

A Probabilistic Model for Identifying Protein Names and their Name Boundaries

Kazuhiro Seki and Javed Mostafa

Laboratory of Applied Informatics Research, Indiana University
1320 East Tenth Street, LI 011, Bloomington, Indiana 47405-3907

{kseki, jm}@indiana.edu

Abstract

This paper proposes a method for identifying protein names in biomedical texts with an emphasis on detecting protein name boundaries. We use a probabilistic model which exploits several surface clues characterizing protein names and incorporates word classes for generalization. In contrast to previously proposed methods, our approach does not rely on natural language processing tools such as part-of-speech taggers and syntactic parsers, so as to reduce processing overhead and probabilistic parameters to be estimated. A notion of certainty is also proposed to improve precision for identification. We implemented a protein name identification system based on our proposed method, and evaluated the system on real-world biomedical texts in conjunction with the previous work. The results showed that overall our system performs comparably to the state-of-the-art protein name identification system and that higher performance is achieved for compound names. In addition, it is shown that our system can further improve precision by restricting the system output to those with high certainties.

Keywords: named entity extraction, protein name identification, probabilistic models, protein name boundaries

1 Introduction

Ever-growing digitized texts have resulted in a demand for automated techniques to extract novel information from texts. Message Understanding Conferences (MUCs) [8] represent one of the major attempts to develop information extraction (IE) techniques targeting general texts (newswire articles) in which the participants independently implement IE systems and compare their system performance on a common test set.

IE is crucial and urgent also in the field of molecular biology because of a strong demand for automatically discovering molecular pathways and interactions in the literature, which is, even for human experts, labor-intensive and time-consuming. Therefore, much research has been done to explore IE techniques on biomedical texts [1, 6, 9, 13, 16, 17, 18, 20].

Our ultimate goal is to realize an automated system to discover novel information in the biomedical literature, specifically, relations and interactions between specific proteins and cancer, which is expected to be beneficial for developing new medicine and treatments peculiar to cancer. To accomplish our goal, we start with identifying protein names appearing in biomedical texts. However, automatic protein name identification is not a trivial task. This is partially because there are no common standards or fixed nomenclatures for protein names which are followed in practice. As new proteins continue to be discovered and named, predefined protein name dictionaries are not necessarily helpful in identifying new protein names. Additionally, protein names frequently appear in shortened, abbreviated, or slightly altered forms (e.g., capital and small letters and hyphens). Therefore, even the protein names that are already known and are supposed to be contained in a dictionary might be overlooked due to way they are actually written. Another challenging issue for identifying protein names is to find their name boundaries. According to our preliminary research on 99 MEDLINE abstracts, 42% of protein names are composed of multiple words (tokens), and these tokens include common nouns, adjectives, adverbs, and even conjunctions, which makes it difficult to distinguish protein names from the surrounding texts [19].

We propose a statistical approach to identifying protein names in biomedical texts. Our approach employs probabilistic models for finding protein name boundaries and for restricting the final output to those with high certainty so as to improve the accuracy of identification. The probabilistic models exploit surface clues which reflect the characteristics of protein names. To evaluate our method, a series of experiments is conducted in comparison with results produced in previous work by other researchers.

Section 2 briefly summarizes previous work related to protein name identification, and Section 3 details our proposed method. In Section 4, the methodology of evaluation is described and the result is presented and discussed. In Section 5, we conclude this paper with our findings and future work.

2 Related Work

There have been number of attempts to develop techniques to extract protein names in the biomedical literature. They roughly fall into three approaches, that is, dictionary-based, heuristic rule-based, and statistical.

A technique based exclusively on a dictionary is not necessarily helpful for identifying protein names because new protein names continue to be created and there are often many variations in the way identical proteins are referred to. To tackle this problem, Krauthammer et al. [11] proposed an approach to protein and gene name extraction, using BLAST [2], a DNA and protein sequence comparison tool. Their basic idea is performing approximate string matching after converting both a dictionary and input texts into nucleotide sequence-like strings, which are then compared by BLAST. The results they reported, however, cannot be directly compared with our case, because they targeted both protein and gene names and the results were not separately reported.

Fukuda et al. [7], Narayanaswamy [12], and Olsson et al. [15] proposed rule-based approaches. They exploited surface clues for detecting protein name fragments (i.e., parts of protein names) and used a part of speech tagger and/or a syntactic parser for finding protein name boundaries. Typically, the surface clues include the following features, where bold characters indicate the corresponding examples.

- Capital letters (e.g., **ADA**, **CMS**)
- Arabic numerals (e.g., **ATF-2**, **CIN85**)

- Roman alphabets (e.g., Fc **alpha** receptor, 17**beta**-estradiol dehydrogenase)
- Roman numerals (e.g., dipeptidylpeptidase **IV**, factor **XIII**)
- Words appearing frequently in protein names (e.g., myelin basic **protein**, PI 3-**kinase**, nerve growth **factor**)

Olsson et al. [15] conducted experiments that compared their system (Yapex) with Fukuda’s system (Kex) on 101 MEDLINE abstracts. Yapex achieved a recall of 65.3% and a precision of 68.8% as compared to a recall of 37.5% and a precision of 34.3% on Kex in terms of exact match.

Statistical approach has made a considerable impact on natural language processing (NLP) research and related areas, such as part-of-speech (POS) tagging, parsing, and speech recognition. In the biomedical domain, Collier et al. [3], Nobata et al. [14], and Kazama et al. [10] employed statistical approaches (e.g., hidden Markov models, decision trees, probabilistic models, and support vector machines) for detecting and classifying gene and gene product names including proteins. The features used in their methods are mostly the same as those used in rule-based approach, that is, surface clues and parts of speech.

Rule-based approach has an advantage in that rules can be flexibly defined and extended as needed, but manually analyzing targeted domain texts and creating rules are often time-consuming. Statistical approach is relatively easy to be applied if appropriate training data are provided. However, statistical approach in general cannot reasonably deal with the cases that do not appear in the training data (i.e., the data sparseness problem). In general, to achieve higher performance, a more complex model is needed, which, however, requires more training data to estimate the increasing number of parameters.

We mainly employ a statistical approach using a probabilistic model for identifying protein names with an emphasis on finding name boundaries. Our method solely exploits surface clues, unlike previous work, avoiding the use of part-of-speech taggers and syntactic parsers. According to our preliminary investigation on the corpus annotated with 1,745 protein names made by Franzén et al. [5], protein name fragments can be not only nouns but also adjectives, adverbs, verbs, and conjunctions, and thus POS tags are unlikely to be helpful to detect protein names and their boundaries. Avoiding the use of NLP tools reduces both processing overhead and the number of parameters to be estimated. Moreover, we generalize words that compose protein names to word classes and also apply a smoothing method in order to compensate for the limited amount of training data.

3 Our Method

3.1 Overview

Figure 1 depicts an overview of our protein name identification system based on the method described in this section. In the preprocessing module, an input text is partitioned into sentences and tokenized, where tokens are defined as words and symbols. For instance, PI 3-kinase will be separated into four tokens, i.e., PI, 3, -, and kinase.

Then, we identify protein names through three steps. First, protein name fragments are detected by heuristic rules relying on surface clues which are commonly used for protein name identification. Second, for each of the detected protein name fragments, the name boundary is expanded based on a

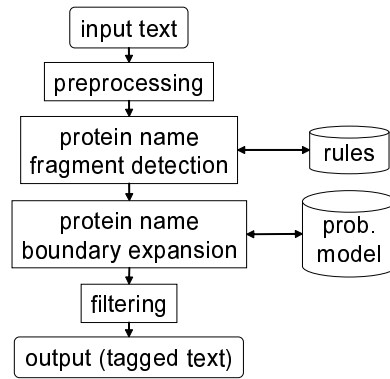


Figure 1. An overview of our protein name extraction system.

probabilistic model to locate complete protein name candidates. Lastly, a filter is applied to the candidates so as to exclude erroneous detections, and only those with high certainties are output. Each step is further explained in Section 3.2–Section 3.4.

3.2 Protein name fragment detection

We use several heuristic rules to detect protein name fragments which have been commonly used in previous studies [5, 7, 12]. Words which satisfy any of the following conditions are detected as potential protein name fragments.

- words which include capital letters (i.e., A, B, C, \dots , Y, Z)
- words which include combinations of Arabic numerals (i.e., 0, 1, 2, 3, \dots , 8, 9) and lower case letters (i.e., a, b, c, \dots , y, z)
- words which have suffixes that often appear in protein name fragments (i.e., $-nogen$, $-ase$, $-in$)
- words which often appear as protein name fragments (i.e., $factor(s)$, $receptor(s)$)
- Roman alphabets that often appear as protein name fragments (i.e., α , β , γ , δ , ϵ , κ)

These conditions unfortunately also detect words that are not protein name fragments. For example, if we extract all words containing capital letters, words located in the beginning of sentences are inevitably extracted as protein name fragments. To decrease these errors, we exclude the following tokens:

- words which have a capital letter in the beginning followed by more than three lower case letters (e.g., $According$, $Basically$)
- words which are composed of only capital letters longer than 6 characters. (e.g., $KTPGKKKKGK$)
- only one character (i.e., A, B, \dots , Y, Z)
- measuring units (e.g., nM , MM , mM , pH , MHz)

- chemical formulas (e.g., CaCl₂, NH₂, Ca₂, HCl, Mg₂)
- words included in a stopwords list. Here, we used the Pubmed Stopword List, which contains 133 function words in a medical domain¹.

3.3 Protein name boundary expansion

We employ a probabilistic model for expanding/finding a protein name boundary leftward and rightward for each of the detected protein name fragments. In the following, we will explain the details of our model, focusing on expanding name boundaries rightward for example.

Let w_i denote one of the protein name fragments detected in the previous step. Given a fragment w_i , the probability that a token w_{i+1} following to w_i is also a protein name fragment can be expressed as a conditional probability $P_p(w_{i+1}|w_i)$, assuming a first order Markov process. Likewise, the probability that w_{i+1} is *not* a protein name fragment is to be expressed as $P_n(w_{i+1}|w_i)$.

We expand/find protein name boundaries based on these probability estimates. In the case where there is not a name boundary between w_i and w_{i+1} (i.e., w_{i+1} is also a protein name fragment), $P_p(w_{i+1}|w_i)$ is expected to be greater than $P_n(w_{i+1}|w_i)$. Thus, we regard w_{i+1} as a protein name fragment if the following condition holds:

$$P_p(w_{i+1}|w_i) > P_n(w_{i+1}|w_i) \quad (1)$$

However, estimating these probabilistic parameters requires a large amount of training texts annotated with protein names, which are labor-intensive to create. To make matters worse, simply using a large corpus cannot be a substantial solution due to the characteristics of protein names: new protein names continue to be created. Previously unseen data are fatal for statistical inference.

To reduce an influence of the data sparseness problem, we generalize words (tokens) to word classes as shown in Table 1. They are automatically uniquely assigned to words (tokens).

class	examples
<i>suffix_in</i>	protein, oncoprotein, lactoferrin
<i>suffix_ase</i>	kinase, transferase, peptidase
<i>word</i>	the, a, an
<i>acronym</i>	CN, TrkA, USF
<i>arabic_num1</i>	1, 2, 3
<i>arabic_num2</i>	12, 76, 32
<i>roman_num</i>	I, II, III
<i>roman_alpha</i>	alpha, beta, gamma
<i>punctuation</i>	comma (,), period (.)
<i>symbol</i>), (, %, +

Table 1. An example of word classes.

Integrating the word classes to the probabilistic model, we define bigram class models as in Equation (2),

¹<http://www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhhelp.html#Stopwords>

where c_i denotes the word class of w_i .

$$\begin{aligned} P_p(w_{i+1}|w_i) &\stackrel{def}{=} P_p(w_{i+1}|c_{i+1}) \cdot P_p(c_{i+1}|c_i) \\ P_n(w_{i+1}|w_i) &\stackrel{def}{=} P_n(w_{i+1}|c_{i+1}) \cdot P_n(c_{i+1}|c_i) \end{aligned} \quad (2)$$

The probabilistic parameters shown in Equation (2) can be estimated based on corpora annotated with protein names. However, the models still contain raw words w_{i+1} , which is likely to cause the data sparseness problem. To avoid it, we use Witten-Bell smoothing [21] in estimating the probabilities of having words w_{i+1} from classes c_{i+1} , which we found to perform better.

Similarly, we apply this method to expand/find protein name boundaries leftward as well. The probability functions are defined as in Equation (3), where w_{i-1} and c_{i-1} denote the token preceding to the detected protein name fragment w_i and its word class, respectively.

$$\begin{aligned} P_p(w_{i-1}|w_i) &\stackrel{def}{=} P_p(w_{i-1}|c_{i-1}) \cdot P_p(c_{i-1}|c_i) \\ P_n(w_{i-1}|w_i) &\stackrel{def}{=} P_n(w_{i-1}|c_{i-1}) \cdot P_n(c_{i-1}|c_i) \end{aligned} \quad (3)$$

3.4 Filtering

Our ultimate goal is automatically extracting novel information associated with proteins and cancer from the literature, where protein name identification is a fundamental element whose performance will strongly affect the rest of the IE process. Although high recall and high precision are ideal, there is a trade-off between the two measures. In this context, it is desirable that we could choose which measure we prefer according to the purpose (i.e., high recall with low precision, high precision with low recall, or balanced). This can be done by restricting the system output based on some certainty measure that indicates the extent to which the detected protein names are likely to be actual protein names.

We are currently using the certainty score $C(\cdot)$ defined as in Equation (4), where $w_1 \cdots w_n$ denotes a sequence of tokens detected as a protein name, and $F(x)$ and $F_p(x)$ denote a frequency of x in training data and a frequency of x appearing as a protein name fragment, respectively.

$$\begin{aligned} C(w_1 \cdots w_n) &\stackrel{def}{=} \frac{1}{n} \sum_{i=1}^n P_c(w_i) \\ P_c(w_i) &\stackrel{def}{=} \begin{cases} \frac{F_p(w_i)}{F(w_i)} & \text{if } F(w_i) \geq 3 \\ \frac{F_p(c_i)}{F(c_i)} & \text{otherwise} \end{cases} \end{aligned} \quad (4)$$

The probability $P_c(w_i)$ will be high in the case where a token w_i is predominantly used as a protein name fragment in training text since $F_p(w_i)$ approaches to $F(w_i)$. In addition, when the frequency of w_i is small, instead we use the frequency of the word class because low frequency tokens are less reliable. We set the cutoff to 3.

The word classes used in computing the certainty score are basically the same as those used in protein name boundary expansion. However, only acronyms are treated differently: more specific word classes are given. For instance, HsMad1 will be associated with a word class *acronym_AaAa*. The suffix *AaAa* is derived as following: consecutive capital letters, small letters, and numbers are squeezed into one

character A, a, and 0, respectively; and then, if any, 0 in the end of strings is stripped. The assumption for this transformation is that protein name acronyms have some patterns in the usage of capital letters, small letters, and numbers.

4 Evaluation

4.1 Overview

To evaluate the effectiveness of our approach, we implemented a protein name identification system based on the probabilistic models (see Section 3) and conducted a series of experiments, in which our system was compared with the Yapex [5, 15] protein name identification system. There are three reasons this particular study was selected for comparison: according to our survey, Yapex is one of the state-of-the-art protein name identification systems, which is based on hand-crafted rules; the system is publicly available through a CGI program on the Web [4]; and the annotated corpora used for Yapex’s evaluation are also publicly available.

As mentioned above, we used the same corpora as Franzén et al. [5] and Olsson et al. [15] used for Yapex’s evaluation. The corpora consist of reference corpus and test corpus, which contain 99 and 101 MEDLINE abstracts, respectively. The reference corpus, which is annotated with 1,745 proteins, was used for training our probabilistic models and the test corpus, which is annotated with 1966 protein names, was used for evaluation.

4.2 Evaluation measures

Precision, recall, and F-score are used as evaluation measures. Precision is the number of protein names a system correctly detected, divided by the total number detected by the system. Recall is the number of protein names a system correctly detected, divided by the total number contained in the input text. F-score combines these measures, i.e., recall and precision, into a single score and is defined as in Equation (5).

$$F\text{-score} = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}} \quad (5)$$

For judgment of correctness, we use three criteria: exact, partial, and fragment matches. As for exact match, every fragment composing a protein name has to be detected correctly, whereas, for partial match, a detected protein name is counted as correct in the case where *any* fragments composing the protein name are correctly detected. For fragment match, the counting unit is a fragment; that is, each fragment composing a protein name is to be judged independently whether it is correctly detected or not.

4.3 Results and discussion

Overall performance

Table 2 shows the result of the comparative experiment. The values in the column “Yapex” are directly cited from the *Proteinhalt i text* (protein concentration in text) project homepage [4], and “*Prob*” denotes our system based on the probabilistic models. A threshold for the certainty score was empirically set to 0.25 in this experiment (see Section 3.4).

Table 2. A comparison between Yapex and our system on the test corpus.

evaluation criteria		Yapex	<i>Prob</i>
exact	recall	59.9	66.9
	precision	62.0	60.1
	F-score	61.0	63.3
partial	recall	81.4	86.0
	precision	84.3	77.2
	F-score	82.8	81.4
fragment	recall	76.2	75.6
	precision	75.8	74.3
	F-score	76.0	75.0

When compared to Yapex, our system obtained about 2–7 points lower precision irrespective of the criteria for judgment of correctness (i.e., exact, partial, and fragment matches), while our system constantly outperformed Yapex in terms of recall. Consequently, the F-scores of our system were found to be quite comparable to those of Yapex, despite the fact that our method does not rely on POS taggers or syntactic parsers as used in Yapex.

We evaluated our system on several criteria, i.e., exact, partial, fragment matches and recall, precision, and F-score. Which criteria is important depends on what purpose we use the system for. Considering our ultimate goal, that is, IE for the cancer-protein interaction, exact match would be important for distinguishing numbers of protein names and associating extracted information with them. Incidentally, high recall would be preferable in the case where comprehensive information is needed, while high precision would be desirable in the case where the reliability of information is important. We will show that higher precision is achievable by varying a threshold for the certainty score (see Section 3.4) in the end of this section.

Performance for compound terms

Since our method is focusing on name boundary expansion using class bigrams (collocations), our method is expected to be more effective particularly for compound protein names. To demonstrate the advantage, we evaluated Yapex and our system on compound protein names. Table 3 shows the result, in which Yapex’s result was obtained by submitting the test corpus to the CGI program on the Web (on March 27, 2003).

In the case where only compound protein names are considered, irrespective of evaluation criteria, our system greatly outperformed Yapex especially in exact match. This result indicates that our proposed probabilistic model is fairly effective in expanding and finding name boundaries for compound protein names, and that the word classes we used for generalization efficiently capture the characteristics of protein names to a large extent.

Filtering based on certainty

Lastly, the effectiveness of the certainty score (see Section 3.4) was evaluated. We varied a threshold for the certainty score, so as to draw a recall-precision curve in terms of exact match. Figure 2 shows the result.

Table 3. A comparison between Yapex and our system for compound protein names on the test corpus.

evaluation criteria		Yapex	<i>Prob</i>
exact	recall	53.2	59.0
	precision	49.7	63.7
	F-score	51.4	61.2
partial	recall	73.3	73.5
	precision	68.5	79.3
	F-score	70.8	76.3
fragment	recall	65.8	65.0
	precision	65.1	76.8
	F-score	65.4	70.4

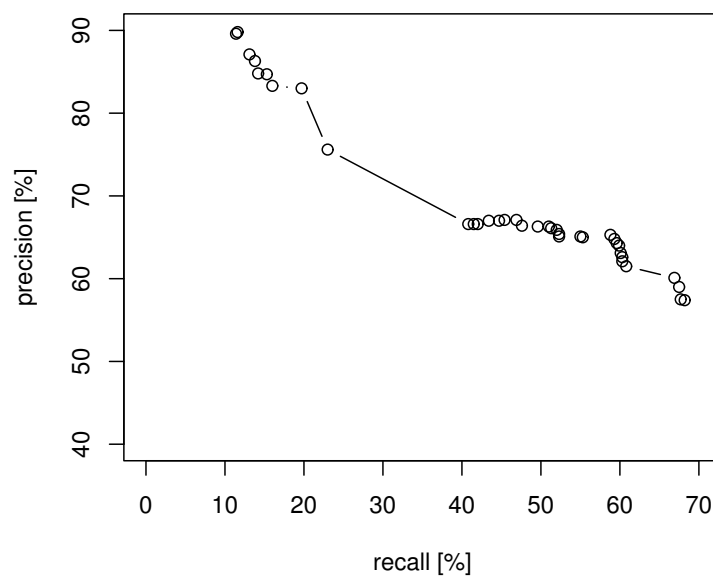


Figure 2. The relation between recall and precision for exact match.

The right most (and lowest) circle corresponds to the result without restriction (i.e., threshold is 0). As threshold increased, precision gradually increased until recall reached down around 40%. Then precision sharply increased up to around 90% with recall decreasing.

Although high precision was achieved, recall steeply dropped at the same time. To prevent recall from dipping, other features need to be integrated into the formula of certainty measure; for instance, surrounding words (contextual cues) may be effective.

5 Conclusions and Future Work

In this paper, we presented a method for identifying protein names in biomedical texts with an emphasis on protein name boundary expansion. Our method utilizes simple heuristics for initial detection of protein name fragments and takes advantage of a probabilistic model for expanding and finding protein name boundaries. The probabilistic model exploits surface clues reflecting characteristics of protein names, and combine word classes so as to avoid the data sparseness problem.

Our method, as opposed to the previous work, does not rely on POS taggers and/or syntactic parsers at all, since the information given by these NLP tools is not necessarily helpful for the task of protein name identification. This will reduce both processing overhead and probabilistic parameters to be estimated. We implemented a protein name identification system based on our method, and conducted comparative experiments to verify the effectiveness of our proposed method. The results demonstrated that our system performed well and is quite comparable to the Yapex protein name tagger which incorporates a syntactic parser. Moreover, in the case where only compound protein names were evaluated, our system showed higher precision than Yapex, especially in exact match. Furthermore, we proposed a notion of certainty to filter out erroneous identifications for improving precision; it was demonstrated to be effective to incrementally raise precision at the expense of recall.

Future work would include a refinement of the certainty measure. One possible extension will be to integrate contextual information, such as adjacent words or verbs governing the detected protein name candidates. Also, we are planning to automatically collect large-scale training corpus in order to further improve our system performance.

Acknowledgment

We would like to thank the members of the *Proteinhalt i text* project for letting us use their resources including tagged corpora and the Yapex protein name tagger.

References

- [1] L. A. Adamic, D. Wilkinson, B. A. Huberman, and E. Adar. A literature based method for identifying gene-disease connections. In *Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB2002)*, pages 109–117, 2002.
- [2] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.
- [3] N. Collier, C. Nobata, and J. Tsujii. Extracting the names of genes and gene products with a hidden markov model. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 201–207, 2000.

- [4] K. Franzén. Proteinhalt i text (protein concentration in text), 2003. Retrived March 27, 2003, from <http://www.sics.se/humle/projects/prothalt/>.
- [5] K. Franzén, G. Eriksson, F. Olsson, L. Asker, and P. Lidén. Exploiting syntax when detecting protein names in text. In *Workshop on Natural Language Processing in Biomedical Applications*, 2002.
- [6] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17 Suppl. 1:S74–S82, 2001.
- [7] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 3, pages 705–716, 1998.
- [8] R. Grishman and B. Sundheim. Message Understanding Conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 466–471, 1996.
- [9] L. Hirschman, J. C. Park, J. Tsujii, L. Wong, and C. H. Wu. Accomplishments and challenges in literature data mining for biology. *Bioinformatics*, 18(12):1553–1561, 2002.
- [10] J. Kazama, T. Makino, Y. Ohta, and J. Tsujii. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain*, pages 1–8, 2002.
- [11] M. Krauthammer, A. Rzhetsky, P. Morozov, and C. Friedman. Using BLAST for identifying gene and protein names in journal articles. *GENE*, (259):245–252, 2001.
- [12] M. Narayanaswamy, K. E. Ravikumar, and V. K. Shanker. A biological named entity recognizer. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 8, 2003.
- [13] S. Ng and M. Wong. Toward routine automatic pathway discovery from on-line scientific text abstracts. In *Proceedings of Genome Informatics*, volume 10, pages 104–112, 1999.
- [14] C. Nobata, N. Collier, and J. Tsujii. Automatic term identification and classification in biology texts. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium*, pages 369–374, 1999.
- [15] F. Olsson, G. Eriksson, K. Franzén, L. Asker, and P. Lidén. Notions of correctness when evaluating protein name taggers. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.
- [16] M. Palakal, M. Stephens, S. Mukhopadhyay, R. Raje, and S. Rhodes. A multi-level text mining method to extract biological relationships. In *Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB2002)*, pages 97–108, 2002.
- [17] D. Proux, F. Rechenmann, and L. Julliard. Detecting gene symbols and names in biological texts: A first step toward pertinent information extraction. In *Proceedings of Genome Informatics*, volume 9, pages 72–80, 1998.
- [18] T. Sekimizu, H. S. Park, and J. Tsujii. Identifying the interaction between genes and gene products based on frequently seen verbs in MEDLINE abstracts. In *Proceedings of Genome Informatics*, volume 9, pages 62–71, 1998.
- [19] L. Tanabe and J. Wilbur. Tagging gene and protein names in full text article. In *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain*, pages 9–13, 2002.
- [20] J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll. Automatic extraction of protein interactions from scientific abstracts. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 5, pages 538–549, 2000.
- [21] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, 1991.