RULE-BASED HUMAN GENE NORMALIZATION IN BIOMEDICAL TEXT WITH CONFIDENCE ESTIMATION

William W. Lau and Calvin A. Johnson*

Center for Information Technology, National Institutes of Health
Bethesda, MD 20892-5624
**Email: johnson@mail.nih.gov*

Kevin G. Becker

Research Resources Branch, National Institute on Aging
330 Cassell Drive, Baltimore, MD 21224
Email: beckerk@grc.nia.nih.gov

The ability to identify gene mentions in text and normalize them to the proper unique identifiers is crucial for "down-stream" text mining applications in bioinformatics. We have developed a rule-based algorithm that divides the normalization task into two steps. The first step includes pattern matching for gene symbols and an approximate term searching technique for gene names. Next, the algorithm measures several features based on morphological, statistical, and contextual information to estimate the level of confidence that the correct identifier is selected for a potential mention. Uniqueness, inverse distance, and coverage are three novel features we quantified. The algorithm was evaluated against the BioCreAtIvE datasets. The feature weights were tuned by the Nealder-Mead simplex method. An F-score of .7622 and an AUC (area under the recall-precision curve) of .7461 were achieved on the test data using the set of weights optimized to the training data.

1. BACKGROUND

Identification of gene and protein mentions is arguably one of the most difficult *named entity recognition* (NER) tasks in the life sciences domain because of the irregularity and ambiguities in gene nomenclature[1]. A majority of genes can be referred to by more than one name and symbol. Some of these terms are common in the English language and some are even shared by two or more genes, of the same and/or different species[2]. An evaluation that was conducted for the BioThesaurus found that a gene/protein has an average of 3.53 synonyms, and that the same term is associated with 2.31 different concepts on average[3]. In search of records for a particular gene, most search engines, including PubMed, performs only basic keyword matching. This leads to substantial number of false positives and false negatives, making it difficult for users to locate the information that are truly useful to them.

A number of systems have been developed in the past few years to address the problem of gene recognition. The techniques fall into two broad categories, machine-learning and rule-based methods, which vary in their degree of reliance on dictionaries, statistics, linguistics, and heuristics[1]. Machine-learning approaches, including hidden Markov Models[4] and support vector machines[5], are very scalable. However, these techniques are very sensitive to the selection of features[6] and the results are difficult to interpret. In rule-based approaches, hand-crafted rules for specific datasets are derived by experts with domain knowledge. These rules are often implemented as regular expression statements. Although this approach can be quite labor-intensive, rule-based systems is often superior in handling genes that do not appear in training data. Traditionally, the more human interventions there are in a system, the better the system performs[7]. As annotated data sets become more readily available and the learning techniques become more sophisticated, this trend may change in the near future.

Many popular tools, such as ABNER[8] and GAPSCORE[9], address the problem of NER without uniquely identifying the entities being mentioned. However, the ability to accurately associate these text mentions with specific entries in biomedical databases is of great value to "downstream" text mining applications, e.g. document classification and knowledge discovery. The next step beyond gene mention tagging is *gene normalization*. It is a procedure in which each gene occurrence in the text is mapped to a unique gene identifier[10]. In case of mentions associated with multiple identifiers, additional steps have to be taken to select the correct identifier among all the candidates. To study associations between genes using information in the literature, Jenssen *et al.*[11] used simple string matching for gene recognition. Up to 40% of associations were incorrect, due to these problems in normalization: symbols shared by several genes, syntactical variations of the terms, and insufficient synonym lists. Thus, a more sophisticated gene normalization technique is required.

Various competitions on text mining have been held in the past to create a platform where different text mining approaches can be compared objectively using common standards and evaluation criteria. BioCreAtIvE is one of the several competitions specifically tailored to the biological domain. The first evaluation was held in 2003, and attracted 27 participants from around the world. We entered Task 2 of the second BioCreAtIvE challenge[12]. The objective of this task was to return the EntrezGene identifiers corresponding to the human genes and direct gene products appearing in a set of MEDLINE abstracts annotated by researchers at the European Bioinformatics Institute.

Our gene normalization algorithm is a prototype component of the PubMatrix system [13], a text mining tool for genetic association studies. The advent of high-throughput microarray analysis has made it possible to measure the expression of thousands of genes and proteins simultaneously. However, the large volumes of data that are being generated create a huge challenge for scientists to effectively interpret and evaluate their results. PubMatrix, among others, can be used to systematically identify associations between sets of genes and diseases using information available in the MEDLINE literature. The assumption is that if the co-occurrence frequency between a gene and a disease is of statistical significance, they probably have an underlying biological relationship. The PubMatrix system thereby helps researchers to validate their experimental results and to select a manageable set of promising genes for further analysis. Since simple string matching of the genes has yielded poor performance in other studies, we developed the gene normalization algorithm to help improve the accuracy of the PubMatrix results.

Our system is essentially a rule-based system utilizing information from knowledge bases, statistical analysis, and empirical evidence. Section 2 is an extension to our paper submitted to the BioCreAtIvE Workshop[14]. This section describes our gene normalization algorithm, in particular the metric we use to estimate the confidence level of a match. In Section 3, our experimental results on the BioCreAtIvE data will be presented. We conclude with Section 4 by discussing performance issues and the significance of each component of the confidence measure.

Table 1. Regular expression rules applied to gene symbol pattern matching to account for several syntactic variations commonly encountered in the literature.

Rules	Examples
Interchange of Roman and Arabic numerals	GAL4 → GAL IV
Interchange of dashes and spaces	NKG2-E → NKG2 E
Allow a dash or space in front of a numeral	NAT2 → NAT-2
Allow an optional 's' at the end of a symbol	EST → ESTs
Allow an optional 'h' at the beginning of a symbol	B1F → hB1F
Allow for case difference if symbol has more than two characters	RAC1 → Rac1

2. IMPLEMENTATION

2.1. Identification of Gene Mentions

The algorithm detects the occurrence of gene mentions by matching input text against the EntrezGene dictionary from the National Library of Medicine. The procedure effectively combines the tasks of gene detection and gene identifier lookup. Different approaches are used in the detection for gene symbols (including "Other Aliases" in the EntrezGene database)

and gene names (including "Other Designations"). Gene-symbol tagging is based on pattern matching. For each symbol in the knowledge base, a set of regular expressions rules, as shown in Table 1, are applied to evaluate every string separated by space and punctuation symbols. For the official symbols, we also generate new symbols by expanding the associated Greek letters into their full names, e.g. "CHKB" to "CHK beta" and "beta CHK".

For gene names, an approximate term matching technique has been employed. After breaking a gene name into individual words or tokens, each token is searched against the text using rules similar to gene symbol matching. Subsequently, the phrase containing the most tokens is identified. This phrase is conditionally accepted if the ratio, r_m, between the number of tokens in the mention candidate and the total number of tokens to be matched is higher than a threshold (0.7 in our submissions). However, the candidate has to include specific tokens as measured by the number of citations containing those tokens (if a token's frequency of occurrence is low, it is too important to be ignored). The system also maintains a list of allowed and prohibited missing words. If a word in the prohibited list, e.g. "receptor", is missing from the phrase, the candidate is rejected. On the other hand, if a word in the allowed list, such as "type" and "subunit", is missing in the candidate, the algorithm calculates r_m as if the word were not in the gene name.

As an illustration, consider the gene "angiotensin II receptor, type 1," which consists of five tokens. The term "angiotensin II type 1" has an r_m of 0.8, but is rejected because "receptor" is missing. On the other hand, the term "angiotensin II receptor alpha" has an r_m of 1.0. In addition, another rule is that candidates are allowed to contain at most two extra words between any two tokens as long as the words are frequently found in the biomedical literature. Besides the names that are already in the knowledge base, additional synonyms are generated by replacing common chemical names with their abbreviations. For example, "acetyl-CoA carboxylase beta" is created from "acyl-Coenzyme A carboxylase beta". This approximate matching technique, which is similar to that proposed by Hanisch *et* al [15], can accommodate typical variations of gene name mentions, such as word ordering, found in the literature.

2.2. Confidence Measure of Gene Mention Candidates

After a gene mention is detected, the algorithm calculates a confidence score using several statistical and heuristic measures. The three most novel features used in our submissions were *coverage, inverse distance,* and *uniqueness*.

2.2.1. *Coverage*

The calculation of the coverage score is quite different between gene names and gene symbols. The score for symbols, ψ_S, is calculated as follows:

$$\psi_S = \left(\frac{\tan^{-1}(2L-3)}{\pi} + 0.5 \right) \times s \qquad (1)$$

where L is the character length of the term extracted from the text and, s is a scaling factor defined as:

$$s = \begin{cases} \left(e^{\frac{r_m - 1}{L}} \right)^2 & \text{if the candidate is enclosed} \\ k_1 + (1-k_1) e^{r_m - 1} & \text{otherwise} \end{cases}$$

where $0 \le k_1 \le 1$ is a parameter (set to 0.8). The intuition is that the more characters the symbol has, the less likely it is that the term is used other than to represent the gene. If the term is enclosed by brackets, i.e. ([{}]), the gene name is probably mentioned in the text as well and score should be scaled accordingly.

For gene names, the coverage score is a weighted average of two ratios, r_L and r_m. r_L is the ratio of the character length of the candidate string to the corresponding name in the knowledge base. Thus,

$$\psi_N = k_2 r_m^{\left(\frac{3 f_{\min}}{f_m} + 1\right)} + (1-k_2) r_L \qquad (2)$$

where $0 \le k_2 \le 1$ is a parameter (set to 0.5), $f_{\min}$ is the minimum occurrence frequency threshold for any missing words not in the allowed list (set to 20,000), and f_m is the occurrence frequency of the least common missing word. In addition to character length, the coverage metric for gene names also takes into account how many words are matched as well as the specificity of the words missing from the mention.

2.2.2. *Inverse Distance*

For gene symbols, inverse distance is based on the edit distance, d_L, of the candidate term to the formal reference in the database. The score, δ_S, is defined as follows:

$$\delta_S = \left(1 - \frac{d_L}{L}\right)^{k_5} \tag{3}$$

where k_5 equals to (1-s)/L. It takes into consideration the variations in capitalization, ordering, and any omissions/additions of punctuations and spaces. The closer the mention matches the actual symbol, the higher the score. For gene names, since syntactic variations are common, Eq. (3) is modified by factoring into the token ratio r_m:

$$\delta_N = \left(2 \cdot r_m \cdot (1 - \frac{d_L}{L})\right) / \left(r_m + (1 - \frac{d_L}{L})\right) \tag{4}$$

2.2.3. *Uniqueness*

Uniqueness is an estimate of the probability that the candidate is referring to something other than the gene in question. If the mention has a very high frequency of occurrence in the literature, the score is reduced accordingly, because frequently occurring terms may have multiple meanings other than just being referred as genes. For gene names, the uniqueness score, μ_N, has two components, one being influenced by the size of the population, T, and the other by a user defined frequency threshold, $f_{\max}$, which limits the maximum number of documents the term can appear in (set to 40,000 in our experiments). Thus, μ_N is given as:

$$\mu_N = \left[1 - \frac{k_3 f}{f_{\max}}\right] \times \left[1 - \left(\frac{f}{T}\right)^{k_4}\right] \tag{5}$$

where k_3 and k_4 are parameters (set to 0.5 and 10, respectively), and f is the number of documents containing the term. The population we use in our system is the entire collection of MEDLINE citations. Formulation of the uniqueness score for gene symbols is the same as Eq. (5), except that the score is further multiplied by the scaling factor s.

2.2.4. *Discrete Features*

We have identified three additional features that could assist the algorithm to select the correct identifier in case of ambiguity. First, if more than one unique mention of a gene is extracted from the text (e.g. both name and symbol), our confidence that the correct identifier is selected increases. This feature is referred to as *number of mentions*. In addition, many genes in the EntrezGene knowledge base have not been approved by the HUGO Gene Nomenclature Committee. We believe that the references for these genes are unstable and few articles on these genes have been written. Therefore, in the *official status* feature preference is given to genes that have been approved. A related feature is *mention type*. A recent study[16] suggests that scientists do not usually follow standard nomenclatures. Suspecting that there exists some degree of correlation, we take into consideration whether the mention is an officially approved term.

We also incorporate a boosting factor into the confidence measure to reward or punish a candidate when there is contextual clue in the citation suggesting whether the mention actually refers to a gene. For example, if the text contains the chromosome location or accession numbers of the gene, its score will be boosted. If the mention is preceded or followed by supporting modifiers, such as "gene" and "encode", we have a much higher level of confidence that this mention is a true positive. On the contrary, if counter-indicators, such as "test" and "cell line", appear adjacent to the candidate, the mention should be penalized by inverting the boosting factor. Therefore in addition to the allowed and prohibited missing word lists, we also maintain a list of indicator terms and a list of counter-indicator terms. Whereas all the other factors are combined linearly to compute the final score, the boosting factor is added last as an exponent to the score. The final confidence score for a mention is simply calculated as:

$$S_c(\vec{w}) = \left(\sum_{i=1}^{6} w_i c_i\right)^{b(w_7)} \tag{6}$$

where b is the boosting factor, n is the number of features not considered in the boosting, and w_i and c_i are the weight and sub-score for feature i, respectively. Consequently, a list of gene mentions with their associated identifiers and confidence scores is created

for each citation. An acceptance threshold can be applied to improve precision. If a gene has more than one unique mention in the text, the maximum score is used.

2.3. Overlapping of Gene Mention Boundaries

When a string is associated with more than one gene identifier, the algorithm needs to determine which gene the authors actually intended. The disambiguation procedure is as follows. First, if a mention appears entirely within another longer mention (Fig. 1a), the algorithm removes the shorter mention if it does not appear anywhere else by itself in the text. If some words of a mention overlap with another mention (Fig. 1b) or if two mentions share the exact same term (Fig. 1c), the one with a lower score is removed. If the scores of two conflicting candidates are equal, their uniqueness scores are both reduced by half. The effect of this operation is that if the mentions are weak at the first place, they can both be eliminated with a smaller threshold. If the candidate had more than one form of occurrence, e.g. both the symbol and the name were detected, the highest score was considered. Moreover, if two genes are adjacent to each other without being separated by any punctuation (Fig. 1d), we remove either the first mention or the mention with a lower score.

Fig. 1. Four cases of boundary conflicts are illustrated. When a mention is completed covered by another mention (a), the shorter mention is taken out from the gene list. The confidence score is used to determine which mention is more probable in cases (b), (c), and (d). For (d), if the score is the same for both mentions, the first mention is removed.

3. EVALUATION

We evaluated our gene-normalization system by finding a (locally) optimized set of weights w^{train} on a training-data set, testing the performance of the system using w^{train} on a testing-data set, and then cross-validating the performance by training on the testing-data set to generate a set of weights w^{test} which were evaluated on the training data set. The training and test data sets were those provided by the by the BioCreAtIvE II gene normalization task. These data sets comprise and 286 and 262 documents, respectively. The results of the optimization process are summarized in Tables 2 and 3. Table 2 provides the values of the original weights w^0 we used in the competition as well as the tuned weights w^{test} and w^{train}. Table 3 gives the results obtained from running the optimized weights through the data set on which they were trained as well as on the other data set. The maximum F-score and area under the recall-precision curve (AUC), which were obtained by testing w^{train} on the testing data set, were found to be 0.7622 and 0.7554 respectively. With the original weights, prior to optimization, these values were 0.7523 and 0.7423 respectively.

To generate w^{train} and w^{test}, the set of starting weights w^0 was first obtained through empirical evidence and knowledge gained through the experience of developing the system. A good starting point for the optimizer was then found by manually exploring the energy landscape of the maximum F-score and AUC. A set of weights was then selected from the trial set which we felt could be considered "close to the maximum." These weights were entered as a starting point to the Nelder-Mead simplex method[17], an unconstrained derivative-free method which can find a local maximum via a geometric process involving reflection, contraction, and expansion. Although it has poor theoretical properties, the Nelder-Mead method is surprisingly robust for objective functions that are not analytical. Although we are using an unconstrained optimizer, our problem is actually constrained, namely

$$\begin{aligned} \max \quad & \tfrac{1}{2}\left(F_{\max}(w) + AUC(w)\right) \\ s.t. \quad & w_j \geq 0, \quad j = 1,\ldots,6 \\ & w_7 \geq 1 \end{aligned} \tag{7}$$

where $F_{\max}(w)$ is the maximum F-score obtained over a set of thresholds and $AUC(w)$ is the AUC from those same set of thresholds. In the results that we report, we used a threshold interval of 0.01, or 100 estimates when the maximum threshold is 1. As is clear from Table 3, we obtained an optimal solution well within the bound constraints.

We also pondered imposing an equality constraint $w_1 + w_2 + \ldots + w_6 = 1$ to enforce the idea that the maximum threshold must be 1 and that the result is properly "scaled." Doing so would have necessitated a genuinely constrained optimizer. Rather than facing these complications, we adjusted the method to allow

for arbitrary thresholds. As a precaution, we used a starting point that was normalized according to the equality constraint.

Table 2. Weights obtained through an optimization process (w^{train} and w^{test}) as well as starting weights w^0. The actual value of the weights w_1 through w_6 are the product of the values shown and the denormalization factor.

		Original weights w^0	Train on training set w^{train}	Train on testing set w^{test}
Mention type	w_1	0.1800	0.1224	0.1389
Coverage	w_2	0.2333	0.2082	0.1743
Inverse distance	w_3	0.2333	0.2744	0.2414
Uniqueness	w_4	0.2333	0.1961	0.2357
Number of mentions	w_5	0.1000	0.0580	0.0629
Official status	w_6	0.0200	0.1407	0.1467
Boosting factor	w_7	1.2500	1.4514	1.4402
Denormalization factor		1.0000	1.0207	0.9699

Table 3. Result of running the normalization system on the training and testing data provided for the BioCreAtIvE II gene normalization task. The combination measure is equal to half the F-score plus half the area under the recall-precision curve (AUC).

Test on:	Measure	w^0	w^{train}	w^{test}
Training set	Max. F-score	0.7703	0.7757	0.7733
	AUC	0.7516	0.7586	0.7593
	Combination	0.7609	0.7671	0.7663
Testing set	F-score	0.7523	0.7622	0.7677
	AUC	0.7423	0.7485	0.7546
	Combination	0.7473	0.7554	0.7611

Figs. 2 and 3 plot the F_{max} and AUC, respectively, versus feature weight values for the first six features, i.e. w_1 through w_6. These figures each contain six plots corresponding to the six features. In each plot, only the corresponding weight is allowed to change through the range of 0 to 1 while the other weights are held to their w^{train} values. Since the results were obtained by testing the training-data-optimized weights against the test data, not surprisingly there exist solutions on the test data with greater maxima than our solution. Despite this, we feel that our solution fared well on a foreign data set. Our explorations did reveal a somewhat difficult energy landscape with multiple maxima. Not surprisingly, the AUC curve is smoother than the maximum F-score. In Fig. 4, the effect of the boosting factor w_7 is demonstrated by plotting the maximum F-score and AUC versus w_7 while the other weights are held to their w_{train} value. In Fig. 5, the recall-precision curve is plotted for weights set to w^0 as well as w^{train}.

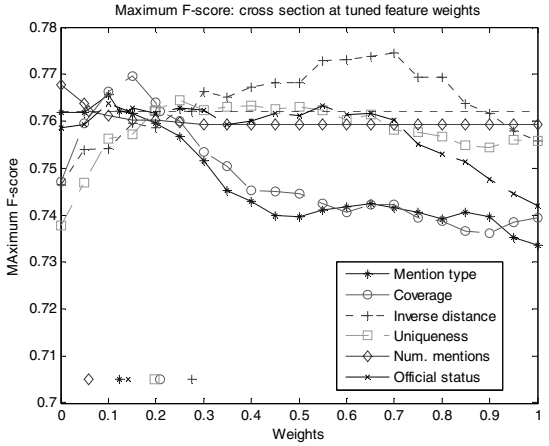

Fig. 2. Maximum F-score for the first six features versus variations in the weights of the corresponding feature while the other weights were set to the w^{train} values. The markers on the lower right indicate the w^{train} values. The horizontal lines are the F-score at w^{train}. Results obtained in tests against the testing set.

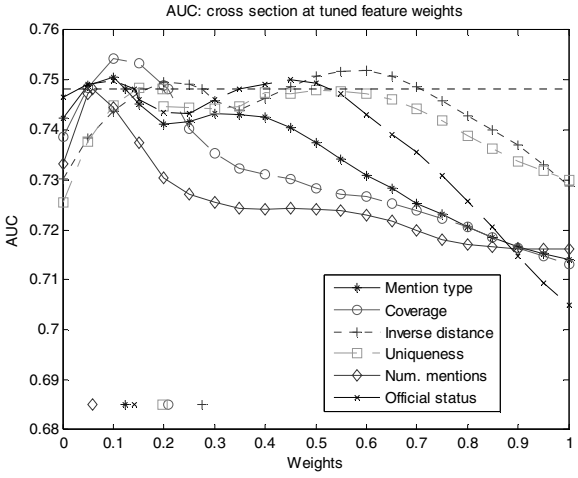

Fig. 3. AUC for the first six features versus variations in the weights of the corresponding feature while the other weights were set to the w^{train} values. The markers on the lower right indicate the w^{train} values.

4. DISCUSSION

We have developed a gene normalization algorithm that separates the task into two processes. First, the

algorithm searches for possible gene mentions with the goal of high recall. Different techniques are applied to the search for gene symbols and gene names, although both rely on the use of dictionaries and rules. The rules are important as they consider many syntactical variations that are commonly encountered in gene nomenclature. Since most gene names are phrases rather than single words, an approximate term matching technique is employed to also account for differences in word ordering and word choices. The second process of the algorithm attempts to improve the precision by measuring the level of confidence of each match and filtering out those mentions that have low confidence score. The confidence score is derived from a set of quantitative measures leveraging statistical, morphological, and contextual information available to the system. In addition to indicating whether a term actually refers to a gene or not, these measures provide a means for the system to disambiguate mentions to which multiple genes are mapped.

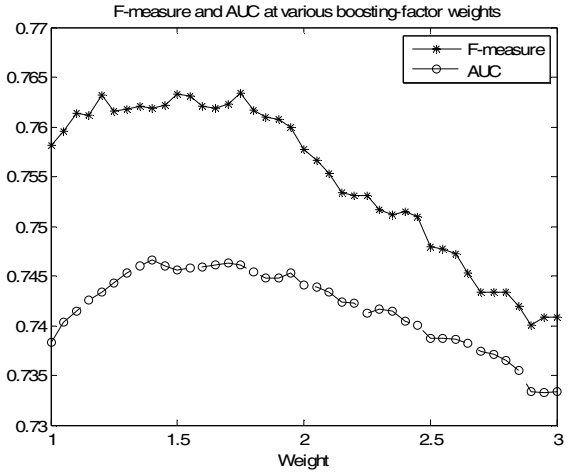

Fig 4. Effect of the boosting factor. F-measure and AUC versus the boosting factor w_7 while the other weights were set to w^{train}. Results obtained against the testing set**.**

Using the BioCreAtIvE datasets for evaluation of the algorithm, the best *F*-score we achieved on the test data was 0.7622 when the feature weights were optimized with the training data. Without the thresholding process, the gene tagging component alone could attain an *F*-score of 0.647 with a recall of 0.869. Recall at this step essentially limits the recall obtainable in the thresholding process. A majority of the undetected mentions have complex syntax not being handled by the rules we defined. Table 4 provides some examples of challenging cases that contributed to the false negative counts in the tagging process. Nevertheless, many genes are referred to in the text both by their name and symbol. The undetected mentions thus result in a smaller impact on the recall performance.

Figs. 2 and 3 show the individual contribution of each internal feature we measure in the confidence score. We call these internal features because the scores are computed out-of-context, based solely on the evidence presented by the mentions themselves. The only exception is the scaling factor *s* on gene symbols, which is influenced by whether the symbol is extracted from text enclosed by a set of brackets. We can observe from the figures that all six features are useful for the gene normalization task because their optimal weights are all greater than zero. As the weight of a feature increases, the feature becomes more dominant in determining the final confidence score. Inverse distance and uniqueness are the only features that produced better results (on AUC) or only slightly degraded (on *F*-score) results from zero weight to a weight of 1. All the other features posted worse performance when they became dominant. Although the best performance is achieved using a combination of these features, our observation suggests that inverse distance and uniqueness have good enough discriminatory power to estimate the level of confidence by themselves when other information is not available. In addition to the internal features, several contextual factors are used to determine whether the confidence score is boosted or not. Since the boosting factor is added as an exponent, the effect is non-linear. Boosting exerts most of its influence on mentions for which the internal features may be ineffective. When a gene is mentioned for the first time in the text, the authors often specify that the entity of interest is a gene, especially when the gene is ambiguous or not very well known. Boosting is useful as illustrated in Fig. 4. However, sometimes a wrong mention can be boosted. Moreover, when counter-indicators are detected, the boosting factor is inverted and the score is thus reduced. It can be argued that the punishing factor should be made more severe in order to successfully remove those mentions that have high scores but actually refer to something else.

Features for confidence measure. In contrast with the other features, the effect of coverage, inverse

distance, and uniqueness are clearly pivotal as there is significant performance improvement from zero weight to their optimal settings. It can be argued that uniqueness is the most important feature in our evaluation. Lack of this feature would result in severe degradation of performance, most noticeable in the AUC. Uniqueness is a statistical measure with the assumption that gene mentions should have a low frequency of occurrence. This is a good assumption in most cases. However, it is not good with legitimate genes that actually appear frequently in the literature (e.g. Interleukin 1) and relatively rare terms with multiple meanings, one of them being a gene reference. For example, "ADA" can stand for the American Diabetes Association or the gene *adenosine deaminase*. Our solution to the second issue is to look at whether a symbol is mentioned within a set of brackets. If it is the case, presence of the gene name becomes a determining factor. We found this contextual feature to be very helpful for improving precision. Another important feature is the inverse distance, which is a dictionary-based measure that calculates the similarity between the candidate mention and the corresponding gene term in the database. Currently, character is the basic unit in the calculation of edit distance. For names, the effect of changing the word order is subject to the length of the words. It may be more appropriate to use word as the unit of measurement. Coverage is mostly a heuristic measure in which we assume longer mentions are more likely to be true. Albeit that it is a very good measure, the performance degraded when it become a dominant factor, suggesting that length alone is not reliable.

Comparison to other gene normalization tools. A number of gene tagging tools are freely available to the community, but to our knowledge, no standalone gene normalization systems have been made publicly accessible. No comparison is made between our tool and ABNER or GAPSCORE because the task of these tools (i.e. NER) is different from ours (i.e. normalization) and such comparison would not be particularly meaningful. In the second BioCreAtIvE challenge, 20 teams entered the gene normalization task[12]. Many teams followed the same general approaches we employed. Several participants built upon "off-the-selves" gene tagging tools. The best F-score from each team ranges from 0.810 to 0.394, with a median of 0.731. The highest recall and precision achieved are 0.833 and 0.841, respectively. The difference in performance is primarily due to the way filtering of candidates, including disambiguation, was performed. Some relied on pruning of the lexicon and some implemented rules of various degrees of sophistication to reduce false positives. Nevertheless, the results of the top scoring teams, including ours, are comparable. It is important to note that the recall of 0.869 at a precision of 0.515 which we achieved after the first step of the process is advantagous when high recall is required. Another benefit that our system provides is that each mention is associated with a confidence score. This feature affords users the ability to choose a suitable balance between recall and precision.

Table 4. Examples of false negative cases in which the algorithm was not able to detect them at all.

Description	Examples
Range	ORP-1 to ORP-6
Ambiguity	p32
Choice of words	IFN-induced protein of 10 kDa
Boundary	Protein kinase C isoforms alpha, epsilon, and zeta

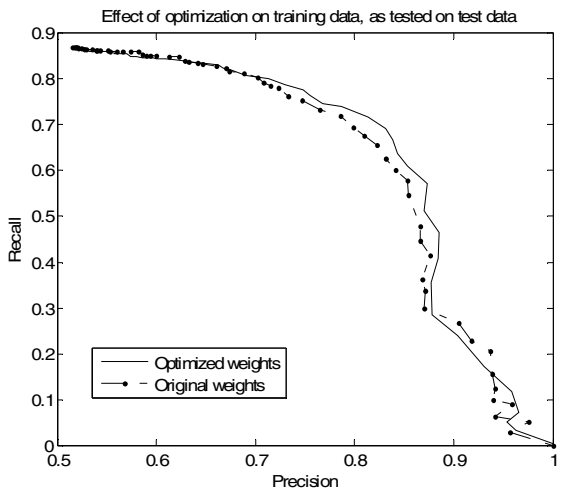

Fig. 5. Recall versus precision as tested on the test data with the original weights (w^0) and the optimized weights w^{train}.

5. CONCLUSION

We have developed a gene normalization algorithm that relies heavily on rules that combine statistics and heuristics. The confidence measure provides a means to quantify the degree of conformance to these rules and allow users to choose the proper compromise between recall and precision based on the situation. In our evaluation, only basic knowledge about the genes was used to disambiguate mentions with multiple mappings. A majority of candidates that mapped to more than one gene identifier actually referred to gene families. For future work, information about gene families and association of various terms can be applied for more sophisticated filtering. Part-of-speech tagging may also help to discern mention boundaries and improve system efficiency by only considering noun phrases.

Acknowledgments

This research was supported by the Intramural Research Program of the National Institutes of Health, Center for Information Technology. We appreciate the contributions of Alex Wang and Jigar Shah.

References

1. Tanabe L, Wilbur WJ. Tagging gene and protein names in biomedical text. *Bioinformatics*. 2002; **18**: 1124-1132.
2. Jensen LJ, Saric J, Bork P. Literature mining for the biologist: From information retrieval to biological discovery. *Nat Rev Genet*. 2006; **7**: 119-129.
3. Liu H, Hu ZZ, Torii M, Wu C, Friedman C. Quantitative assessment of dictionary-based protein named entity tagging. *J Am Med Inform Assoc*. 2006; **13**: 497-507.
4. Zhou G, Zhang J, Su J, Shen D, Tan C. Recognizing names in biomedical texts: A machine learning approach. *Bioinformatics*. 2004; **20**: 1178-1190.
5. Hakenberg J, Bickel S, Plake C, et al. Systematic feature evaluation for gene name recognition. *BMC Bioinformatics*. 2005; **6 Suppl 1**: S9.
6. Leser U, Hakenberg J. What makes a gene name? named entity recognition in the biomedical literature. *Brief Bioinform*. 2005; **6**: 357-369.
7. Dickman S. Tough mining: The challenges of searching the scientific literature. *PLoS Biol*. 2003; **1**: E48.
8. Settles B. ABNER: An open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*. 2005; **21**: 3191-3192.
9. Chang JT, Schutze H, Altman RB. GAPSCORE: Finding gene and protein names one word at a time. *Bioinformatics*. 2004; **20**: 216-225.
10. Hirschman L, Colosimo M, Morgan A, Yeh A. Overview of BioCreAtIvE task 1B: Normalized gene lists. *BMC Bioinformatics*. 2005; **6 Suppl 1**: S11.
11. Jenssen TK, Laegreid A, Komorowski J, Hovig E. A literature network of human genes for high-throughput analysis of gene expression. *Nat Genet*. 2001; **28**: 21-28.
12. Morgan A, Hirschman, L. Overview of BioCreative II Gene Normalization. *Proc of the Second BioCreative Challenge Evaluation Workshop* 2007.
13. Becker KG, Hosack DA, Dennis G,Jr, et al. PubMatrix: A tool for multiplex literature mining. *BMC Bioinformatics*. 2003; **4**: 61.
14. Lau W, Johnson C. Rule-based gene normalization with a statistical and heuristic confidence measure. *Proc of the Second BioCreative Challenge Evaluation Workshop* 2007.
15. Hanisch D, Fundel K, Mevissen HT, Zimmer R, Fluck J. ProMiner: Rule-based protein and gene entity recognition. *BMC Bioinformatics*. 2005; **6 Suppl 1**: S14.
16. Tamames J, Valencia A. The success (or not) of HUGO nomenclature. *Genome Biol*. 2006; 7: 402.
17. Nelder JA, Mead R. A simplex method for function minimization. *Comput J*. 1965; **7**: 308-313.