precision gene identification tools to re-label incorrect instances. The system was based on a similar framework to Crim *et al.* (2005), but was unable to make any performance gains.

In Table 2.2, we present a summary of the results achieved by the various systems over the BioCreative data. As Morgan *et al.* (2004) did not use the same data, it has not been included.

Chapter 3

System Description

The system we have created uses a framework similar to that used by Crim *et al.* (2005). We have altered this to allow for the generation of two different systems; a Naive Dsystem which performs no disambiguation and a more complex Supervise Disambiguation system. This architecture is outlined in Figure 3 and is described in the following section.

There are 3 main stages in our system. First, the document is run through a high recall gene identification system. By processing the provided synonym list in a variety of ways, we are able to identify more than 89% of gene mentions, though we generate a large amount of false positives. The choice of system then determines the disambiguation stage. If naive disambiguation is employed, all gene identifiers related to the gene mentions we have found are added to the output gene list. If a machine learning-based system is used, then each candidate mention is used to create a series of instances for each possible gene identifier related to the mention. A variety of features are then extracted from contextual text and instances are passed to a maximum entropy classifier. In the training phase, these instances are used to train the classifier while in the testing phase, each instance is classified and a confidence value is returned. A gene identifier is added to a document's gene list if it has the highest positive confidence value out of the instances for each mention. The following sections explain each of these steps in greater detail.

3.1 Candidate Identification

The first phase, the identification of possible genes, tries to capture all textual entities which could be a gene mention. Its inputs are a synonym list, where all forms of gene mentions are stored, and the abstracts to be processed. The goal is to create a high recall system, as the recall found here will place an upper bound on total recall that can be achieved by the system. The synonyms and the text in each abstract are converted to lowercase and all symbols are removed. Words from the synonym list are then matched to text in the abstract using longest extent pattern matching to locate candidate gene mentions. While Crim *et al.* (2005) describes a similar phase,

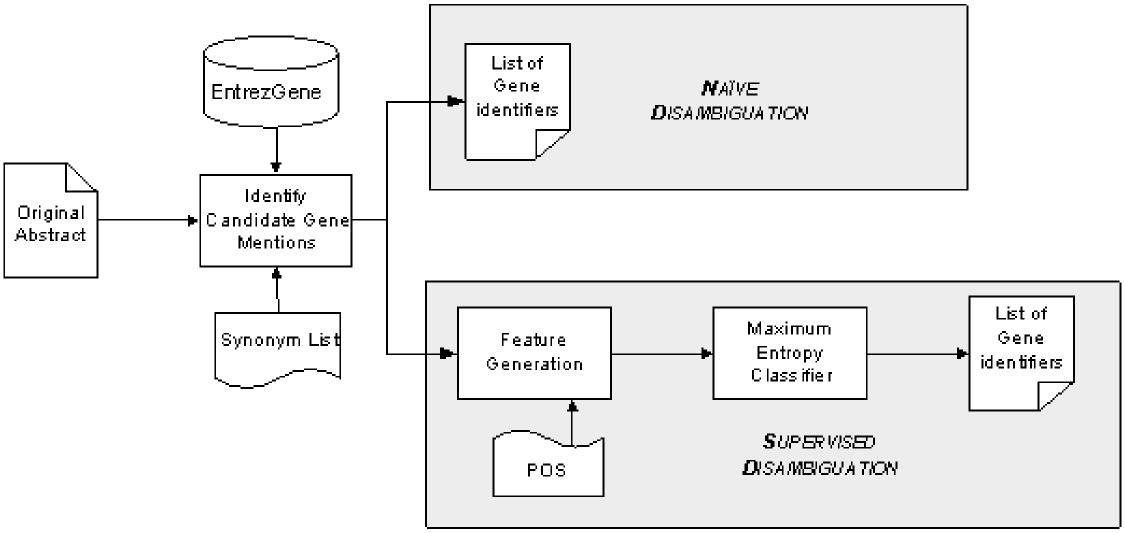


Figure 3.1: A outline of the system architecture

few details are given as to how it is performed.

This system is reliant on an exact match of words in the synonym list to the text. As we have noted that the synonym list is incomplete, we attempt to account for gene name variations by expanding the given synonym list using three methods: generation of lexical variations, extracting additional names from external sources and stemming. Identification of gene mentions has been limited in other gene normalisation systems as the synonym list provided does not cover all possible variations. By adding to this list, the recall achieved after classification should be improved.

Generation of lexical variations has been used with great success by biomedical information retrieval systems to help account for the small differences in spelling and punctuation between similar biomedical terms(Bttcher *et al.* 2004). First, mentions in the synonym list which contain no letters or consist of only a single character are discarded. For any mention with a subtype specifier, a variation is created with the specifier discarded (Cytochrome b → Cytochrome). If the candidate gene mention contains numbers or any Greek letters, it is considered a candidate for expansion. For such gene mentions, hyphens are replaced with spaces and variations are created with different combinations of Greek letters converted to their English equivalent and vice-versa (alpha ↔ a), and variations with spaces placed between alphabetic and numeric characters. No organism specific variations are created. An example of such variations can be illustrated with the mention *LSP-1 beta* which would have the following variants: *LSP 1 beta*, *LSP 1 b*, *LSP 1beta*, *LSP 1b*, *LSP1 beta*, *LSP1 b*, *LSP1beta*, and*LSP1b*.

We also seek to augment the synonym list by looking at external sources. We used information extracted from Entrez Gene(Maglott *et al.* 2005), a gene database containing information extracted from a variety of more specific, curated databases. From this, we were able to add a variety of gene and protein names that were not found in the original list. This allows our system to achieve higher recall without the need for organism specific rules or manually entered data.

Stemming was explored using the Porter stemmer (Porter 1980). Although designed for general English text, we felt a stemmer would help to match multi-word names with variations that we would otherwise be unable to capture. For example, the mention ”proliferator antigen receptor” is in the synonym list of the mouse, but is referenced in an abstract as ”proliferating antigen receptor”. When we use the Porter stemmer, both mentions become ”prolifer antigen receptor” and the match in the abstract would be found.

From this stage, the naive and supervised systems diverge. If the naive approach is used, all identified gene mentions and all of their associated gene identifiers will be added to the resulting gene list. While this method is quite simple, it has been shown that naive disambiguation with an augmented synonym list can achieve state-of-the-art performance. The performance of the synonym list also gives a rough estimation as to the performance of the identification phase.

If a machine-learning approach is taken then we use the results from the identification phase to generate instances, describe in the following sections.

3.2 Instance Generation

Once all potential gene mentions have been discovered, we use each match to generate data to train our binary classifier. In the training phase, this is done by creating a series of positive and negative instances from each match found in the identification module. If a gene mention is identified in a document and the mention’s associated gene identifier appears in that document’s gene list, the generated instance is labeled as a positive match. If a gene is identified and its associated gene identifier fails to appear in the documents gene list, it is labeled as a negative instance. This processes is best illustrated with the following two text fragments;

- ***fused*** *(fu) is a segment-polarity gene*
- *A minimal promoter* ***fused*** *to such sites*

Consider the above two fragments of text, taken from separated documents, in which the word *fused* was found to be a possible gene identifier. The gene list associated with the first example contains the gene identifier, *FBgn0001079*, an identifier which is also associated with the gene *fused*. When generating training instances, the combination of this identifier and one of its synonyms, in this case the term *fused*, creates a positive instance. However, *fused* is an ambiguous term with nine associated gene identifiers, similar to unique senses in word sense disambiguation. Each of the eight remaining gene identifiers which were not contained in the documents gene list are used to create negative instances. In test data, we would still generate nine instances, one for each associated gene identifier, but leave the positive or negative classification to the machine learning algorithm.

The second example fragment also contains the text *fused*, however, in this case it is as an English word. As no gene identifiers in the document's gene list are associated with the mention, 9 negative instances are created.

3.3 Features

For each instance that is created, a set of features are generated which are used to assist disambiguation. The system implemented by Crim *et al.* (2005) contains only very basic features: the two tokens before and after the candidate mention, the gene identifier associated with the mention and the total number of gene identifiers associated with the current gene identifier. Though quite simple, these features were shown to perform quite well. In this paper we investigate a variety of features which intuitively gave more context to each mention. The range from external information, such as part of speech (POS) tags created using the MedPost tagger (Smith *et al.* 2004), to contextual information, such as spacing and capitalisation. The feature set is listed below with features used in Crim *et al.* (2005) marked with a ★.

Matching Text (★): The gene name found in the text, as it occurs in the original abstract.

Gene Identifier (★): The unique gene identifier associated with the candidate gene mention.

Gene Mention POS: The part of speech tags associated with the matched text. If multiple words exists, all tags are concatenated together separated by underscores.

Previous/Following Word (★): Extract the words from before and after the candidate gene mention within a given context size. Each of these words is then used as a separate feature. We used a window size of 2.

Previous/Following POS: Extract the part of speech tags associated with each of the words used in the previous feature

Closest Verb Left/Right: Extract the verbs which closest to the gene mention on either side.

Is English Term: A binary feature testing whether the candidate gene mention is an English word This is determined by checking whether the mention has an entry in the WordNet database (Fellbaum 1998).

Is Stopword: Does the candidate mention appear in the Zettair (Billerbeck *et al.* 2004) stopword list.

Same Othography: A binary feature testing if the capitalisation of the candidate mention is the same as that of its match in the synonym list. This is necessary as the matching performed is done on lowercase text with no symbols.

Seperated From Next: A binary feature testing if the word is separate or hyphenated to the following token.

Appears in Entrez Gene: A binary feature testing if the gene appears in Entrez Gene.

Number of Words: The more words a gene mention has, the more likely it is to be a positive match.

Length in characters: Similar to above, longer gene names tend to be a correct matches.

Amount of Polysemy (★): The number of gene identifiers that are associated with the current gene mention.

Amount of Synonymy: The number of gene mentions associated with the current gene identifier.

Conditional probability: Determines the probability that if a gene mention occurs in a document, what is the probability that the an identifier of the gene will be in the document's gene list. This applies to each combination of gene mention and identifier and is given by the ratio:

$$\frac{\text{\# of abstracts where mention occurs with specific identifier}}{\text{\# of abstracts containing gene mention}}$$

3.4 Disambiguation

Instances are then passed through to the classification module to be used in a maximum entropy classifier. Maximum entropy estimates the conditional probability of a class based on a set of constraints with Each constraint expressing a characteristic of the training data(Nigam *et al.* 1999). The probability distribution that satisfies these constraints is the one with the highest entropy. This model takes the form:

$$P(c|\mathbf{f}) = \frac{\exp(\sum_i \lambda_i f_i(c, \mathbf{f}))}{Z(\mathbf{f})}$$

where c is the class, **f** is a feature vector and Z(**f**) is a normalising term. Our system also used a Gaussian prior of 1.0 to reduce the probability of overfitting the model to the training data.

Maximum entropy classification has been shown to perform well in NLP tasks due to its ability to perform well over noisy datasets and highly dependent features(Berger *et al.* 1996). They are also very efficient to train and to classify, even over large datasets with high dimension feature spaces. We chose to use the maximum entropy classifier implemented in MALLET(McCallum 2002).

Each of the positive and negative instances generated from the training data are used to train the classifier which is then used to classify all test instances. Once each test instance has been classified, we look at each candidate mention which has been identified in each test document. If a mention has one or more positive classifications, we take the instance with the highest probability and add the associated gene identifier to the current document's gene list. If there are no positive instances, the match is discarded and we examine the next possible candidate.

Chapter 4

Results

In this section, we explore the performance of various system combinations. After presenting our evaluation metrics and a series of baseline systems, we explore the use of an out-of-the-box gene identification system for the identification phase. We then test our proposed methods for synonym list augmentation over the Naive Disambiguation and Supervised Disambiguation systems, comparing and contrasting their performance. Finally, we explore the effects of using an expanded feature set in an attempt to build a more effective classifier.

4.1 Evaluation Metrics

The system is evaluated in terms of 3 scores: recall, precision and F-score. These measures allow one to interpret how successful a method was at its task compared to a gold standard dataset. In the context of gene normalisation, precision and recall are defined as:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}$$

where true positives (TP) refers to the number of gene identifiers correctly added to the gene list, false positives (FP) refers to the number of gene identifiers incorrectly added to the gene list while false negatives (FN) refers to the number of gene identifiers that have not been detected by the system. The F-score is the harmonic mean of precision and recall or

$$\textit{F-score} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

These values are calculated using the scoring script supplied with the BioCreative data.

4.2 Baseline Experiments: Naive Disambiguation

In order to judge the difficulty of creating a gene normalisation system, we created a series of simple dictionary-based lookup systems which match gene mentions found in the synonym list

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
First	95.2	67.5	79.0	12.1	65.8	20.4	5.9	60.8	11.8
MostCommon	95.2	67.7	79.2	12.4	67.5	21.0	7.9	81.1	**14.8**
AddAll	92.0	79.1	**85.1**	14.3	83.5	**24.4**	4.1	94.6	7.8

Table 4.1: Baseline normalisation systems using a common lookup approach with different disambiguation methods. The best F-score per organism is shown in bold

to each abstract. Any match found is counted as an instance of a gene. Three variations of this system were created; their differences are based on how disambiguation was performed when an identified gene mention was associated with multiple identifiers. These naive disambiguation systems are as follows:

- **First -** Simply choose whichever gene identifier was the first in the synonym list
- **MostCommon** - Add the gene identifier which had appeared most in the training data
- **AddAll** - Add all gene identifiers to the gene list.

To demonstrate the difference between these systems, consider this example: if the gene mention *fused* was found with possible identifiers *FBgn0014573*, *FBgn0001079* and *FBgn0017900*. The first system would match *FBgn0014573*, the second system would match *FBgn0001079*, the most common of the three identifiers, while the third system would add all identifiers to the gene list, creating much higher recall at the cost of precision.

As shown in Table 4.1, there is generally little difference between adding the first identifier in the list and adding the most common, though the latter generates a reasonable increase in the amount of recall in the fly. Performing no disambiguation and adding all gene identifiers, as in the third method, leads to a large increase in recall for all organisms. As expected this comes at the cost of precision, with severe degradation occurring in fly data, as it has more identifiers per gene mention than the other two organisms.

4.3 Use of Pre-existing Gene Identification Methods

As a contrast to our baseline systems, we chose to examine the performance of a pre-existing gene identification system. We chose the BioTagger package (McDonald and Pereira 2005), one of the best performing gene identification systems seen in the BioCreative 2004 Task 1A (Yeh *et al.* 2005).

The gene identification system is used to processes all abstracts. For each gene found by the tagger, we attempt to locate it in the synonym list. If a match is found, we add all associated gene identifiers to the resulting gene list. Again, this is a naive disambiguation system, but our

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
AddAll	92.0	79.1	**85.1**	14.3	83.5	24.4	4.1	94.6	7.8
BioTagger	95.8	59.9	73.7	88.4	46.1	**51.6**	68.6	24.9	**36.6**

Table 4.2: Results of using BioTagger to identify gene mentions. No disambiguation is performed; instead all possible identifiers for each identified gene are added to the resulting gene list. The best F-score per organism is shown in bold

hope is that it will maintain or increase the recall of the dictionary-based systems while also increasing precision.

From Table 4.2, we can see that the system performs well for the mouse, and while it achieves high precision on the fly and the yeast, it misses many gene mentions, resulting in low recall, compared to state-of-the-art gene normalisation systems. This may be because although trained on MEDLINE abstracts, similar to the data used by our system, the abstracts used to train BioTagger were from a variety of organism. Given the differences between the gene nomenclature of different organisms, the lack of recall in the fly and mouse is unsurprising. Another factor is that BioTagger has been optimised to generate the highest F-score rather than to maximise recall. More work in tuning the tagger to produce recall may allow it to perform more successfully in this role.

4.4 Naive Disambiguation Enhancements

After examining the results of our baseline systems, as well as the lack of recall shown by BioTagger, we hypothesise that the incompleteness of the synonym list limits the ability of the system to achieve high recall. To alleviate this issue, we experiment with a variety of synonym list expansion and filtering methods. We apply these methods to the Naive Disambiguation (ND) system described in Section 3.1, equivalent to the AddAll system in table 4.1 which was the best performing of the baseline systems. As no real disambiguation is performed in this system, the approach can be used to estimate the performance of the identification phase and allow us to gauge how altering the synonym list may affect identification performance.

4.4.1 Synonym List Expansion

We begin by implementing the three methods that were introduced in Section 3.1:

- **Variation** - The creation of gene names variations with different spacing, greek letters etc.
- **EntrezGene** - The addition of genes to the synonym list from Entrez Gene

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
ND	92.0	79.1	85.1	14.3	83.5	24.4	4.1	94.6	**7.8**
ND_Variations	88.3	80.3	84.1	1.9	89.7	3.7	2.0	95.3	3.8
ND_EntrezGene	92.3	89.7	**91.0**	14.3	83.5	**24.4**	4.1	94.6	**7.8**
ND_Stemming	91.2	79.1	84.7	12.7	82.4	22.0	3.7	92.8	7.2
ND_Vars+Entrez	89.0	90.9	89.9	1.9	89.7	3.7	2.0	95.3	3.8

Table 4.3: The effect of various synonym list expansion techniques on the Naive Disambiguation (ND) system. The best F-score for each organism is shown in bold

- **Stemming** - Reducing both the words in the synonym list and the abstract back to their root forms.
- **Vars+Entrez** - The combination of lexical variations and additions from Entrez Gene.

These methods are contrasted with the baseline system using no synonym list expansions. The combination of both Entrez Gene additions and lexical variations is also included. The performance of each technique using the Naive Disambiguation system (ND) are summarised in 4.3

The results from the synonym list expansion are somewhat mixed. Using variations causes an increase of recall in all organisms, but at the cost of degraded precision, especially in the mouse. While this increases the coverage of the synonym list, many of the newly created variations simply add more noise to the data.

Adding information from Entrez Gene works well for yeast, but does not significantly impact on the recall of the mouse or fly. The yeast synonym list seems to be lacking a large number of protein names which we are able to extract from Entrez Gene. This is more efficient than previous systems (Fundel *et al.* 2005; Hanisch *et al.* 2005) as it removes the need to examine the data and create a complex organism-specfic rules. The combination lexical variations and additions from Entrez Gene (ND_Vars+Entrez) gives us a score slightly above that of (Crim *et al.* 2005) for yeast, but brings no change from the variations method in the mouse and fly.

Stemming, though used by other biomedical text mining systems (Hanisch *et al.* 2005), appears less useful for this task, having a negative impact on all organisms. After investigation, it appears that this is due to the inability of the Porter stemmer to deal with the complexities of biological nomenclature. There are a number of cases where two separate gene are reduced to a single gene name after stemming; for example, gene mentions *fu* and *fus* are both reduced to *fu*. The reduction of both names to a single gene mention increases the level of ambiguity, reducing the likelihood of successful disambiguation. While only using additions from Entrez Gene (ND_EntrezGene) achieve the highest F-score, the recall is improved by the combination of Entrez Gene additions and lexical variations (ND_Vars+Entrez). As we believe that more complex systems will achieve improve precision by using better disambiguation techniques, we use the ND_Vars+Entrez system for the following experiments.

4.4.2 Filtering

While the combination of lexical variations and additions from Entrez Gene outlined in the previous section successfully increase the recall of our system, they also result in the addition of more noise. To combat this problem, we experimented with four possible filtering methods:

- **Stopwords** - Uses a stopwords list to filter out gene names that match common words. The list used was taken from Zettair, an information retrieval engine designed for TREC (Billerbeck *et al.* 2004).
- **Nouns/Adj** - Most gene names are recognised as either nouns of adjectives, with many false positives identified as conjunctions or verbs. Removing these should lead to better results.
- **English** - Filtering out all English words is performed by determining whether or not a word exists in the lexical database, WordNet (Fellbaum 1998). English words are less likely to be true observations of a gene mention.
- **Cond** - The conditional probability filter which was used by Crim *et al.* (2005) in their pattern matching system. This statistic gives us a measure of how a specific candidate mention has behaved in the training data, with a higher value showing it has been used more as a gene name than as another word type. This is used later as a feature and has been explained in greater detail in Section 3.3. We chose to filter out all genes which had a conditional probability lower than 0.1, as it discards candidate gene mentions that are almost certainly false positives while retaining high recall.

Each filter was tested using the synonym list with expansions derived from Entrez Gene and the addition of lexical variations (ND_Vars+Entrez system in Table 4.3). The results are summarised in Table 4.4 below.

The first three methods provide no benefit, while removing candidate mentions with a low probability of being gene names works remarkably well. This filter dramatically increases precision in all organisms, with precision in the mouse increasing from 2% to 60%. A manual examination of the results indicates that the gene names targeted by other filters are mainly encompassed by this filter while leaving in those that regularly appear as gene names.

4.5 Supervised Disambiguation Approach

While the results obtained by the Naive Disambiguation system in the previous section are quite high, we believe that the implementation of the Supervised Disambiguation phase can improve results even further. As outlined in Section 3, after the identification phase is

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
ND_Vars+Entrez	89.0	90.9	89.9	1.9	89.7	3.7	2.0	95.3	3.8
ND_Vars+Entrez_Stopwords	89.3	90.9	90.1	2.1	89.7	4.1	2.4	95.6	4.7
ND_Vars+Entrez_Nouns/Adj	92.3	89.6	90.9	1.9	77.6	3.7	2.2	83.2	4.3
ND_Vars+Entrez_English	92.4	89.7	91.1	1.5	75.9	2.9	4.6	65.0	8.5
ND_Vars+Entrez_Cond	93.5	89.6	**91.5**	63.4	77.9	**70.1**	41.8	92.1	**57.5**

Table 4.4: The effect of various filtering methods on a Naive Disambiguation (ND) system using lexical variations and additions from Entrez Gene (Vars+Entrez). The best F-scores are shown in bold

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
SD_NoExpansion	95.4	77.3	85.4	84.4	56.8	**67.9**	75.1	72.5	73.8
SD_Variations	95.2	78.3	85.9	81.8	48.7	61.1	75.0	72.0	73.5
SD_EntrezGene	95.1	88.3	91.5	84.4	56.6	67.8	75.5	72.5	**74.0**
SD_Stemming	95.2	77.3	85.3	82.2	54.4	65.5	72.7	68.3	70.4
SD_Vars+Entrez	94.9	88.9	**91.8**	81.5	49.3	61.4	74.6	72.0	73.3

Table 4.5: Results of various synonym list expansion methods on the Supervised Disambiguation (SD) system

performed using the same technique as in the Naive Disambiguation system, the Supervised system disambiguates candidate gene names by using a maximum entropy classifier, utilising the same features as in the Crim *et al.* (2005) system.

4.5.1 Performance of synonym list expansion methods

To contrast performance with the Naive Disambiguation approach, each experiment from the previous section is repeated using our Supervised Disambiguation (SD) system. We begin by evaluating the effect of the synonym list expansion techniques, with results outlined in Table 4.5.

Performance is consistently better than the Naive Disambiguation system, though the effect of the augmentations are somewhat different. Here, the use of lexical variations (SD_Variations) lowers overall performance for the mouse and fly datasets as the noise introduced makes disambiguation more difficult. The exception to this is the yeast dataset, where the noise which already exists in the data and the amount of noise that we introduced are quite low, resulting in a small increase in performance. Entrez Gene assists yeast by making up for missing names,

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
ML_Vars+Entrez	94.9	88.9	**91.8**	81.5	49.3	61.4	74.6	72.0	73.3
ML_Vars+Entrez_Stopwords	94.9	88.9	**91.8**	81.1	49.6	61.6	74.9	72.5	73.7
ML_Vars+Entrez_Nouns/Adj	94.9	87.9	91.3	75.7	40.6	52.9	74.2	63.6	68.5
ML_Vars+Entrez_English	95.1	88.1	91.4	84.7	45.8	59.4	78.0	48.7	60.0
ML_Vars+Entrez_Cond	94.7	88.3	91.4	78.7	68.6	**73.3**	71.8	82.5	**76.8**

Table 4.6: The results of filtering methods when using a Supervised Disambiguation (SD) system with lexical variations and Entrez Gene

while having very little impact on the mouse and fly. Stemming again has a negative impact on results.

4.5.2 Filtering

The increased noise due to lexical variations appears to lower overall system performance by generating a worse-performing classifier. Previously, we demonstrated that effective filtering can have a dramatic improvement on results by removing some of the noise and ambiguity within the data. We again test our filtering techniques to remove unlikely candidate genes, before passing the remaining candidates to the disambiguation classifier. Results are shown in Table 4.6

While the conditional probability previously helped all systems, here it is only the mouse and fly that improve; yeast results are slightly worse. As yeast data is already quite consistent and suffers from little ambiguity, the filtering of results removes a number of correctly found mentions.

4.6 Comparison of Naive and Supervised Disambiguation

Looking across the results from both the Naive Disambiguation and Supervised Disambiguation systems, we can see quite different responses in each organism. In Table 4.7, we compare the results from each organism using both systems using the best overall setting; that is with lexical variations, additions from Entrez Gene and filtering using conditional probability.

The fly data performs quite poorly with the Naive Disambiguation system due to its high ambiguity while the reasonable quality of its training data means that a useful classifier can be created. In yeast, we are able to obtain better results using only naive disambiguation rather than using maximum entropy to perform disambiguation. This is not especially surprising for the yeast data, given that the nomenclature is very consistent and there is little ambiguity or overlap with general English terms.

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
ND_Vars+Entrez_Cond	93.5	89.6	**91.5**	63.4	77.9	70.1	41.8	92.1	57.5
SD_Vars+Entrez_Cond	94.7	88.3	91.4	78.7	68.6	**73.3**	71.8	82.5	**76.8**

Table 4.7: A comparison of Naive and Supervised Disambiguation systems

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
Crim et al. *(2005) System*	95.6	88.1	**91.7**	78.7	73.2	**75.8**	70.4	78.3	74.2
Crim Features	94.7	88.3	91.4	78.7	68.6	73.3	71.8	82.5	**76.8**
All Features	95.1	88.1	91.4	79.4	71.0	75.0	69.5	82.8	75.5

Table 4.8: The results of our system using all extended features and using only the features specified in Crim *et al.* (2005) using the SD_Vars+Entrez_Cond system

The mouse data is quite different. It is unexpected that only a slight improvement is achieved using the Supervised Disambiguation system. There are two likely causes behind this lack of improvement and both are related to the low recall of the mouse's training data. The first is that the low recall means that a great deal of the data used to train our classifier is incorrect, resulting in a disambiguation phase will is more likely to discard correct genes or perform disambiguation incorrectly. The second factor is that all conditional probabilities are based on the appearance of genes in the training data. If a gene mention in the training data appears, but is never included on the related gene lists due to noise, its conditional probability will be very low. This results in a number of genes being incorrectly discarded. Hence, a lower F-score is achieved when a classifier is decrease in recall outweighs the gains in precision.

4.7 Expanding the Feature Set

After identifying the optimal method for identification and recognising the poor performance of machine learning-based disambiguation over the yeast and mouse datasets, we focus on the features that are used by the system.

We begin with a comparison with results achieved by (Crim *et al.* 2005). We examine the results of two separate feature sets on classification: the first using only those features specified by (Crim *et al.* 2005) and the second using all features in our extended set, as outlined in Section 3.3. Each method is run using the Supervised Disambiguation system with lexical variations, additions from Entrez Gene and using conditional probability as a filter (SD_Vars+Entrez_Cond). The results are in Table 4.8

	Yeast			Mouse			Fly		
	P	R	F	P	R	F	P	R	F
Crim Features	94.7	88.3	91.4	78.7	68.6	73.3	71.8	82.5	76.8
All Features	95.1	88.1	91.4	79.4	71.0	75.0	69.5	82.8	75.5
Incremental Addition	95.5	89.1	92.2	79.4	71.5	75.2	75.3	81.6	78.3
Manual Selection	94.4	90.0	**92.2**	78.8	73.7	**76.2**	75.6	81.5	**78.5**

Table 4.9: The results of feature selection methods on the SD_Vars+Entrez_Cond system

Our gene normalisation system appears to perform somewhat differently to the Crim *et al.* (2005) system. While performance over yeast is comparable, the system by Crim *et al.* (2005) performs better over the mouse than the fly, while our system does the reverse. The differences likely stem from the identification phase where Crim *et al.* (2005) gave few implementation details, providing only the levels of recall achieved. As the Crim *et al.* (2005) system reports recall over 90% over the yeast data, it is highly unlikely that only a dictionary-based lookup is performed.

It is clear that using all available features does not produce the best possible classifier. In order to locate the best performing subset of features, we chose to implement incremental addition feature selection, where the system begins with no features and iteratively tests all features, adding the highest performing feature each round until no higher score can be achieved. We furthered the results of this by manually adding features to determine if any further improvements could be made. The results of each of these are displayed in Table 4.9.

While we are able to make some improvements using only a subset of the features, the gains are quite modest from both incremental addition and manual experimentation. Table 4.10 shows the features that were used in the best run for each organism.

The features used in the different systems vary considerably, with the yeast requiring very few to make a decision, while the fly must draw on more information. Considering the large increase in features used, the gain in performance is quite small. The noise of the training data means that correlations between certain feature values and classes is lessened, reducing the impact of that feature.

	Yeast	Mouse	Fly
Matching Text	✔	✗	✔
Gene Identifier	✗	✗	✔
Gene Mention POS	✗	✔	✔
Previous/Following Word	✔	✗	✔
Previous/Following POS	✔	✗	✔
Closest Verb Left	✗	✗	✔
Closest Verb Right	✗	✔	✔
Is English Term	✗	✔	✔
Is Stopword	✗	✗	✔
Same Othography	✗	✔	✔
Separated From Next	✗	✔	✔
In Entrez Gene	✗	✗	✔
Number of Words	✗	✔	✔
Length in characters	✔	✔	✗
Amount of Polysemy	✗	✔	✔
Amount of Synonymy	✗	✔	✔
Conditional Prob	✗	✗	✔

Table 4.10: The best performing subset of features for each organism

Chapter 5

Discussion

Throughout our experiments, we found the results vary considerably between organisms. In this section, we examine the properties of each organism's data to try and account for this difference in performance. We also examine the types of errors that our system generates to more accurately determine the strengths and weaknesses of our current approach. Finally, we analyse the learning curve of our classifier to determine whether additional data would assist performance.

5.1 Analysis of Datasets

The level of ambiguity differs significantly across the different organisms. Table 5.1 shows the percentage of gene mentions with multiple associated identifiers, with the fly containing considerably more ambiguity than the other two organisms. The fly also has considerably more gene names that are also English terms, at approximately 4%, compared to the 1% for the mouse or the insignificant six words for yeast. Terms such as *period*, *in*, and *type 1* can create a great deal of false positives causing more work and oppertunities for error in the disambiguation stage.

The yeast data is more straightforward: almost all gene names consist of only a single word, it has the fewest identifiers associated with each gene mention and almost no overlap with English words. Its naming conventions have been quite strictly enforced with few variations used and a fairly complete synonym list provided. This is reflected in the results achieved by previous systems and our own, with the yeast performing at least 10-15% better on average (Hirschman *et al.* 2005a).

The mouse data has less ambiguity than the fly, but suffers from other issues. Figure 5.1 shows the number of words per gene mention in each organism. The mouse tends to have longer names, with over 40% of mentions consisting of multiple words. These are generally harder to identify as any slight variation in the spelling stops us from detecting the mention. However, once detected, these longer names are often easier to normalise as a longer string has

# of identifiers	Yeast	Mouse	Fly
1	97.61%	96.83%	84.14%
2	2.14%	2.72%	4.80%
3-4	0.20%	0.30%	3.13%
5+	0.05%	0.15%	7.92%

Table 5.1: Breakdown of number of associated gene identifiers (senses) per gene name. This indicates the ambiguity of gene names

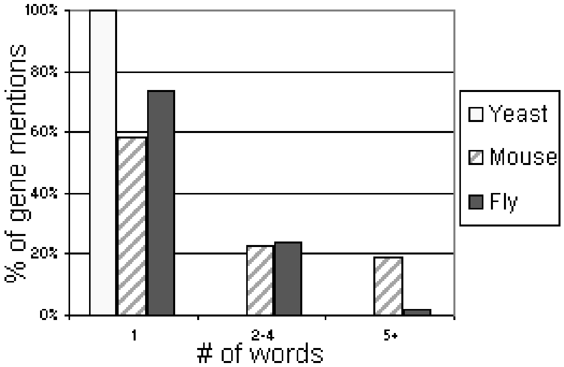


Figure 5.1: The number of terms per gene mention. Yeast contains less than 1% multi-term mentions, while the mouse has over 40%

less chance of randomly occurring in a document.

The low quality of the training data for the mouse also affects its performance. Many of the genes contained in the mouse abstracts were of little interest to the model database's target audience, resulting in their absence from the gene lists. This causes the mouse to have much lower recall than other organisms, with recall in the training data estimated to be only 55%, compared to the 86% and 80% for the yeast and fly respectively, as discuessed in Section 2.3.3. The low recall of the training data severely affects the recall abilities of our classifier.

These varying degrees of ambiguity and quality are consistent with the results that we have obtained in all experiments. Yeast consistently obtains the highest results while the mouse, whose training data is very poor, is generally the worst. The levels of noise in the training data make it very difficult to create an accurate system using supervised techniques. Development of manually annotated data such as that seen in MUC would raise the overall standard of gene normalisation systems, and would allow a much richer comparison between systems.

5.2 Analysis of Errors

In order to determine the major weakness of our system, we examined the first 50 test documents for each organism when using additions from Entrez Gene, lexical variations and removing low probability candidates. From this examination, we determined the causes behind the false positives and false negatives in each document, grouping these into similar classes. A breakdown of the top six classes of errors are shown in Table 5.2 with a brief explanation of each error type as follows:

- **Gene mention not in synonym list** - The gene mention is missing from the synonym list and a similar variation does not exist.
- **Variation missing from synonym list** - A similar gene mention is contained in the synonym list, but there is a slight lexical variation causing us to miss the mention (e.g. PKC-delta and PKCdelta).
- **Filtered due to low conditional probability** - The conditional probability of a gene mention is too low resulting in the gene mention being discarded.
- **Biological term caused incorrect match** - A biological term such as a locus or a pathway is related or derived from a gene mention e.g. *Mos1 promoter*. This term causes an incorrect match as does not actually refer to a gene mention.
- **Match found due to loose tokenisation** - A match was found due to the removal of symbols in the identification phase to allow high recall, resulting in incorrect matches like **um** → ***UM**-HET3*.
- **Match found in different organism** - A gene mention was found, but refers to a gene in a different organism.
- **Other** - Various errors which do not fit in to the categories above.

The analysis revealed the various problems still affecting each dataset. 69% of the errors on the fly dataset are caused by false positives, while 80% and 60% of errors in the mouse are due to false negatives. These support our view that while the mouse and yeast can be normalised by simply filtering out low probability candidates, more powerful solutions are needed to disambiguate fly genes.

Yeast appears to suffer from an incomplete synonym list, as it only includes genes conforms to the yeast nomenclature standards. Gene names that do not conform to these standards are not detected by our identification phase.

The mouse is also affected by missed gene names. These are mainly caused by lexical variations which are not currently covered by our system. A more precise analysis of the kinds of variations that appear in the literature may yield more accurate variation generation.

Error Type	Yeast	Mouse	Fly
FN - Gene mention not in synonym List	11 (55.0%)	1 (2.2%)	2 (6.9%)
FN - Variation missing from synonym list	0 (0.0%)	17 (37.8%)	0 (0.0%)
FN - Filtered due to low conditional prob.	0	8 (17.8%)	4 (13.8%)
FP - Biological term caused incorrect match	3 (15.0%)	10 (22.2%)	7 (22.2%)
FP - Match found due to loose tokenisation	0 (0.0%)	6 (13.3%)	4 (13.8%)
FP - Match found in different organism	0 (0.0%)	2 (4.4%)	7 (24.1%)
Other	6 (30.0%)	1 (2.2%)	5 (17.2%)
Total	20 (100%)	45 (100%)	29 (100%)

Table 5.2: Type and frequency of errors that affect recall. The percentage of overall errors for each type is shown in brackets

The greater level of ambiguity in the data of the fly means more errors tend to stem from incorrect disambiguation. One of the major sources or these is confusion with biological terms related to gene names. This is a difficult problem to solve using only the data provided as these biological terms are often used in quite similar contexts.

It is encouraging that two of the main issues which our gene normalisation system was created to target, cause few of the errors that our system generates. After conditional filtering has been performed, most candidate mentions which consist of English words are removed from consideration due to the fact that they rarely occur as gene names. This removes one of the major sources of ambiguity for both the fly and mouse.

The analysis revealed a number of problems which are outside the scope of our current system. Homologs, that is genes which are equivalent in different organisms, often have the same name regardless of species. For example, the mouse and fly share the *clock* gene. If a mouse abstract discusses such a gene for the fly and the name appears in the *mouse* synonym list, our system will incorrectly associate the gene with the mouse. The analysis also shows that ranges of genes, e.g. *"in genes Otf-3a through Otf-3h"*, are a cause of errors within our system. Currently the system is only able to normalise genes if they are explicitly mentioned. Being able to deal with references to specific clusters of genes is a difficult problem due to the variety of ways in which this can be expressed.

5.3 Learning Curve

Figure 5.2 shows the learning curve of the three different organisms, showing the relationship between the number of documents used to train the system compared to its performance. The graph indicates that while the yeast and mouse data sets are almost at full performance immediately, the fly data requires approximately 40% of the data before reaching a fairly consistent score. Once again, we are able to see the limitations that the classifier has on the yeast

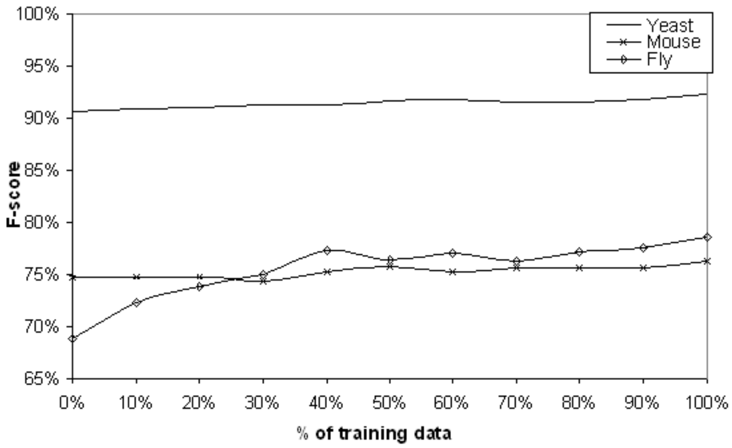


Figure 5.2: Learning curve of the yeast, mouse and fly

and mouse datasets. When using no training data, the classifier assigns a positive classification, adding the gene identifier which appears first to the gene list. On the mouse and yeast datasets, there is little improvement from this point. On the yeast, this is because the data is already quite unambiguous, while on the mouse, it is because the noise in the training data allows the classifier to learn very little. No matter how much more data is added, the classifier is unable to significantly improve. The fly, with its reasonable quality data and high levels of ambiguity, is unable to perform disambiguation without a trained classifier, achieving maximum results after roughly 40% of the training data is provided.

Chapter 6

Conclusion

6.1 Summary

In this work, we presented a gene normalisation system which uses a machine learning-based approach. A variety of extensions were implemented to increase identification coverage, remove unlikely gene mention candidates and improve disambiguation. We evaluated these techniques on data related to yeast, mouse and fly, taken from the BioCreative challenge. The main conclusions from our experiments can be summarised in the following points:

- State-of-the-art gene identification tools are designed to work over the data of all organisms. However, given the differences between different organism's nomenclature, we have found that a basic lookup system utilising a synonym list derived from a model database may be able to achieve better results.
- The addition of gene names taken from external resources proved to be very successful. Turning to additional databases should lead to even higher performance. The generation of lexical variations assists the mouse and fly, but the noise introduced in this process removes some of the possible performance gains. Further refinement of this method is required to achieve the full potential of this technique.
- Filtering of the synonym list by different means shows a lot of promise by removing unlikely candidates and some ambiguity in the data. As we show in Section 5.2, this can be improved by taking into account the fact that we are basing our judgements on noisy data.
- The combination of synonym list expansion and filtering can reduce the need for complex disambiguation in some organisms, with our naive disambiguation system obtaining results better than those of the more complicated machine learning-based approach.
- Improvements to the original feature set used by Crim *et al.* (2005) proved disappointing; our results show increases in performance of less than 2% for each organism. While some of this is caused by noise in the training data, another factor is that many of features do not seem to provide much information as to the classification of specific gene names.

- Our analyses show it is unlikely that a single approach will be able to provide high quality gene normalisation for all organisms as the nomenclature and levels of ambiguity are too varied. This is magnified by differences in the quality of data for each organism.

6.2 Future Work

Improving training data using semi-supervised methods: Clearly one of the limitations current performance is the noise within the training data as this has negative impact on many aspects of our system. Future systems could account for this noise in a variety of ways. (Wellner 2006) used semi-supervised techniques to relabel the data. We feel that a better technique would treat the BioCreative data as unlabelled, using a small set of manually annotated data to bootstrap a semi-supervised learner. The advantage of this technique is that more training data could be easily generated by obtaining additional abstracts.

Improving training data using synonym list expansion: The method used by the BioCreative organisers to generate training data was entirely automatic. However, the process relied on the synonym list provided by the model database related to each organism. As we have demonstrated in this work, the synonym list can be improved through a combination of expansion and filtering. Therefore, if we use the same method as BioCreative to generate training data, but with our improved synonym list, we may end up with higher quality data.

Contextual Information and Topic Signatures: Traditional word sense disambiguation has often had success with the use of topic signatures; context vectors which try to associate a topical vector to each word sense. This technique could be applied to the biomedical domain by associating biological terms, taken from external resources such as UMLS (Bodenreider 2004) or GO (Consortium 2000), to each gene identifier. The occurrence of these biological terms could then be used to determine the sense of a gene mention in a given context.

6.3 Concluding Remarks

To conclude, let us recall the example of the *clock* gene, and examine how our system would perform over the given documents, with the additions of our synonym list augmentations. Rather than the six possible genes which we needed to distinguish before, our conditional filtering reduces the possibilities down to two. For the first document, our classifier successfully discards both of these, identifying this is not a mention of a gene. In the third document, conditional filtering has reduced the mention *period* down to a single option which is successfully classified as the gene we were looking for. Unfortunately, in the second document, the system is unable to recognise that we are no longer discussing the fly and the gene is confidently, and incorrectly, classified as the gene we are after. This example successfully demonstrates that while we are often able to successfully normalise within a specific organism, to be useful in real world applications, we must be able to determine which organism is under discussion in an abstract.

Bibliography

Alfarano, C., C. E. Andrade, , K. Anthony, N. Bahroos, M. Bajec, K. Bantoft, D. Betel, B. Bobechko, K. Boutilier, E. Burgess, K. Buzadzija, R. Cavero, C. D'Abreo, I. Donaldson, D. Dorairajoo, M. J. Dumontier, M. R. Dumontier, V. Earles, R. Farrall, H. Feldman, E. Garderman, Y. Gong, R. Gonzaga, V. Grytsan, E. Gryz, V. Gu, E. Haldorsen, A. Halupa, R. Haw, A. Hrvojic, L. Hurrell, R. Isserlin, F. Jack, F. Juma, A. Khan, T. Kon, S. Konopinsky, V. Le, E. Lee, S. Ling, M. Magidin, J. Moniakis, J. Montojo, S. Moore, B. Muskat, I. Ng, J. P. Paraiso, B. Parker, G. Pintilie, R. Pirone, J. J. Salama, S. Sgro, T. Shan, Y. Shu, J. Siew, D. Skinner, K. Snyder, R. Stasiuk, D. Strumpf, B. Tuekam, S. Tao, Z. Wang, M. White, R. Willis, C. Wolting, S. Wong, A. Wrong, C. Xin, R. Yao, B. Yates, S. Zhang, K. Zheng, T. Pawson, B. F. F. Ouellette3, and C. W. V. Hogue. 2005. The biomolecular interaction network database and related tools 2005 update. *Nucleic Acids Resarch* 33 (Database Issue).

Bekhuis, Tanja. 2006. Conceptual biology, hypothesis discovery, and text mining: Swanson's legacy. *Biomedical Digital Library* 3.

Berger, Adam, Vincent Pietra, and Stephen Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics* 22.

Billerbeck, Bodo, Adam Cannane, Abhijit Chattaraj, Nicholas Lester, William Webber, Hugh E. Williams, John Yiannis, and Justin Zobel. 2004. Rmit university at trec 2004. In *2004 Text Retrieval Conference (TREC 2004)*.

Bodenreider, Olivier. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Resarch* 32 (Database Issue).

Buetow, Kenneth H. 2005. Cyberinfrastructure: Empowering a ”third way” in biomedical research. *Science* 308.

Bttcher, Stefan, Charles L. A. Clarke, and Gordon V. Cormack. 2004. Domain-specific synonym expansion and validation for biomedical information retrieval. In *2004 Text Retrieval Conference (TREC 2004)*.

CHAUSSABEL, DAMIEN, and ALAN SHER. 2002. Mining microarray expression data by literature profiling. *Genome Biology* 3.

CHINCHOR, NANCY. 1998. Overview of muc-7. In *In Proceedings of the 7th Message Understanding Conference*.

COHEN, KEVIN B., and LAWRENCE HUNTER. 2004. Natural language processing and systems biology. In *Artificial Intelligence Methods and Tools for Systems Biology*, ed. by W. Dubitzky and F. Azuaje, 147–173. Kluwer Academic Publishers.

COLLINS, FRANCIS S., ERIC D. GREEN, ALAN E. GUTTMACHER, and MARK S. GUYER. 2003. A vision for the future of genomics research. *Nature* 422.12.

COLOSIMO, MARC, ALEXANDER MORGAN, ALEXANDER S. YEH, JEFF B. COLOMBE, and LYNETTE HIRSCHMAN. 2005. Data preparation and interannotator agreement: Biocreative task 1b. *BMC Bioinformatics* 6 (Supplement I).

CONSORTIUM, THE GENE ONTOLOGY. 2000. Gene ontology: tool for the unification of biology. *Nature Genetics* 25.

CRIM, JEREMIAH, RYAN MCDONALD, and FERNANDO PEREIRA. 2005. Automatically annotating documents with normalised gene lists. *BMC Bioinformatics* 6 (Supplement I).

DICKMAN, STEVEN. 2003. The challenges of searching the scientific literature. *PLoS Biology* 1.

FELLBAUM, CHRISTIANE (ed.) 1998. *WordNet:An Electronic Lexical Database*. MIT Press.

FUNDEL, KATRIN, DANIEL GUTTLER, RALF ZIMMER, and JOANNIS APOSTOLAKIS. 2005. A simple approach to protein name identification: prospects and limits. *BMC Bioinformatics* 6 (Supplement I).

HACHEY, BEN, HUY NGUYEN, MALVINA NISSIM, BEA ALEX, and CLAIRE GROVER. 2004. Grounding gene mentions with respect to gene database identifiers. In *BioCreative Workshop Handouts*, Granada, Spain.

HANISCH, DANIEL, KATRIN FUNDEL, HEINZ-THEODOR MEVISSEN, RALF ZIMMER, and JULIANE FLUCK. 2005. Prominer: rule-based protein and gene entity recognition. *BMC Bioinformatics* 6 (Supplement I).

HERSH, WILLIAM, and RAVI TEJA BHUPATIRAJU. 2003. Trec genomics track overview. In *In Proceedings of the the Twelfth Text Retrieval Conference (TREC 2003)*.

HIRSCHMAN, LYNETTE, MARC COLOSIMO, ALEXANDER MORGAN, and ALEXANDER S. YEH. 2005a. Overview of biocreative task 1b: normalized gene lists. *Journal of Bioimedical Informatics* 37.

——, Alexander S. Yeh, Christian Blaschke, and Alfonso Valencia. 2005b. Overview of biocreative: critical assessment of information extraction for biology. *Journal of Bioimedical Informatics* 37.

Lander, E., L. Linton, B. Birren, C. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. Fitzhugh, R. Funke, D. Gage, K. Harris, A. Heaford, J. Howlan, L Kann, J Lehoczky, R Levine, P McEwan, K. McKernan, J. Meldrim, J. P. Mesirov, C. Miranda, W. Morris, J. Naylor, C. Raymond, M. Rosetti, R. Santos, A. Sheridan, C. Sougnez, N. Stange-Thomann, N. Stojanovic, A. Subramanian, D. Wyman, and J. Rogers. 2001. Initial sequencing and analysis of the human genome. *Nature* 409.

Maglott, Donna, Jim Ostell, Kim D. Pruitt, and Tatiana Tatusova. 2005. Entrez gene: gene-centered information at ncbi. *Nucleic Acids Resarch* 33 (Database Issue).

McCallum, Andrew, 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

McDonald, R., and F. Pereira. 2005. Identifying gene and protein mentions in text using conditional random fields. *Journal of Bioimedical Informatics* 37.

Morgan, Alexander, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh, and Jeff B. Colombe. 2004. Gene name identification and normalisation using a model organism database. *Journal of Bioimedical Informatics* 37.

Nigam, K., J. Lafferty, and A. McCallum. 1999. Using maximum entropy for text classification. In *In IJCAI-99 Workshop on Machine Learning for Information Filtering*.

Porter, Martin. 1980. An algorithm for suffix stripping. *Program* 14.

Smith, L., T. Rindflesch, and W.J. Wilbur. 2004. Medpost: a part-of-speech tagger for biomedical text. *Bioinformatics* 20.

Tamames, Javier, and Alfonso Valencia. 2006. Mining microarray expression data by literature profiling. *Genome Biology* 7.

Tanabe, Lorraine, Natalie Xie, Lynne H Thom, Wayne Matten, and W John Wilbur. 2005. Genetag: a tagged corpus for gene/protein named entity recognition. *Journal of Bioimedical Informatics* 37.

Wellner, Ben. 2006. Weakly supervised learning methods for improving the quality of gene name normalisation data. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature and Ontologies and Databases:Mining Biological Semantics*, Detroit.

WHEELER, DAVID L., DEANNA M. CHURCH, RON EDGAR, SCOTT FEDERHEN, WOLFGANG HELMBERG, THOMAS L. MADDEN, JOAN U. PONTIUS, GREGORY D. SCHULER, LYNN M. SCHRIML, EDWIN SEQUEIRA, TUGBA O. SUZEK, TATIANA A. TATUSOVA, and LUKAS WAGNER. 2004. Database resources of the national center for biotechnology information: update. *Nucleic Acids Resarch* 32 (Database Issue).

YEH, ALEXANDER S., MARC COLOSIMO, ALEXANDER MORGAN, and LYNETTE HIRSCHMAN. 2005. Biocreative task 1a: gene mention finding evaluation. *BMC Bioinformatics* 6 (Supplement I).

——, ALEXANDER MORGAN, and LYNETTE HIRSCHMAN. 2003. Evaluation of text data mining for database curation: lessons learned from the kdd challenge cup. *Bioinformatics* 19 (Supplment 1).