

## SUREGENE, A SCALABLE SYSTEM FOR AUTOMATED TERM DISAMBIGUATION OF GENE AND PROTEIN NAMES

RAF M. PODOWSKI

*Karolinska Institutet  
SE-171 77 Stockholm  
rpodowski@cmmt.ubc.ca*

JOHN G. CLEARY

*Reel Two, Ltd. and University of Waikato  
Innovation Park, Ruakura Rd  
Hamilton, New Zealand  
jcleary@reeltwo.com*

NICHOLAS T. GONCHAROFF

*Reel Two, Inc.  
2255 Van Ness Avenue, San Francisco, CA 94109  
nicko@reeltwo.com*

GREGORY AMOUTZIAS

*AstraZeneca, R&D  
35 Gate House Road, Waltham, MA 02451  
gregory.amoutzias@astrazeneca.com*

WILLIAM S. HAYES

*AstraZeneca, R&D Boston  
35 Gate House Road, Waltham, MA 02451  
william.s.hayes@astrazeneca.com*

Received 31 August 2004

Revised 3 January 2005

Accepted 7 January 2005

Researchers, hindered by a lack of standard gene and protein-naming conventions, endure long, sometimes fruitless, literature searches. A system that is able to automatically assign gene names to their LocusLink ID (LLID) in previously unseen MEDLINE abstracts is described. The system is based on supervised learning and builds a model for each LLID. The training sets for all LLIDs are extracted automatically from MEDLINE references in the LocusLink and SwissProt databases. A validation was done of the performance for all 20,546 human genes with LLIDs. Of these, 7344 produced good quality models (F-measure  $>0.7$ , nearly 60% of which were  $>0.9$ ) and 13,202 did not, mainly due to insufficient numbers of known document references. A hand validation of MEDLINE documents for a set of 66 genes agreed well with the system's internal

accuracy assessment. It is concluded that it is possible to achieve high quality gene disambiguation using scaleable automated techniques.

*Keywords:* Text mining; WCL; document classification; supervised learning; term disambiguation; word sense disambiguation.

## 1. Introduction

Biological researchers are constantly hindered in their work by a lack of standard naming conventions for genes and proteins. Near-frivolous choices of gene synonyms result in gene names like “IT” “midget”, or “ER”. These inherently ambiguous names cannot be effectively filtered by current search tools, nearly all of which are based on keyword queries. As a result, researchers must endure long, and sometimes fruitless, searches for literature about genes or proteins. Automated disambiguation of gene and protein names could significantly help improve access to biological literature and increase the efficiency of text analytics in the biomedical domain.

We present a system, called SureGene, for performing automated term disambiguation that can easily scale to tens of thousands of unique gene and protein names. SureGene uses a combination of machine learning and natural language processing technologies to identify abstracts relevant to specific genes and return these results as a ranked list.

Over 20,000 human genes have been identified in LocusLink and over 100,000 different names have been used to refer to them. A gene disambiguation system that is truly useful to a wide range of researchers must address some key, heretofore unsolved, challenges:

- it must scale to cover tens of thousands of genes and proteins per organism;
- it must be able to automatically generate training data with minimal human intervention;
- it must be able to make use of quantities of training data ranging from a few paragraphs to tens of thousands of paragraphs;
- it must be able to make use of low-quality training data that hasn't been annotated or enhanced with meta-data; and
- it mustn't rely on a comprehensive list of all possible gene and protein synonyms, since creating such a list is impractical.

SureGene addresses these gaps using a supervised learning system. This article presents test results that show SureGene is capable of accurately distinguishing between highly ambiguous gene terms, as well as between synonymous gene and non-gene terms.

## 2. Background

### 2.1. *Problem overview*

The absence of an automated approach for resolving ambiguity between gene synonyms is a key problem.<sup>1,2</sup> Further, text analytics in the biomedical domain are

dependent upon good gene name tagging and disambiguation. Natural Language Processing, in particular, is dependent upon term disambiguation, which has been called the “great open problem” of natural language lexical analysis.<sup>3</sup> In the biomedical domain, gene and protein name disambiguation is essential for providing quality protein-protein interactions, disease associations, and other complex biomedical analysis. This problem can also have a substantial impact on the efficiency of information retrieval methods, such as biomedical thesauri<sup>4</sup> or molecular pathway identification.<sup>5</sup>

Disambiguation tasks fall into two basic categories: determining if a term refers to a gene or gene product (does “PI” refer to “glutathione transferase” or “Permeability Index”); and identifying the true meaning of a synonymous gene name or abbreviation (does “PI” refer to “glutathione transferase” or “alpha-1-antitrypsin”). Both of these problems often elude keyword searches.

## **2.2. Previous work**

Natural language researchers began focusing on automated approaches to term disambiguation in the late 1980s and early 1990s. Yarowsky<sup>6</sup> used statistical models built from entries in Roget’s Thesaurus to assign sense to ambiguous words in text, using a Bayesian model to weight the importance of words related to the targeted ambiguous term. Gale, Church and Yarowsky<sup>7</sup> outlined an approach that used the 50 words preceding and following the target term to define a context for that term’s sense. In developing a method for general word sense disambiguation using unsupervised learning, Yarowsky<sup>8</sup> took a document classification approach to solving the problem of general term disambiguation. He also showed in this study that generic English language terms often have only one sense per co-location with neighboring words.

Around the year 2000, computational linguists and computational biologists began to look at term disambiguation in the biomedical domain. A number of researchers<sup>4,9,10</sup> have proposed solutions that involve manually crafted rules to help natural language processing and information retrieval systems correctly process ambiguous synonyms. These rules are often combined with supervised learning methods (in which systems are provided with human-curated training data) and in some cases unsupervised learning methods (also often referred to as “clustering”).

Recent work by Yu and Agichtein<sup>11</sup> compared four different approaches to solving the disambiguation problem — manual rules, fully supervised learning, partially supervised learning and unsupervised. The manual method is then combined with several of the machine learning approaches to yield a system capable of extracting synonymous genes and proteins from biomedical literature. Liu *et al.*<sup>1</sup> also explore a partially supervised learning approach based on disambiguation rules defined in the Unified Medical Language System. In the case of both papers, results are promising, but the systems require a pre-existing set of hand-crafted corpora, raising questions about scaling up to a level where a significant portion of human genes and proteins

can be covered. Hatzivassiloglou, Duboue and Rzhetsky<sup>5</sup> apply machine learning to the problem of gene, protein and RNA in text, showing that accuracy levels, as defined by F-measure, of nearly 85% can be attained for classifying terms as belonging to the class of gene or protein. Note, however, that the problem they have tackled is simpler than the one reported here, which seeks to identify the specific gene referred to.

### **3. Methods**

#### **3.1. Data collection**

Individual genes included in SureGene are defined by the LocusLink (LL)<sup>12,13</sup> human gene set. Gene names, symbols and synonyms were collected from LL and SwissProt (SP)<sup>14</sup> databases. The system is designed to query and recognize gene or gene product context in MEDLINE abstracts.

The contextual information for the AG-vs-NG model was collected from Medline as discussed in Sec. 3.4. The contextual information for the G-vs-OG models are collected from LocusLink and SwissProt Medline references as well as gene/protein descriptions from the preceding databases.

#### **3.2. Disambiguation strategy**

The disambiguation process, shown in Fig. 1, outlines the steps followed to decide if a document contains a reference to a particular LLID. To begin, a gene and the corpus of documents to be searched are selected (e.g. MEDLINE). The first step of the disambiguation process begins with the documents being classified by a model that decides if a document genuinely refers to genes or gene products. This is called the “AllGenes vs NotGene” (“AG-vs-NG”) model. If the document is classified as a NotGene, it is rejected and not considered further. The next step is a classification of each retained document with a model that is specific to the selected LLID. This is called a “Gene vs OtherGene” (“G-vs-OG”) model. If the model classifies the document as “Gene” then it is accepted as having a reference to the LLID.

Section 3.4 describes the process of building the “AG-vs-NG” model and the 20,546 “G-vs-OG” models. Section 3.5 describes how the accuracy of these models was validated. Section 5.1 discusses the process of selecting the set of names for each gene.

#### **3.3. Machine learning classification system**

The classification models were all built using Reel Two’s proprietary Classification System.<sup>15</sup> The decision to use this system was made for a number of reasons including the accuracy of the algorithm. As well as a desire for good classification accuracy there was the need for the underlying algorithm to be able to scale to large numbers of categories (38,000 human Locus Ids or even 400,000 Locus Ids for different organisms), to deal with training documents with positive instances varying

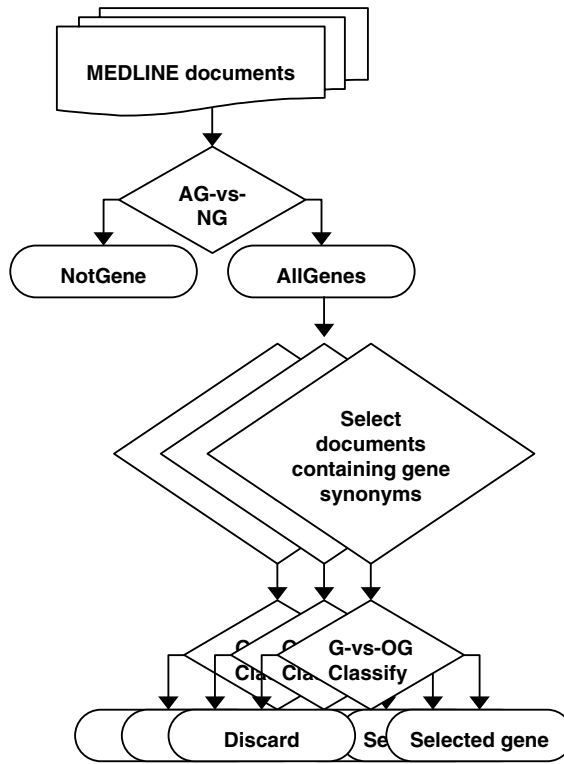


Fig. 1. Disambiguation scheme.

from 3 to hundreds of thousands of documents, and an even larger number of documents to be categorized (about 6 million). Within the set of training documents, there are about a million distinct words that were used as features. The important point to be made here is that it is possible, using extant Machine Learning technology, to meet these requirements. As improved algorithms and implementations are produced they can be substituted in this part of the process.

The traditional algorithms that have been used for this type of application are N ave Bayes (NB) and Support Vector Machines (SVM).<sup>16</sup> We experimented with using a number of different SVM implementations. On small subsets of the data SVM was able to achieve good accuracy. Unfortunately the large and uncertain memory requirements of the implementation we tried<sup>17</sup> coupled with the super-linear dependence of its execution time on the number of training instances means that it was infeasible to use more than a few hundred training instances. We note recent work that might make SVMs feasible for this application.<sup>18</sup>

We also experimented with a straightforward N ave Bayes algorithm but found that its accuracy was too low to be useful. This experience contradicts that reported elsewhere in the literature.<sup>16</sup> To get maximal accuracy, both N ave Bayes and SVM require an initial pass to select a feature set. During our initial investigations

this was felt to be too slow to be feasible. In retrospect, we could probably have engineered it to work, but given that the WCL system does not require it for good performance, we have not revisited this issue.

Given this experience, we decided to use the WCL algorithm. It is loosely based on *Näive Bayers* in that it makes use of the same “bag of words” statistics that *Näive Bayers* does. That is, it only keeps track of whether or not each word has occurred in a document of each class. An advantage of this is that, like NB, it can easily make use of leave-one-out (LOO) predictions for evaluating performance. The LOO procedure is used when predicting documents taken from the training set. First the statistics for the document are subtracted from the underlying word counts, then the prediction is done and finally the statistics are restored. The time for this is roughly the same as for adding a document to the statistics, so it is feasible to do it for all documents in the training set. LOO prediction is thus not biased by overfitting, that is, it is a true reflection of how new previously unseen documents are likely to be predicted by the system.

### 3.3.1. WCL

The WCL algorithm assigns a score  $S$  to each document  $D$  for category  $C$  as follows:

$$S = \sum_{w \in D} f(w, C), \quad (1)$$

where  $f(w, C)$  is some function of the statistics for the word  $w$ . The final score  $S$  is used for comparatively ranking different documents within one class, later we will deal with the issue of actually computing probabilities of membership. Both NB and SVM fit within this framework. For SVM, the values for  $f(w)$  are computed by an iterative relaxation algorithm. NB is formulated this way by taking  $f(w) = \log(P(w|C))$  where  $P(w|C)$  is an estimate of the probability of the word  $w$  being in a document given that it is in category  $C$ .

In describing the actual formulation of  $f(w, C)$  for WCL we will give a series of refinements. The first of these looks like an incorrect version of NB,  $f(w, C) = \log(P(C|w)) - \log(P(C))$ . ( $P(C|w)$  is an estimate of the probability that a document is in category  $C$  given that the word  $w$  is in the document.  $P(C)$  is an estimate of the probability that a document is in category  $C$ ). The intuition here is that  $f(w, C)$  is zero when  $P(C|w)$  and  $P(C)$  are the same, that is,  $w$  is uninformative. As is typical for such estimates we use the Dirichlet estimators  $P(C|w) = (n_{w,C} + 1/2)/(n_w + 1)$  and  $P(C) = (n_C + 1/2)/(n + 1)$ . ( $n$  is the total number of documents,  $n_w$  is the number of documents that the word  $w$  appears in and  $n_{w,C}$  is the number of documents that are in category  $C$  and which contain the word  $w$ .) Note that using these estimators  $\log(P(C|w)) = \log(n_{w,C} + 1/2) - \log(n_w + 1)$ .

Estimators of the form  $(n + \alpha)/(N + 1)$  are obtained by assuming a prior probability distribution of  $p^{\alpha-1}$  and computing the Bayesian estimate of the expected posterior probability. However, because we are using the  $\log(P(C|w))$  in  $f(w, C)$  what we should strictly do is estimate the expected logarithm of the

probability rather than the probability itself. This gives an estimator of the form  $\psi(n + \alpha) - \psi(N + 1)$  replacing  $\log$  by  $\psi$ . ( $\psi(x)$  is the digamma function).<sup>9</sup> Now we get  $f(w, C) = \psi(n_{w,C} + 1/2) - \psi(n_w + 1) - \psi(n_C + 1/2) + \psi(n + 1)$ .

This formulation gave a significant performance improvement over the simple logarithmic form. However, like NB and SVM we found that selecting a feature set of words was necessary in order to get the best performance. This is unsatisfying because adding more information (that is the statistics for words outside the feature set) should not degrade performance. One problem that was apparent was that words that occurred very seldom, say once or twice, could have high values for  $f(w, C)$  and were contributing unduly to the final scores. To reduce the contribution of such words, we formulated a function  $\sigma(w, C)$  which estimates the standard deviation of  $f(w, C)$ . Then the score can be reformulated as:

$$S(D, C) = \sum_{w \in D} f(w, C) / \sigma(w, C). \quad (2)$$

The expected value of the standard deviation of the logarithm of the probabilities is  $\sigma(w, C) = \psi_1(n_{w,C} + 1/2) - \psi_1(n_w + 1) - \psi_1(n_C + 1/2) + \psi_1(n + 1)$ . ( $\psi_1$  is the first polygamma function that is the derivative of  $\psi$ ).

This formulation gave a further significant improvement to performance. However, another issue arose that the scores were dependent on the size of the documents. That is, a document with many words often gave a much larger score. This became an issue when some of the documents we were working with were short abstracts and others were full academic papers. To correct for this, the score was normalized to allow for the length using:

$$N(D, C) = \sum_{w \in D} 1 / \sigma(w, C), \quad (3)$$

and then setting the final corrected score to be

$$R(D, C) = S(D, C) / N(D, C). \quad (4)$$

It is this formulation that was used in SureGene.

The score obtained above only ranks documents within a particular category it makes no decision about actual membership in the category. WCL does this by first collecting the LOO prediction score for each training document. This allows the number of incorrect decisions to be calculated for each possible setting of the threshold. The actual threshold is chosen as the *breakeven* point where the number of false positives equals the number of false negatives and where the precision and recall are the same.

### 3.3.2. Feature set creation

Words are extracted using a standard white space tokenizer. This treats all white space characters such as space, tab, end of line etc. as separators between words. Punctuation at the ends and beginnings of words is removed. Simple stemming is

done by dropping endings such as “ed”, “est”, and plurals. Also, 280 high frequency function words such as “the”, “a” etc. are deleted. Many learning systems also do a feature selection step where a subset of the words are used in the calculation. WCL does not do this, it uses all the words other than those in the stop list.

### 3.3.3. WCL benchmarking

We have performed extensive benchmarking of WCL, including tests on standard datasets used by both industry and academia. Chief among these is the Reuters-21578 set of documents, a corpus that is widely used to evaluate the performance of text categorization systems. This contains a set of short newspaper articles in 10 different categories. Table 1 compares recall and precision rates of this system with the results published for Reuters experiments on three other systems: a rule-based tree ensemble,<sup>20</sup> a Bayesian Network<sup>21</sup> with EM-fitting procedures, and kNN.<sup>22</sup> For clarity, the best recall rates in the table are underlined and the best precision rates are shown in bold for each class. These results show WCL delivers the best precision in all cases and the best recall in the majority of cases.

Table 2 compares the mean of the recall and precision (used because this is the way the SVM results were reported in Hearst *et al.*<sup>23</sup> with the optimized results published for Reuters experiments on four other systems: FINDSM, the Naïve Bayes algorithm, Bayes Nets, a version of SVM (all from Hearst *et al.*<sup>23</sup>) with our values for WCL. For clarity, the best result for each category is shown in bold. WCL delivers the best average in all but one case.

### 3.4. “AG-vs-NG” model

The “AG-vs-NG” classifier is an initial screen designed to eliminate documents that are clearly not about genes. For example, this filter should eliminate documents where the symbol “AR” refers to the gas argon, autoregressive model, acrosome reaction, etc., and retain any referring to the genes androgen receptor, aldose reductase, amphyregulin or any other gene-related content.

Table 1. WCL compared with published algorithms on Reuters 21578 data set.

Category	Docs	Trees		Bayers-EM		kNN		WCL	
		Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec
acq	719	96	95	93	93	<u>100</u>	91	97	<b>97</b>
com	56	<u>98</u>	82	50	52	35	76	<u>98</u>	<b>94</b>
crude	189	93	85	81	81	<u>96</u>	70	95	<b>96</b>
earn	1087	<u>99</u>	97	96	96	95	92	95	<b>98</b>
grain	149	95	92	71	69	82	75	<u>98</u>	<b>97</b>
interest	131	65	83	58	58	80	71	<u>98</u>	<b>94</b>
money-fx	179	77	76	73	73	92	65	<u>98</u>	<b>91</b>
ship	89	76	86	84	84	85	77	<u>98</u>	<b>96</b>
trade	117	81	70	66	65	89	66	<u>95</u>	<b>97</b>
wheat	71	<u>97</u>	83	68	70	69	73	95	<b>99</b>

Table 2. F-Measures for WCL versus published algorithms from Hearst *et al.*<sup>23</sup> on Reuters 21578 data set.

Category	FINDSM	NäiveBayers	BayersNets	LinearSVM	WCL
acq	65	88	88	94	<b>97</b>
corn	48	65	76	90	<b>96</b>
crude	70	79	80	90	<b>95</b>
earn	93	96	96	<b>98</b>	96
grain	67	79	81	95	<b>97</b>
interest	63	65	71	78	<b>96</b>
money-fx	47	57	59	74	<b>94</b>
ship	49	85	84	86	<b>97</b>
trade	65	64	69	76	<b>96</b>
wheat	69	70	83	92	<b>97</b>

### 3.4.1. Training set selection

A pool of training documents for the AG-category was obtained by searching MEDLINE for articles containing one or more terms suggestive of gene or gene product context. Using a query composed of the terms “gene”, “genes”, “cDNA” and “mRNA”, we obtained 672,675 documents containing one or more of the terms in the title or abstract.

The NG-category training set document pool comprised MEDLINE documents that had at least 500 characters of text and did not contain terms from a stoplist. This stoplist included terms such as: “gene”, “protein”, “cDNA”, “mRNA”, “kinase”, “receptor”, “amino acid”, “encode”, “subunit”, “express”, “pathway”, “repress”, “inhibit”, “transcript”, “oncogene” and “oncoprotein” as well as plurals and other variants. This gave a final pool of 4.5 million documents. The AG and NG category document pools were then used to obtain random subsets for the final classifier training sets (see Secs. 3.4.3 and 3.5).

The AG set of documents were hand-curated to further clean the set of abstracts based on the abstracts that were most like the NG model using the initial AG-vs-NG model. The NG model was reviewed and found to be of sufficient accuracy that no further steps were necessary. The bias between the general nature of the NG training set and the focused nature of the AG training set means that the FN rate for the AG documents was 0.996 (at the point of FN = FP) which is an exceedingly accurate model especially given the ratio of documents in Medline estimated to be 1:2.5 as seen in Sec. 3.4.3.

### 3.4.2. Training set bias

When selecting training documents, a bias will be introduced if the proportions of positive and negative examples in the categories do not represent the “real world” data distribution. The result of this bias is that suboptimal decisions are made about which categories the document belongs to. The classification system chooses

category membership at the “break-even point” where the number of False Positives (FP) is predicted to equal the number of False Negatives (FN).

There are two possible remedies for this problem. Both require an estimation of the correct “real-world” document ratios. The first method involves changing the prediction threshold. This threshold will differ from the default break-even point because of the training bias. An alternative and preferred method is to use the correct document ratios in the training sets. This will improve the accuracy of the result since at the break-even region for the threshold there will be roughly equal numbers of FP and FN documents from each category, unlike the first method. Thus the noise or error in the region will be minimized. We adopted the latter approach.

### 3.4.3. *Model parameters*

The “AG-vs-NG” model was prepared with 175,000 training documents. Based on an analysis of real-world data distribution, a training data ratio of 1:2.5 was selected, resulting in 50,000 AG and 125,000 NG training documents. This ratio selection was based on keyword screening of random sets of 10,000 MEDLINE documents with abstracts (45% of all MEDLINE records do not contain an abstract) and an independent classification of the same set with a number of AG vs. NG models. To validate this estimate, groups of 10,000 randomly sampled MEDLINE articles were categorized by the “AG-vs-NG” model several times, each time varying the model bias slightly. Despite the changes in the bias, the AG:NG ratios only ranged between 1:2.41 and 1:2.68. Because the bias change had little effect on the data distribution, it was decided a final ratio of 1:2.5 is reasonable.

## 3.5. *Individual gene models: “G-vs-OG”*

Individual “G-vs-OG” models are intended to recognize and prioritize documents with context matching a specific gene, recognizing and eliminating those documents with context matching that of other, ambiguously named genes or non-gene entities that evaded the “AG-vs-NG” filter. There are 20,546 “G-vs-OG” models, one for each human LLID. The classifier is expected to realize that an abbreviation such as “ER” referring to “Endoplasmic Reticulum” is not, for example, the desired target “Estrogen Receptor”, even when it occurs in a gene-context abstract.

### 3.5.1. *Training set selection*

A number of possible training set sources were considered. Information linking MEDLINE documents to genes through keyword searching, MeSH,<sup>24</sup> SP and LL databases was evaluated. MeSH linkage turned out to be too non-specific, with a large fraction of documents containing no actual mention of a given gene in the title or abstract. (MeSH headings appear to better define the contents of a full text

document.) Gene name-based keyword searching, even with phrase searching capabilities, cannot be relied on to automatically supply a quality set for more than a handful of genes. Therefore, we chose to use the SP and LL MEDLINE references to compile training set documents for the initial gene models. Documents referencing more than 20 individual genes were excluded, as they most often represent large scale sequencing projects, rather than discussing individual genes. Finally, functional description texts for each gene from LL and SP were added to each gene's positive category (G) training set.

Contextual disambiguation must be supported with sufficient training data. A gene has to have between 5–10 literature references to become sufficiently accurate for general use. Only a fraction of the genes and proteins found in LocusLinks and SwissProt have at least that much training data. However, this system for disambiguation can take advantage of ever increasing amounts of training data for more accurate results.

For the negative category (OG) in the “G-vs-OG” model, we relied on a combination of random AG-category documents (excluding any overlaps with G documents) and documents with known name or symbol ambiguities to the gene in the G category. For example, the androgen receptor gene (LLID 367) is frequently referred to by the symbol “AR”. “AR” is also known to refer to the “aldose reductase” (LLID 231) and “amphiregulin” (LLID 374) genes. The OG-category training set would then include SP and LL references to the latter two genes.

### 3.5.2. *Training set bias*

To accurately set the “real-world” proportions of documents in the G and OG categories an estimate is needed of the proportions of documents that are: associated with the gene; associated with other known genes, and; those associated with unlisted genes or with terms that are not genetic. Because of the large numbers of models that must be built it is impossible to require human intervention for this step.

There is also the possibility that some documents do not refer to any of the synonymous genes, but might instead refer to an unknown gene or to another non-genetic meaning (for example, “ER” can refer to “Endoplasmic Reticulum” or “Emergency Room”). The AG-vs-NG filter eliminates most documents where a gene name synonym refers to a non-gene subject and no other gene. To account for all the above possibilities, the OG-category training set included an equal number of documents to those in the G-category, but chosen at random from the AG-category set (excluding any that were already in the G-category). It has yet to be confirmed that this is the correct number of such documents to include. This is discussed further in the results section.

To recapitulate, the training sets were chosen as follows. For the G-category, the training set documents comprised gene-specific MEDLINE references from SP and LL databases. For the OG-category, a set of documents equal in number to the

G-category set were chosen at random from the complete AG-category document pool (taking care to exclude any that already occurred in G). This OG-category training set also included documents with SP and LL references to known, ambiguously named genes. The preceding assumptions give a ratio of  $1:1 + x$  for the G:OG categories, where  $x$  is determined by the number of other synonymous genes included in category OG.

The effect of a bias between the training sets and the “real-world” proportions becomes smaller the more accurate the classifier is. Consider, for example, a classifier that is perfect, that is, the true and false instances are completely separated by the threshold point generated by the classifier. The break-even point will be set between the lowest-scoring true instance and the highest-scoring false instance. This will work regardless of the real world data distribution and thus for a perfect classifier, bias does not matter. So one way of compensating for the crude initial estimates used here is to accumulate more data and to improve the performance of the classifier. Methods for doing this are discussed in Sec. 5.

### 3.5.3. *Model parameters*

SureGene requires a specialized model for each gene. Thus the training set size and composition differs for each gene model.

## 3.6. *Validation document set collection and markup*

Validation of our approach was done for a total of 66 genes (Tables 2 and 3). Twenty genes were selected based on the inherent ambiguity of at least one of their commonly used symbols in regard to other genes as well as to non-gene acronyms and English words (see the second column of Table 2). For each of these 20 LLIDs, a subset of MEDLINE documents containing the selected ambiguous symbol was collected (excluding any that were used to construct the training sets). Each of these was marked up by hand as one of the following: having the ambiguous symbol referencing a specific gene in human LL; another gene not in human LL or having unexpected usage based on LL gene symbol information; or as referring to a non-gene entity. The documents were then categorized with the appropriate “G-vs-OG” classifiers. In the case of the ambiguous symbol “AR”, the validation was performed for three individual genes based on one set of appropriately marked up documents. Performance of each model was evaluated against the LOO validation (see Sec. 4.2).

Models for 46 members of the human Nuclear Receptor (NR) gene family (Table 3) were similarly evaluated, with an additional screening by the “AG-vs-NG” model. For each NR gene, MEDLINE documents matching any one of the LL-defined gene symbols were selected and processed with the disambiguation algorithm (Fig. 1). Accuracy of each gene model was estimated by human examination of all — or in some case the first several hundred top-scoring — documents in the G and OG categories.

Table 3. AG vs. NG model performance.

Model	Automatic		Partially curated	
	allgenes	notgene	allgenes	Notgene
Total documents	50,000	125,000	49,493	125,108
F-Measure	0.990	0.922	0.996	0.924
FP = FN	519	9689	185	9490

## 4. Results and Validation

### 4.1. “AG-vs-NG” model

The automatically generated “AG-vs-NG” model’s performance is summarized in Table 3. The AG-category accuracy is 99.0% at the break-even point, based on a LOO validation. Partial hand curation of the model increased the LOO validation accuracy to 99.6%. Any document with a below-threshold score in the AG category is treated as having a non-gene context regardless of the NG category score. It is important to note that the majority of NG documents in the False Negative (FN) group are not classified as AG, but as “other”, that is not fitting either the AG or NG categories.

### 4.2. “G-vs-OG” model

Models for 20,546 LocusLink human genes were generated automatically. Of these, 4879 genes contained no references in the LL or SP databases and thus offered no training data to create models. Figure 2 shows the distribution of the training documents for all the genes, comparing the number of genes (LLIDs) against the number of G-category training documents. The distribution is heavily skewed toward low numbers of documents.

Despite this, Figs. 3 and 4 indicate it is possible to get consistent, good model performance for genes with sufficient training examples; usually five documents were

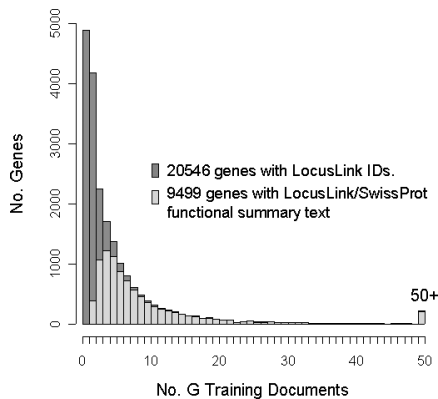


Fig. 2. Number of genes vs. number of training documents.

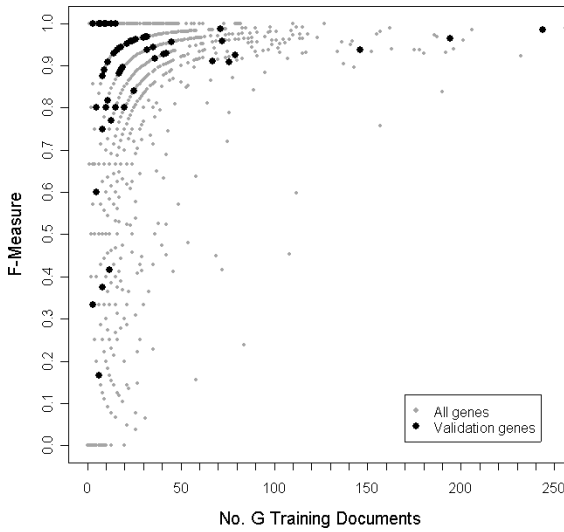
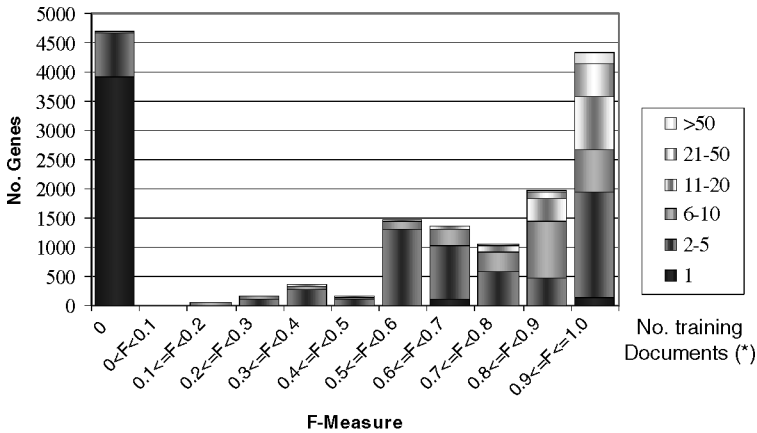


Fig. 3. F-Measure vs. number of G training documents. The patterned lines result from integer values of FP and FN.



(\*) The number of training documents represents MEDLINE documents only, and does not reflect the addition of LocusLink summaries for 6528 genes nor the SwissProt functional description text for 7119

Fig. 4. Effect of number of training documents on gene context predictive accuracy.

sufficient. Figure 3 plots the LOO F-measure for all 20,564 gene models against the number of G-category training documents. The bulk of the gene models have F-measures above 70% and very few models with more than 20 training documents fall below 70%. The 66 hand-validated genes are also highlighted, and as well seem representative of the overall distribution.

The chart in Fig. 4 shows a different view of the same data, with bars broken down by the number of documents in the G training set. It can be seen that almost all the poorer performing gene models have small numbers of training documents. In addition, more than 84% of gene models with more than five training documents have an accuracy of greater than 70%. For models with more than 10 training documents, this increases to 91% of the models.

Model accuracy was greatly improved by adding functional description texts from the LL and SP databases to the training data. There were 6528 genes with such information in LL and 7119 in SP. Overall, 9499 genes benefited from one or both sources. As shown in Fig. 5, the median F-Measure for the 9499 genes improved from 0.714 to 0.833, while the middle 50% of the data improved from a range of 0.4–0.846 to 0.667–1.0.

To ensure the training set contained documents that were indeed about genes, all 61,727 LL and SP gene reference documents were categorized with the “AG-vs-NG” model. The classification system predicted 58,241 (94.4%) documents as AG, while the partially hand-curated model (described in Sec. 4.1) indicated 57,631 (93.4%).

The addition of MeSH gene references to LL and SP increases the total gene model training set size from 61,727 to 1,090,641 individual documents. However, adding MeSH references did not increase, and sometimes decreased, accuracy. Figure 6 shows lines joining the LOO F-measures of the training sets with and without the MeSH-linked references. The majority of the lines trend horizontally or downward showing that MeSH-linked references have a negligible effect on the model performance for most genes with 10 or more SP and LL training documents, and can cause an occasional significant drop in accuracy. The decrease in accuracy is often due to non-gene context of the document or lack of references to the specific gene in the abstract. Training data based purely on MeSH references thus does not seem to be useful.

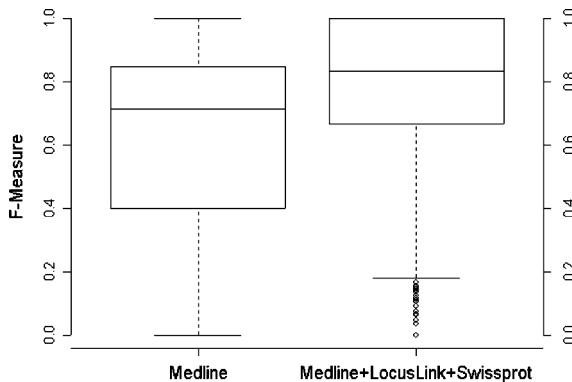


Fig. 5. Effect of LocusLink and SwissProt functional text on gene model performance.

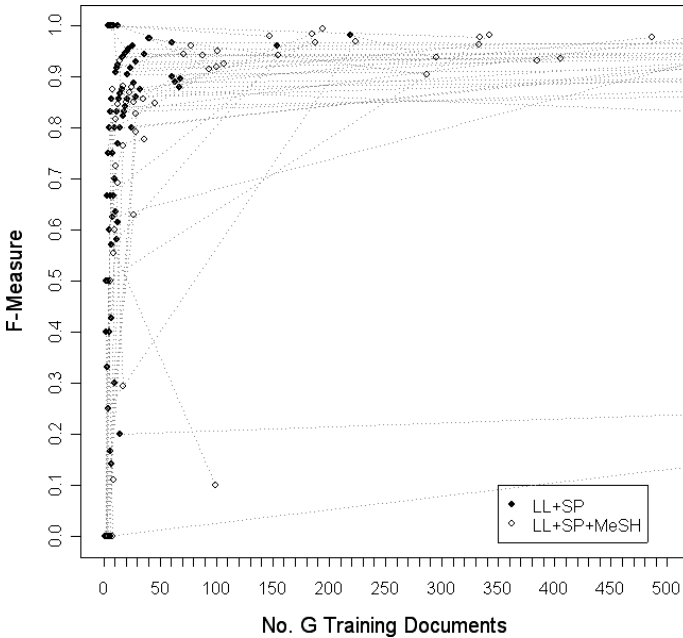


Fig. 6. Impact of MeSH references on model performance.

Initial model predictive performance, based on a LOO method, was compared to a real-life predictive performance for a set of 20 genes with highly ambiguous gene symbols (Table 4) and 46 genes from the human NR gene family (Table 5). F-Measure comparisons between the two methods are shown in Fig. 7 and the last two columns of Tables 2 and 3. Points below the diagonal in Fig. 7 indicate models where human validation showed higher performance than that suggested by the LOO evaluation. Points above the diagonal reflect instances where the classification system’s LOO estimate is more optimistic than the results obtained by human validation.

The general consistency between the LOO F-measures and the validated F-measures gives some confidence that the performance on the bulk of genes is actually reflected in the results in Figs. 3 and 4. Tables 4 and 5 also show that the ratios of G:OG in the training data differ markedly from the actual ratios in the validation sets. Despite this, the models still yield good performance.

One outlier in the ambiguous gene symbol data set is LLID 796 (symbol CT), which shows drastically worse performance on documents marked up as LLID 796 due to a large False Positive set. This is due to frequent use of the gene name “calcitonin-related polypeptide alpha” in non-gene, clinical context in a large fraction of documents. Strategies for overcoming this problem are presented in Sec. 5. Another poorly performing model is for the NR gene, NCOA5 (LLID 57727). This gene has known ambiguous gene symbols while there are only three training documents for this model — additional training documents are required.

Table 4. Validation performance on 20 genes selected for high ambiguity.

LocusLink ID	Ambiguous gene symbol	Model training documents		Marked up validation documents		Model predictions				Validation performance			Leave-one-out
		G	OG	G	Other	TP	FP	FN	TN	Prec	Rec	F-Measure	F-Measure
12	ACT	30	40	163	35	146	6	17	29	0.96	0.90	0.93	<b>0.97</b>
54	TRAP	18	139	120	6	106	0	14	6	1.00	0.88	0.94	<b>0.94</b>
231	AR	19	278	254	2306	215	5	39	2301	0.98	0.85	<b>0.91</b>	0.90
367	AR	244	276	1700	859	1615	25	85	834	0.98	0.95	0.97	<b>0.98</b>
374	AR	16	275	99	2461	98	49	1	2412	0.67	0.99	0.80	<b>0.94</b>
434	ASP	8	72	21	118	21	22	0	96	0.49	1.00	0.66	<b>0.88</b>
718	ASP	31	67	62	77	33	10	29	67	0.77	0.53	0.63	<b>0.97</b>
796	CT	36	48	172	1069	170	1019	2	50	0.14	0.99	0.25	<b>0.92</b>
847	CAT	24	27	282	318	279	21	3	297	0.93	0.99	<b>0.96</b>	0.96
948	FAT	35	61	56	38	55	0	1	38	1.00	0.98	<b>0.99</b>	0.94
1356	CP	26	25	313	24	304	3	9	21	0.99	0.97	<b>0.98</b>	0.96
1890	TP	25	59	216	102	212	5	4	97	0.98	0.98	<b>0.98</b>	0.84
2099	ER	194	196	468	7	455	5	13	2	0.99	0.97	<b>0.98</b>	0.96
2950	PI	79	145	149	153	141	53	8	100	0.73	0.95	0.82	<b>0.92</b>
3240	HP	18	30	512	4	497	3	15	1	0.99	0.97	<b>0.98</b>	0.94
4860	NP	18	41	43	25	36	0	7	25	1.00	0.84	0.91	<b>0.94</b>
5241	PR	45	48	438	26	427	10	11	16	0.98	0.97	<b>0.98</b>	0.96
5265	PI	67	145	100	202	65	6	35	196	0.92	0.65	0.76	<b>0.91</b>
6476	SI	7	20	117	16	117	4	0	12	0.97	1.00	0.98	<b>1.00</b>
7298	TS	32	56	137	12	131	1	6	11	0.99	0.96	<b>0.97</b>	0.97

Table 5. Validation performance on 46 nuclear receptor family genes.

LocusLink ID	Official gene symbol	Model training documents		Marked up validation documents			Model predictions				Validation performance			Leave-one-out
		G	OG	G	Other	TP	FP	FN	TN	Prec	Rec	F-Measure	F-Measure	
190	NR0B1	25	24	225	75	196	10	29	65	0.87	0.95	0.91	<b>0.96</b>	
367	AR	244	276	102	100	102	0	0	100	1.00	1.00	<b>1.00</b>	0.98	
2063	NR2F6	3	3	28	4	28	3	0	1	1.00	0.90	<b>0.95</b>	0.33	
2099	ESR1	194	196	45	102	40	2	5	100	0.89	0.95	0.92	<b>0.96</b>	
2100	ESR2	72	72	100	96	99	1	1	95	0.99	0.99	<b>0.99</b>	0.96	
2101	ESRRA	15	14	32	0	32	0	0	0	1.00	1.00	<b>1.00</b>	0.80	
2103	ESRRB	6	4	16	0	16	0	0	0	1.00	1.00	<b>1.00</b>	1.00	
2104	ESRRG	8	12	5	0	5	0	0	0	1.00	1.00	<b>1.00</b>	0.75	
2494	NR5A2	11	11	54	159	54	16	0	143	1.00	0.77	<b>0.87</b>	0.82	
2516	NR5A1	18	24	117	82	100	0	17	82	0.86	1.00	<b>0.92</b>	0.89	
2649	NR6A1	13	12	43	18	43	2	0	16	1.00	0.96	0.98	<b>1.00</b>	
2908	NR3C1	71	80	104	118	97	3	7	115	0.93	0.97	0.95	<b>0.99</b>	
3164	NR4A1	17	39	99	97	99	0	0	97	1.00	1.00	<b>1.00</b>	0.88	
3172	HNF4A	41	42	90	110	89	11	1	99	0.99	0.89	<b>0.94</b>	0.93	
3174	HNF4G	5	4	2	0	2	0	0	0	1.00	1.00	<b>1.00</b>	0.80	
4306	NR3C2	23	22	94	106	94	6	0	100	1.00	0.94	0.97	<b>0.99</b>	
4929	NR4A2	21	35	130	27	119	0	11	27	0.92	1.00	<b>0.96</b>	0.95	
5241	PGR	45	48	182	112	180	16	2	96	0.99	0.92	0.95	<b>0.96</b>	
5465	PPARA	42	43	104	43	82	18	22	25	0.79	0.82	0.80	<b>0.93</b>	
5467	PPARD	17	17	80	21	79	2	1	19	0.99	0.98	<b>0.98</b>	0.94	
5468	PPARG	146	155	146	0	100	0	46	0	0.69	1.00	0.81	<b>0.94</b>	
5914	RARA	31	35	111	87	94	4	17	83	0.85	0.96	0.90	<b>0.97</b>	

Table 5. (Continued)

LocusLink ID	Official gene symbol	Model training documents			Marked up validation documents			Model predictions				Validation performance			Leave-one-out	
		documents			documents			TP	FP	FN	TN	Prec	Rec	F-Measure	F-Measure	
		G	OG	G	Other	TP	FP	FN	TN	Prec	Rec	F-Measure	F-Measure			
5915	RARB	36	45	101	99	100	0	1	99	0.99	1.00	<b>1.00</b>	0.92			
6095	RORA	9	19	47	34	42	1	5	33	0.89	0.98	<b>0.93</b>	0.89			
6096	ROB	9	9	20	2	18	0	2	2	0.90	1.00	0.95	<b>1.00</b>			
6097	RORC	8	8	8	39	8	14	0	25	1.00	0.36	0.53	<b>0.75</b>			
6256	RXRA	30	32	116	0	99	0	17	0	0.85	1.00	0.92	<b>0.97</b>			
6257	RXRB	14	14	102	0	99	0	3	0	0.97	1.00	<b>0.99</b>	0.93			
6258	RXRG	5	5	94	0	92	0	2	0	0.98	1.00	<b>0.99</b>	0.80			
7025	NR2F1	12	18	86	0	52	0	34	0	0.61	1.00	<b>0.75</b>	0.42			
7026	NR2F2	6	59	43	45	24	0	19	45	0.56	1.00	<b>0.72</b>	0.17			
7067	THRA	20	27	148	54	97	3	51	51	0.66	0.97	0.78	<b>0.80</b>			
7068	THRB	32	32	128	82	99	0	29	82	0.77	1.00	0.87	<b>0.94</b>			
7101	NR2E1	5	38	37	33	35	5	2	28	0.95	0.88	<b>0.91</b>	0.60			
7181	NR2C1	7	38	50	49	47	13	3	36	0.94	0.78	0.86	<b>1.00</b>			
7182	NR2C2	10	31	33	101	33	1	0	100	1.00	0.97	<b>0.99</b>	0.80			
7376	NR1H2	8	22	25	158	24	60	1	98	0.96	0.29	0.44	<b>0.75</b>			
7421	VDR	76	99	101	99	100	0	1	99	0.99	1.00	<b>1.00</b>	0.91			
8431	NR0B2	2	2	52	172	52	73	0	99	1.00	0.42	0.59	<b>0.77</b>			
8856	NR1I2	18	72	100	100	100	0	0	100	1.00	1.00	<b>1.00</b>	0.89			
9572	NR1D1	8	27	35	45	19	0	16	45	0.54	1.00	<b>0.70</b>	0.38			
9970	NR1I3	17	72	99	101	99	1	0	100	1.00	0.99	<b>1.00</b>	0.88			
9971	NR1H4	15	72	98	102	98	2	0	100	1.00	0.98	0.99	<b>1.00</b>			
10002	NR2E3	10	10	14	102	14	4	0	98	1.00	0.78	0.88	<b>1.00</b>			
10062	NR1H3	11	12	107	2	100	0	7	2	0.94	1.00	<b>0.97</b>	0.91			
57727	NCOA5	3	10	12	180	12	108	0	72	1.00	0.10	0.18	<b>1.00</b>			

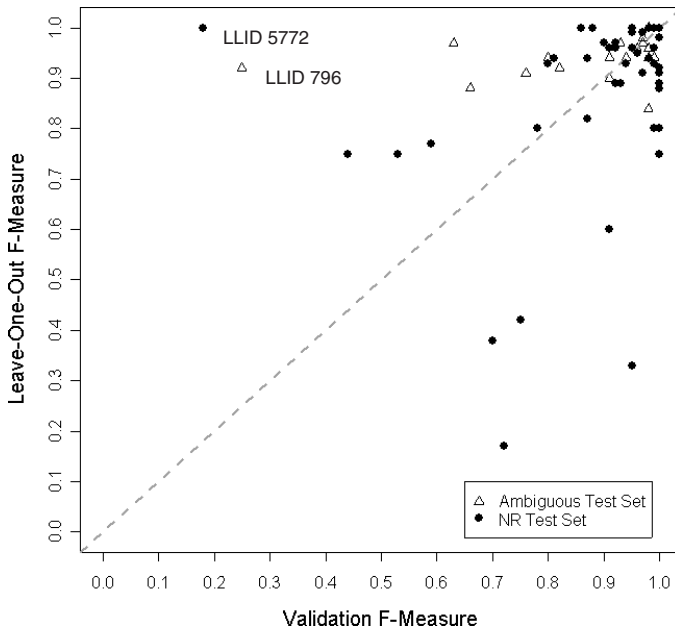


Fig. 7. Comparison of G vs. OG model performance.

#### 4.3. Kappa statistics for major pharmaceutically relevant gene families

The kappa statistic measures the agreement between document raters. In this particular case, one “rater” is the standard being used for training, the training documents for each gene referenced by LocusLink and SwissProt. The other “rater” is the G vs. OG gene model. Perfect agreement between the two raters results in a kappa statistic ( $K$ ) = 1.0. Agreement between the raters at the rate of random chance is  $K = 0.0$ . Values of  $K > 0.7$  are generally considered to be very good.  $K < 0$  indicates that disagreement between the model and the training dataset occurs more often than random chance.

Figure 8 shows the histogram of the number of genes vs. the Kappa statistic for all genes. Each bar of the histogram is segregated into sets where the training set sizes are within the range indicated by the legend. As shown, most of the genes with a Kappa statistic  $\leq 0.0$  have less than 10 training documents. Most of the genes with more than 10 training documents have a Kappa statistic  $> 0.7$ .

The Kappa statistic measure does not handle small decision sets well. This is why a significant number of the very small training set models have a Kappa statistic equal to 1.0. One may often get complete agreement by chance when reviewing small training sets (refer to Fig. 2). Correspondingly, complete disagreement (e.g.  $K < 0.0$ ) is more likely to occur by random chance with small decision sets.

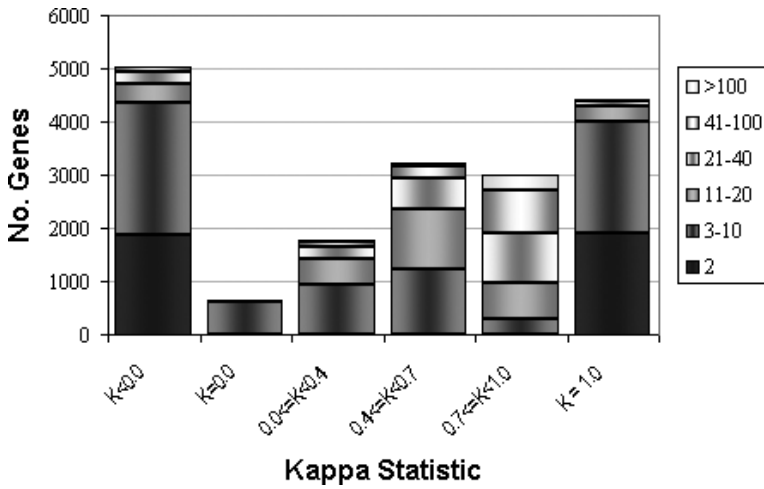


Fig. 8. Kappa statistic for all gene models.

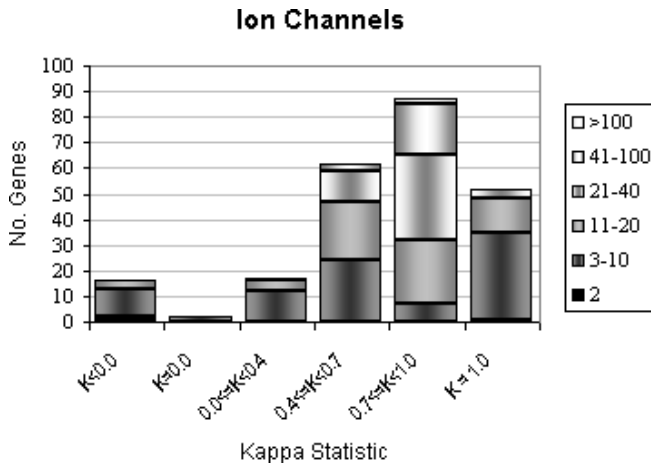


Fig. 9. Kappa statistic for Ion channel genes.

The gene families: G-protein coupled receptors (GPCR's), Kinases, Nuclear hormone receptors (NHR's) and Ion channels are important drug target genes. They comprise the majority of the targets for currently marketed drugs. As one can see from Figs. 9–12, the Kappa statistic peaks in the  $K = 0.7$ – $1.0$  range. These gene families are overall well-characterized and most members have more than 10 training documents which explains the better results than seen in Fig. 8.

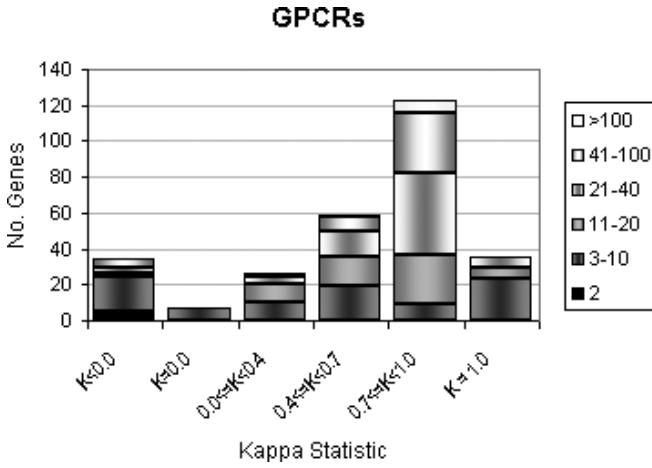


Fig. 10. Kappa statistic for GPCR genes.

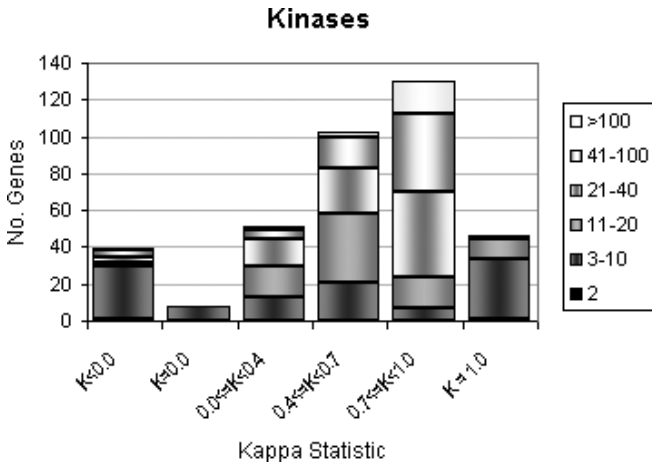


Fig. 11. Kappa statistic for Kinase genes.

## 5. Ongoing Work

### 5.1. Classifying all of MEDLINE

The first steps toward deploying the SureGene system have been taken. All 6.5 million PubMed documents have been scanned for each of the genes evaluated above. Of these 2 million passed the AGvsNG filter. Column 4 of Table 6 shows the number of documents predicted for each of the genes. As can be seen in nearly all cases, significantly more documents have been recovered than were available in the original training sets. Work is currently proceeding on evaluating the quality of the returned documents and on scaling to a much larger number of genes.

Classification of all of MEDLINE against the 63 validation genes required less than one hour using a dual 1.8 GHz machine with AMD Opteron processor and a

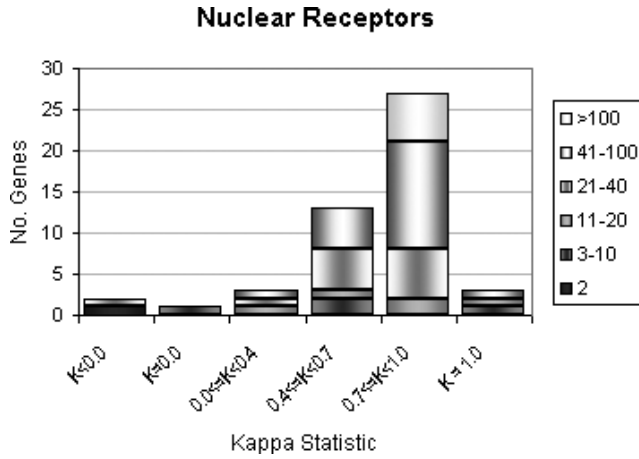


Fig. 12. Kappa statistic for nuclear receptor genes.

Table 6. Original training data vs. number of predicted documents.

LLID	Gene symbol	Training documents	Predicted documents	LLID	Gene symbol	Training documents	Predicted documents
12	ACT	70	366	5241	PR	93	2744
54	TRAP	157	2516	5265	PI	212	2306
190	NR0B1	49	102	5465	PPARA	85	1102
231	AR	297	946	5467	PPARD	34	166
367	AR	520	4782	5468	PPARG	301	1710
374	AR	291	248	5914	RARA	66	980
434	ASP	80	347	5915	RARB	81	688
718	ASP	98	6251	6095	RORA	28	174
796	CT	84	6665	6096	RORB	18	32
847	CAT	51	5422	6097	RORC	16	33
948	FAT	96	983	6256	RXRA	62	615
1356	CP	51	1926	6257	RXRB	28	167
1890	TP	84	546	6258	RXRG	10	121
2063	NR2F6	6	14	6476	SI	27	1016
2099	ER	390	3985	7025	NR2F1	30	103
2100	ESR2	144	1618	7026	NR2F2	65	39
2101	ESRRA	29	45	7067	THRA	47	482
2103	ESRRB	10	8	7068	THRB	64	177
2104	ESRRG	20	19	7101	NR2E1	43	10
2494	NR5A2	22	100	7181	NR2C1	45	49
2516	NR5A1	42	369	7182	NR2C2	41	39
2649	NR6A1	25	45	7298	TS	88	1095
2908	NR3C1	151	2697	7376	NR1H2	30	44
2950	PI	224	518	7421	VDR	175	1255
3164	NR4A1	56	40	8431	NR0B2	4	78
3172	HNF4A	83	267	8856	NR1I2	90	661
3174	HNF4G	9	54	9572	NR1D1	35	27
3240	HP	48	1740	9970	NR1I3	89	165
4306	NR3C2	45	1612	9971	NR1H4	87	110
4860	NP	59	221	10002	NR2E3	20	15
4929	NR4A2	56	34	10062	NR1H3	23	122
				57727	NCOA5	13	49

1024KB cache. Memory usage was less than 1 GB. Building an index of the results for the AG-vs-NG classification of all of MEDLINE took less than one day on the same machine.

#### 5.1.1. *Deployment of web-based service*

The process of building a web-based service has also been completed. This chose a subset of the 20,000 available Locus Ids which had 3 or more training documents and where the reported LOO F-measure was  $>0.5$  or greater. Some 8300 genes fit this category.

The earlier sections have focused on the accuracy of the AG-vs-NG and the G-vs-OG models. The primary role of these models is to reduce the number of (hopefully extraneous) documents returned. However, the set of names used for each gene will largely determine the set of documents which are then pruned by the G-vs-OG models. The process of building the complete system for the 8300 genes quickly showed that the sets of gene names provided in the LocusLink database is not comprehensive. If the names given in the database are taken literally then many documents listed as referencing the gene contain none of the names. There are a number of reasons for this:

- the gene may be referred to in the full text of the document but not in the abstract;
- the name may have been completely missed by the curators of the database; or
- the gene is referred to by a similar variant of the listed name.

We have focused on dealing with the last of these reasons. An example is LLID 2099 “estrogen receptor alpha”. One of the listed names for this is “ER-alpha” however the variants “ER alpha”, “ERalpha”, “ERA”, “hERalpha”, “hERA” are not listed but all occur in some documents. We use a set of transformation rules which take names like “ER-alpha” to these variants. Initial results from this approach are encouraging but we are actively continuing work on refining the set of rules and how they are applied.

The second of the reasons listed above may be able to dealt with by checking that all the training documents contain at least one gene name and, during curation, by checking documents that pass the G-vs-OG model but which do not contain any listed gene name.

## 5.2. *Enabling user curation*

This will enable users to significantly reduce time spent hunting for gene and, eventually, protein related literature. This internet-based service will be freely available to all users, and will cover public databases of biomedical literature, including MEDLINE. Users will be able to submit queries in the form of a canonical gene name or any of its synonyms, and will be returned a ranked list of documents about that gene. The service will also allow users to submit feedback regarding

accuracy. User will also be able to upload training examples to enable searches for genes which either have few training documents, or which are not currently covered by the system. This functionality will enable a “worldwide curation” function — submissions from users will continuously increase the amount of training data, and thus the scope and accuracy of the system.

With very large numbers of gene names, it is difficult to do any form of manual processing. One way that this can be alleviated is to allow users to provide curation of the data. Reel Two plans to launch a public website that gives gene literature search access to academic and non-profit research institutions and gathers user curation and feedback. Each entry returned by a query can be marked as being correct or incorrect. Users will likely be most motivated to mark incorrect entries, but marking previously unseen entries as correct will also help improve performance. All query result pages will have the ability to garner user curation. Figure 13 shows an example of a curation page, though this prototype does not include a facility for uploading one’s own training data.

The main problem with such capability is often a social one of motivating people to actually provide feedback. Motivating user feedback is most successful when it is one, easy and two, focused on delivering positive results immediately to the user. Both conditions are expected to be in place for the user feedback aspect of the system. Reel Two has found through work with clients and researchers such as those at the European Bioinformatics Institute and Flybase, that biomedical users, already used to lengthy literature searches, are willing to expend effort and provide feedback if it will quickly yield better search returns.

The feedback facility will be provided as part of the initial web based deployment on a small set of genes. This will enable early exploration of how to best offer the

<p>(66) PMID 10565838: <b>Binding, partial agonism, and potentiation of alpha(1)-adrenergic receptor function by benzodiazepines: A potential site of allosteric modulation.</b></p> <p>Benzodiazepines, a class of drugs commonly used to induce anesthesia and sedation, can attenuate intracellular calcium oscillations evoked by alpha(1)-adrenergic receptor (alpha(1)-AR) stimulation in pulmonary artery smooth muscle cells. We postulated a direct action of benzodiazepines in modulating alpha(1)-AR function at the receptor level. Benzodiazepines bound to each of the cloned alpha(1)-AR subtypes (alpha(1a)-, alpha(1b)-, or alpha(1d)-AR) on COS-1 cell membranes transiently transfected to express a single population of alpha(1)-AR subtype. The ability of benzodiazepines to alter alpha(1)-AR signal transduction was investigated by measuring total inositol phosphate generation in rat-1 fibroblast cells, stably transfected to express a single alpha(1)-AR subtype. By themselves, benzodiazepines displayed partial agonism. At alpha(1b)-ARs and alpha(1d)-ARs, the maximal inositol phosphate response to phenylephrine was potentiated almost 2-fold by either midazolam or lorazepam (100 microM). At alpha(1a)-ARs, diazepam, lorazepam, and midazolam all increased the maximal response of the partial agonist clonidine at these receptors, whereas the response to the full agonist phenylephrine was unaltered or inhibited. The potentiating actions of midazolam and its partial agonism at alpha(1)-ARs was blocked by the addition of 1 microM prazosin, an alpha(1)-AR antagonist, and not by a gamma-aminobutyric acid(A)-receptor antagonist. These studies show that benzodiazepines modulate the function of alpha(1)-ARs in vitro, and this is the first report of a potential allosteric site on alpha(1)-ARs that may be therapeutically useful for drug design.</p>	<ul style="list-style-type: none"> <li><input type="radio"/> Not Marked Up</li> <li><input type="radio"/> LLID 231</li> <li><input type="radio"/> LLID 367</li> <li><input type="radio"/> LLID 374</li> <li><input type="radio"/> Not Gene</li> <li><input type="radio"/> Unknown Gene</li> </ul>
---	---

Fig. 13. Prototype of the user curation interface.

facility to users. The goal is to motivate researchers to curate particular subsets of the data that they are particularly interested in. One problem that will be explored is how to handle the feedback that is reported. It is unclear whether it will be of sufficient quality to be entered into the training data un-moderated.

## 6. Discussion

The problems resulting from ambiguous gene and protein names have caused enormous difficulties in biomedical text mining as well as simple text searches for gene-related information. The algorithms presented here provide a scalable system for disambiguating gene and protein names for a variety of purposes. One can use it for improving text searches against the literature by tagging all potential gene names in the literature with their canonical forms. Much more accurate NLP systems for gene and protein relation extraction will be possible given accurate disambiguation.

The results show that when more than 20 abstracts per gene are available for training, accuracy of the system is mostly over 90%. Even with 5 documents we usually get significant enrichment of the search results. The system can easily be altered dynamically to provide greater precision or recall by altering the thresholds associated with the gene disambiguation models.

The next step (in process) is a developing a central, publicly available web service to allow researchers to access this system when searching the literature for specific genes or proteins. The users of the public system will be able to provide performance feedback and additional training data if the gene of interest has too little training data to yield accurate disambiguation results, or if the existing training data displays an unexpected bias (such as that found in LLID 796 as documented in Sec. 4.2). Although models will initially have small numbers of training documents, training can be quickly bootstrapped as users submit feedback on initial predictions. In this way, additional training data can be collected in a scalable manner based on distributed feedback/annotation. Further, genes of specific interest such as pharmaceutically relevant genes (GPCR's, NHR's, Kinases, etc.) can be enhanced in an organized way based on their family membership or if the gene shows a low F-measure.

## Acknowledgments

We thank Ian Dix for providing the MeSH to LLID linkage map and Julia Kozlovsky for help in validation of model predictions.

## References

1. Liu H, Johnson SB, Friedman C, Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS, *J Am Med Inform Assoc* **9**:621–636, 2003.
2. MacNeil JS, What big pharma wants, *Genome Tech* **29**:31–38, 2003.

3. Resnik P, Yarowsky D, Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation, *Nat Lang Engi* **5**(3):113–133, 2000.
4. Aronson AR, *Ambiguity in the UMLS Metathesaurus*, National Library of Medicine, 2001.
5. Hatzivassiloglou V, Duboue PA, Rzhetsky A, Disambiguating proteins, genes, and RNA in text: A machine learning approach, *Bioinformatics* **1**:1–10, 2001.
6. Yarowsky D, Word-sense disambiguation using statistical models of Roget's categories trained on large corpora, in *Proceedings of the 14th International Conference on Computational Linguistics*, 454–460, 2000.
7. Gale WA, Church KW, Yarowsky D, A method for disambiguating word senses in a large corpus, *Comp Human* **26**:415–439, 1993.
8. Yarowsky D, Unsupervised word sense disambiguation rivaling supervised methods, in *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA, pp. 189–196, 1995.
9. Aronson AR, Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program, *Proc AMIA Symp*, pp. 17–21, 2001.
10. Rindflesch T, Tanabe L, Weinstein J, Hunter L, EDGAR: extraction of drugs, genes and relations from the biomedical literature, in *Proceedings of the Pacific Symposium on Biocomputing* **5**:514–525, 2000, World Scientific, Singapore.
11. Yu H, Agichtein E, Extracting synonymous gene and protein terms from biological literature, *Bioinformatics* **19**(Suppl. 1):i340–i349, 2003.
12. LocusLink Database, 2003, [ftp://ftp.ncbi.nlm.nih.gov/refseq/LocusLink/LL\\_tmpl.gz](ftp://ftp.ncbi.nlm.nih.gov/refseq/LocusLink/LL_tmpl.gz), accessed December 2004.
13. Pruitt KD, Maglott DR, RefSeq and LocusLink: NCBI gene-centered resources, *Nucleic Acids Res* **29**(1):137–140, 2001.
14. O'Donovan C, Martin MJ, Gattiker A, Gasteiger E, Bairoch A, Apweiler R, High-quality protein knowledge resource: SWISS-PROT and TrEMBL, *Brief Bioinform* **3**:275–284, 2002.
15. Reel Two Classification System, Reel Two Inc. San Francisco, CA, 2001–2004, <http://www.reeltwo.com>.
16. Mitchell T, *Machine Learning*, McGraw-Hill, 1997.
17. Joachims T, Learning to classify text using support vector machines. Dissertation, Kluwer, 2002.
18. Keerthi SS, DeCoste DM, 2004. A modified finite Newton method for fast solution of large scale linear SVMs, Yahoo! Research Labs Tech Report YRL-2004-037.
19. Abramowitz M, Stegun IA (eds.). Psi (digamma) function, in *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Chap. 63, 9th printing, Dover, New York, pp. 258–259, 1972.
20. Weiss SM, Maximizing text-mining performance, *IEEE Int Syst* July/August, pp. 2–8, 1999.
21. Winkler WE, Machine learning, information retrieval, and record linkage, American Statistical Association, in *Proceedings of the Section on Survey Research Methods*, pp. 20–29, 2000.
22. Aas K, Eikvil L, Text categorisation: a survey, Technical report: Norwegian Computing Center, June, 1999.
23. Hearst M *et al.*, Support vector machines, *IEEE Intelligent Systems*, **13**(4), July–August 1998.
24. Medical Subject Headings, National Library of Medicine, 1998, <http://www.nlm.nih.gov/mesh/meshhome.html>.

**Raf M. Podowski** is a Life Sciences Senior Product Manager at Oracle and a Bioinformatics Ph.D. candidate at the Karolinska Institute with a concentration in genome characterization. He has a background in engineering physics and molecular biology and has specialized in development of text mining applications and solution workflows at X-Mine, AstraZeneca and Oracle. His current interests revolve around cognitive text analysis algorithms and knowledge visualization.

**John G. Cleary**, is a professor in the Dept. Computer Science at the University of Waikato, New Zealand as well as Chief Technology Officer of ReelTwo Ltd. His current work is focused on analyzing and extracting content from large document repositories. His research areas include data compression, machine learning, parallel and distributed algorithms and logic programming.

**Nicko T. Goncharoff** is Senior Vice President of Reel Two, Inc. His current work includes product development and project management as well as overseeing research collaborations. He is experienced with text mining and data mining software development, with an emphasis on life sciences applications.

**William S. Hayes**, Ph.D. Molecular Biology and Bachelors in Aerospace Engineering from Georgia Tech., is the head of the Library and Information Services at Biogen Idec. He is focused on extracting the maximum value out of the available literature content through the use of the best techniques in literature analytics and newer operational guidelines. He has extensive experience with text mining, bioinformatics, and grid computing in meeting the challenges of informatics- driven drug discovery.