Extracting Protein Names from Biological Literature

Huang-Cheng Kuo[1] and Ken-I Lin[2]

[1] Department of Computer Science and Information Engineering
National Chiayi University, Chia-Yi City, Taiwan
hckuo@mail.ncyu.edu.tw

[2] Department of Computer Science and Information Engineering
National Chiayi University, Chia-Yi City, Taiwan
x1457541@gmail.com

Abstract

Name entity recognition is an essential task in extracting biological knowledge. In biological corpus, protein names and other terminologies are mixed in natural language sentences. Sometimes whether an abbreviation is a protein name or not depends on the context. Protein names are often composed of gene names, cell names, or even drug names. Moreover, the number of newly coined protein names is increasing. Even with the assistance of a dictionary, it is still hard to correctly automatically identify all protein names in a biological corpus. We modify a hierarchical model of protein name tokens. On the one hand, we choose rule-base method to improve protein name recognition prediction accuracy rate. On the other hand, we use the N-gram language model to determine the boundary of protein name. Numerous studies mentioned that the hardest part is to identify abbreviations and words beginning with uppercase. In order to enhance the recognition performance, we use a dictionary to strengthen recognition for abbreviations and words beginning with uppercase. Experimental results show that about 10% increase in performance.We use YAPEX corpus and GENIA corpus datasets for experiment. In our study, an F-score can achieve 0.697 on the YAPEX corpus and 0.691 on the GENIA corpus. Finally, strengthening the abbreviation for part recognition, we use the Uniprot dictionary database to recognize, an F-score can achieve 0.797 on the YAPEX corpus and 0.806 on the GENIA corpus.

Keywords: *Name Entity Recognition, Protein Name Recognition, N-gram Language Model.*

1. Introduction

Bioinformatics is integration of database management, data capture, data inventory, analysis of engine development, web user interface and so on into a system. Utilizing the information got from bioinformatics is an important issue for researchers. Through the use of these information, it can save a lot of human effort in many geological related areas, such as the development of new drugs, gene therapy, explore the biological mechanisms of cancer research, protein-protein interactions, protein structure prediction. The information is structured and analyzed effectively by utilizing computer that is very helpful for researchers. Therefore, automated bioinformatics data mining is necessary.

Information extraction does not only recognize the importance entity and must be able to determine the relationship between entities [16]. In this study, information extraction is referred to correctly identify various biological field nouns, such as protein, protein family, genes, viroids, plasmids, organelles, bacteria, archaea and eukaryota, etc. To correctly identify the names and areas in complex corpus is a critical step. Then the relationship between the individual and the areas can be further understood by identifying complex name and areas.

Bioinformatics research often requires a very large amount of research literature for providing reference evidence. Information retrieval and information extraction therefore becomes a very important issue. Traditionally, information extraction has basic work items, such as named entity identification or (also known as proper nouns marked or named entity tagging), coreference resolution, scenario template [18].

Named entity identification or tagging is literally as the words. Coreference resolution is synonymous with its corresponding series proper nouns. Scenario templates are in accordance with pre-determined patterns, the files to retrieve information about a field populated templates. These three works is treated as a hierarchical relationship. It can just perform resolving of coreference resolution after named entity identification is complete, and then perform a scenario template of the record. A basic set of information retrieval system is composed of hyphenation modules, lexical analysis module and syntax module composed. Moreover, articles of different disciplines have their own special considerations, therefore the system must introduce different processing modules.

In bioinformatics corpus, there are often a large and complex proper nouns and natural language, therefore it is difficult to analyze due to the following reasons:

1. Bioinformatics proper noun is quite complicated, even it is the gene name, protein name, cell name, or the name of the drug that they are all the source of protein named entity. And new bioinformatics proper nouns increase gradually, such as Medline leading U.S. biomedical sites, it is gradually increase about 500,000 new pen data each year.

2. Old dictionary cannot effectively and immediately recognize the new words. Many new biological terms are not in the dictionary, which resulting in illegible cases. There is no complete dictionary which includes all areas of bioinformatics dictionary. System may not find the characteristics of new words in information retrieval process, which resulting in problem of identifying the new biological text correctly.

3. The same words can be represented in different biological areas or non-biological areas and so are abbreviations. For instance, “TCF” may refer to “T Cell Factor” or “Tissue Culture Fluid” in different articles. Whether “TCF” is a protein name or not depends on the context.

4. Most protein names are often composed of multiple short terms. The meaning of verb is possessed of the strong biological uniqueness. The information of lexical contexts is unobvious features, or data sparseness problems caused by insufficient training data, and so on. Therefore, the recognized results will be affected because of the fuzzy boundaries caused by the problems mentioned above in protein names.

5. So far, there are no common rules to express the protein names. Researchers often use abbreviations to name the protein. There is no common nomenclature for the represented abbreviation, and the represented abbreviation is varied variability. The same protein names may have many different ways of writing or description, e.g. “N-acetylcysteine”, “N-Acetyl-Cysteine”.

In our study, we propose an integrated approach, the use of hierarchical token, N-gram language model and rule-based method can solve the problems mentioned above. First, we use the hierarchical token for transferring protein names to token type and collect the style of most protein names. Assuming there exist a new word which can’t be recognized by the old dictionary. As long as the token type contains the style of the new word, then the system can perform recognition actions. Even if the training data is not enough, the training data can also collect the style of most protein names through the token type. Thus the sparse problem of data can be solved. In addition, the system can filter out weak patterns in the token type in which the most of the protein names can be recognized while the small part of them can’t be recognized and will be converted into a class type to recognize.

On protein name boundary detection problem, we still use the N-gram language model to determine the boundary. First, we retrieve the contextual features of proteins name, then statistical frequency of context features. Finally, we use this probability to recognize the boundaries of protein names, therefore the multi-word protein names boundaries can clearly divided.

In addition, we chose rule-base method to improve protein name recognition rate. In the manual part of rule-base method, we use the known non-protein names and define some rules to filter out those non-protein name words. In the automated part of rule-base method, we use the N-gram language model. After doing statistic the probabilities of protein name and non-protein name. Word which belongs to a protein name or non-protein name can be recognized after doing statistics analysis on these probabilities. Thus the system can filter out some of the weak patterns.

After completing above methods, we observed the recognition results. We found that most of misidentification causes occur in protein name styles of abbreviation or proper case. Therefore, in order to make the system more complete, we have chosen the Protein Data Bank (Uniprot database) to create a dictionary to enhance recognition for these problems. Finally, the system performance (F-score) is also significantly improved by about 10%.

2. Related Works

There are a lot of researches that focus on biomedical name recognition. These research methods generally can be divided into several categories such as rule-based methods, machine learning methods, dictionary-based methods, hybrid approach and so on. Each method has its advantages and disadvantages.

2.1 Machine learning methods

Machine-learning method is often used in natural language processing or biological name recognition [1,5,8,11,13,17] and generally has good performances. The common machine learning methods are including such as SVM (Support Vector Machine), ME (Maximum Entropy), and HMM (Hidden Markov Model), and so on. The advantage of machine learning is that the system can automatically learn and generate the best parameters, and then the most suitable features can be selected through the best parameters.

Machine-learning method is often with the dictionary, rules, preprocessing and post processing methods to help it learning. First, it use dictionary to build a training data and reuse Machine-learning methods to train the model, and then use some rules to improve the performance of the system or use the rule to find the name of fragments and extend their full names.

Cheng [1] is mainly of POS Tagging and combined with an SVM for improving the system performance. It achieves an F-score 73.8% on the Yapex corpus. Seki et al. [5] use probabilistic model, their system performance can be achieved F-score = 63.3%. Zhou et al. [8] present a named entity recognition system called PowerBioNE, all the features are through a hidden Markov model (HMM) and a HMM-based named entity recognizer. In addition, a k-Nearest Neighbor (k-NN) algorithm is proposed to resolve the data sparseness problem, their system performance can be achieved F-score = 66.6%. Ju et al. [11] choose SVM to identifying biological terms in GENIA corpus, they get the precision rate= 84% and recall rate=81% in total for the two categories classification problem. When meeting the multiple categories classification problem, SVM can identify biological terms accurately, but the recall rate is very low. BioTagger-GM [13] system use CRF and MEMM model to train through using dictionary lookup results as one type of feature, which can achieve an F-Measure of 0.8887 on BioCreAtIvE II Gene Mention (GM) corpus. Finkel et al. [17] present a system based on a maximum-entropy sequence tagger. It focuses on correctly identifying entity boundaries, and the use of several external knowledge sources is including full MEDLINE abstracts and web searches.

2.2 Rule-based methods

In study of biological text recognition, almost all studies are with rule-base methods [6,7,15]. The rule-base methods are generally divided into manual and automatic to build rules.

When creating rules via Manual, it needs to analyze on the features and characteristics of the protein names while automatically creating rules is often with machine learning method to build rules.
The disadvantage is very time-consuming to observe protein name to build rules, and it is difficult to take into account the comprehensive rules and also need more rules to achieve the desired effect. For instance, "HIV" is a name of the virus and the same with the length of the protein name "NGF" (nerve growth factor) as containing three uppercase letters, but "HIV" is not a protein name. Therefore it requires additional rules for assistance.

In the present day, there is no standard protein nomenclature rule. Therefore it can’t find out all the corresponding representation of protein names. The commonality of rule is not enough, the same rules can not apply to all databases, it may need to change rule depending on different data.

PROPER (PROtein Proper-noun phrase Extracting Rules) system [6] achieved 94.70% precision and 98.84% recall on a set of 30 abstracts using simple lexical patterns and orthographic features. Franzen et al. [7] introduced the YAPEX system that combines lexical and syntactic knowledge, and used heuristic rules and a document-local dynamic dictionary. YAPEX system can be achieved recall = 61.0% and precision = 62.0%. Tatar et al. [15] explore the Bigram language model and automatic rule learning method with any dictionary. They generalize protein names by using hierarchically categorized syntactic token types. Bi-gram model achieved F-score=67.7% on YAPEX and 66.8% on the GENIA; Rule learning method achieved F-score = 61.8% on YAPEX and 61.0% on the GENIA.

2.3 Dictionary-based methods

Most of the studies are based on a database [4,9,10,12,14]. They are through the database to build their own dictionary, and with other methods to recognize protein names. The advantage is to achieve high recognition performance. It does not have data sparse problem due to the dictionary as training data. The disadvantage is that the use of dictionary method requires frequent updates, new protein names are updated all the time. In addition, each person has his/her own ways of writing and habits. That people describe the same protein name may have a slightly different. For example, "CN-deuterohemin" and "CN-deutero-hemin", if using a dictionary to perform an exact match, it will result in unable recognition. Thus, a complete system is able to identify all styles of protein names that require other methods.

Ding [4] integrated the EMBL-EBI, PRI, UniProt, NCB dictionary into a complete training data. Then, with Minning Association and Sequential Pattern recognition proteins, their system can be achieved F-score = 74.5%.Chang [9] build gene and protein dictionary by EnterZ gene and BioThesaurus to provide the responsible gene ID and use various string transformations to match gene and protein names in literature. Hsieh [10] uses ABNER to recognize the protein names, which use a lexicon for labeling the cancer names and a bag of keywords as the features of cancer biomarker. LINNAEUS [12] uses a dictionary-based approach to identify species names and a set of heuristics to resolve ambiguous mentions. When compared against the manually annotated corpus, LINNAEUS performs with 97% precision and 94% recall at the mention level, and 98% recall and 90% precision at the document level. Schuemie et al. [14] combine two complementary methods for automatic generation of a comprehensive dictionary. In addition, they combine the gene and protein names with several existing databases of different organisms. The combined dictionaries show a substantial increase in recall on three different test sets as comparing to any single database.

2.4 Hybrid methods

Based on the advantages of various resources, methods, information, data, etc., a pluralistic recognize method is constituted [2,3].

PROTEX [2] system sets a simple heuristics and uses a probabilistic model for locating complete protein names. It avoids using of natural language processing e.g. POS or syntactic. It solely relies on surface clues so as to reduce the processing overhead. Wang et al. [3] use the Generalized Winnow algorithm, heuristic rules and a statistic method to detect the protein name and analyze the reliability of recognized protein boundary, which can be used for expanding protein boundary.

Most of the researches are through dictionary method with other methods to recognize. In our study, we propose two versions of the method to recognize protein names. The first is the use of hierarchical token, N-gram language model, rule-based method without any dictionary while the second is the first method with dictionary method, but the dictionary only recognizes for a capital letter at the beginning of a word or abbreviations.

3. Training and Testing Data

In our study, we choose Yapex corpus and GENIA corpus for data sets and UniProt database for dictionary.

Yapex corpus (http://www.sics.se/humle/project/prothalt) consists of 200 MEDLINE abstracts, 99 abstracts for training data and 101 abstracts for testing data. The corpus use specific tag to label protein name, protein individual, protein entity, or denote small groups of nearly identical proteins.

GENIA corpus: (http://www.nactem.ac.uk/genia/genia-corpus/term-corpus). The identification of linguistic expressions referring to entities of interest in molecular biology such as proteins, genes and cells is a fundamental task in biomolecular text mining. The GENIA technical term annotation covers the identification of physical biological entities as well as other important terms. GENIA corpus contains 1999 Medline abstracts, selected using a PubMed query for the three MeSH terms "human", "blood cells", and "transcription factors". The corpus has been annotated with various levels of linguistic and semantic information.

We choose the newest corpus (GENIA corpus version 3.2). It contains 17295 protein-related information. There are some other type tokens such as protein complex, domain or region, family or group of protein etc. There are some other name classes such as cell type, DNA, RNA, virus etc. except protein name class.

Uniprot(Universal Protein Resource): (http://www.uniprot.org/). Uniprot was established in 2002 by European Bioinformatics Institute(EBI), the Swiss Institute of Bioinformatics(SIB), and the protein Information Resource(PIR). Across the three institutes more than 100 people are involved through different tasks such as database curation, software development and support. Uniprot consists of three components:

1. UniProt Knowledgebase(UniProtKB) :
   Swiss-Prot: it is a protein sequence database, which is manually annotated and reviewed by human expert; we choose this database to build our protein name dictionary in our study. TrEMBL, which is

automatically computer annotated and is not reviewed.

2. UniProt Non–redundant Reference(UniRef):
Sequnce clusters, used to speed up sequence similarity searches.

3. UniProt Archive(UniParc):
UniParc contains only comprehensive and non-redundant protein sequences database. It combined many databases into one at the sequence level and searching UniParc is equivalent to searching many databases simultaneously.

4. Methods

Although the protein names does not have a standard rule for naming, most of they are named by reactant, chemical, physical characteristics, etc., to complete the protein naming. Most of the protein name composition may be in several short terms. We can observe the laws of some structural characteristics of the protein names.

4.1 Type of word

Since rule of naming protein is composed of a number of descriptive noun names, most of them contain uppercase letters, numbers and special symbols. It constructs a complete set of tokens, then it classifies the similar structure of protein name to the same category, and this category is defined as the same token type. In recognition protein name issue, although the word-type features can be summed up in many areas, they are still helpful for recognition result.

Due to the limited amount of protein name in the training data, so in the case of recognition of protein name without using any dictionary, the training data will cause data sparseness problems. Thus, many studies have mentioned that token conversion can effectively enhance the recognition rate and can solve the problem of sparse training data. However, most studies are based on the rule of thumb to determine the protein name belongs token categories and does not have a regular expression to represent all proteins name belongs token position.

Tatar et al. [15] proposed a new concept of hierarchical token conversion. They defined the regular expression to classify the token. We adopted this hierarchical token classification and the new definition of the 7 kinds of class types and 22 kinds of token types, as shown in Table 1. Thus, it makes classification of protein name more detailed and more effective to solve the problem of sparse training data.

Delimiter: This class type is used to represent punctuation and special symbols. It divided into three token types according to the frequency of occurrence, namely Frequent (e.g. “-”, “/”), Rare (e.g. “:”, “<”), Very Rare (e.g. “%”, “*”).

Single: This class type is used to represent digital or single character, which contains four token types, namely Roman Numeral (e.g. “V”, “XI”), Number (e.g. “24”, “1998”), Single Letter (e.g. “a”, “h”), Greek Letter (e.g. “alpha”, “beta”).

Table 1. Hierarchical conversion token

Class	Token	Regular Expression	Length
Delimiter	Frequent	[. ( ) – /]	No restrictions
Delimiter	Rare	[: { } < >]	No restrictions
Delimiter	Vary Rare	[% = ; , + ! ? *]	No restrictions
Single	Roman Numeral	[ivxdlcm]+|[IVXDLCM]+	No restrictions
Single	Number	[0–9]+	No restrictions
Single	Single Letter	[a–zA–Z]	No restrictions
Single	Greek Letter	alpha|beta|gamma|delta|epsilon|theta|kappa|lambda|sigma|mu	No restrictions
Abbreviation	Very Long Abbreviation	[a–zA–Z] + ([A–Z][a–z]* | [0–9]+) ([a–zA–Z] + | [0–9] + | [’])*	Length > 12
Abbreviation	Long Abbreviation	[a–zA–Z] + ([A–Z][a–z]* | [0–9]+) ([a–zA–Z] + | [0–9] + | [’])*	Length > 7; length < 13
Abbreviation	Abbreviation	[a–zA–Z] + ([A–Z][a–z]* | [0–9]+) ([a–zA–Z] + | [0–9] + | [’])*	Length > 3; length < 8
Abbreviation	Short Abbreviation	[a–zA–Z] + ([A–Z][a–z]* | [0–9]+) ([a–zA–Z] + | [0–9] + | [’])*	Length = 3
Abbreviation	Very Short Abbreviation	[a–zA–Z] + ([A–Z][a–z]* | [0–9]+) ([a–zA–Z] + | [0–9] + | [’])*	Length = 2
Bioregular	Long Frequent Type-1	[a–zA–Z] + (ase|gen)	Length > 8
Bioregular	Frequent Type-1	[a–zA–Z] + (ase|gen)	Length < 9
Bioregular	Frequent Type-2	[a–zA–Z] + (in)	No restrictions
Bioregular	Frequent Type-3	[a–zA–Z] + (al|um|ide)	No restrictions
Regular	Lower Case	[a–z][a–z’]+	No restrictions
Regular	Long Proper Case	[A–Z][a–z’]+	Length > 9
Regular	Proper Case	[A–Z][a–z’]+	Length > 3; length < 10
Regular	Short Proper Case	[A–Z][a–z’]+	Length<4
Sequence	ATCG Sequence	([A]+)|([C]+)|([T]+)|([G]+)	Length>4
Other	Other	No specific pattern	No specific pattern

Abbreviation: This class type is used to represent abbreviation of protein names, which contains both alphabetic and numeric characters. It divided into five token types according the length, namely Very Long Abbreviation (e.g.“KKLSMYGVDLHKAKDL”), Long Abbreviation (e.g.“FcepsilonRI”), Abbreviation (e.g.“CD23”“ PTHrP”), Short Abbreviation (e.g.“IgE”, “MBP”), Very Short Abbreviation (e.g. “E7”, “HA”).

BioRegular: This class type is used to represent word suffixed, namely Long Frequent Type-1 (tokens suffixed with “ase” or “gen” and have more than eight characters, e.g. “acetyltransferase”), Frequent Type-1 (tokens suffixed with “ase” or “gen”and have less than nine characters, e.g. “kinase”), Frequent Type-2 (tokens suffixed with “in”, e.g.
“apohemoglobin”), Frequent Type-3 (tokens suffixed with “al”, “um” or “ide”, e.g. “antiserum”).

Regular: This class includes a wide range of protein name or non-protein name, which contains English characters only, namely Lower Case (consists of all lowercase characters, e.g. “interferon”), Long Proper Case (only first letter is uppercase and have more than nine characters, e.g. “Accordingly”), Proper Case (only first character is uppercase, e.g. “Groucho”), Short Proper Case (only first character is uppercase, e.g. “Nck”, “Jun”).

Sequence: This class type is used to represent DNA sequences, namely ATCG Sequence (Sequence from the A C T G consisting of four characters, length must exceed 4, e.g. AAGCTTGGGT).

Other: This category labeled other categories which are not marked.

In our system, the priorities of protein name converted into token type are according to the sequence order in Table 1. Figure 1 shows the steps of protein name is converted into token type.
In order to solve the problems of multi-word protein name or complex patterns protein name, our system prepared two conversion types. The timing of conversion is according to whether the current type of word can be recognized or not. Eq. (1) describes the pseudocode for converting word into token type, where T_i denotes hierarchical token type of word W_i , and C_i denotes hierarchical class type of word W_i .

$$\begin{aligned}
&Do\ P(T_i \mid T_{i-1}) = P(W_i \mid W_{i-1}) \\
&If\ \ P(T_i \mid T_{i-1})\ can\,be\ recognized \\
&\qquad recognition; \\
&Else\ P(C_i \mid C_{i-1}) = P(T_i \mid T_{i-1}) \\
&\qquad recognition;
\end{aligned} \tag{1}$$

Figure 2 shows the type conversion samples. First, the protein names in the training data and testing data are converted into token type. After the pattern matching, we can observe in the test data. After “dkk1”, “Akt”, “Rnase” these three protein names are converted into token, each token style has been contained by training data. Thus it can be recognized successfully. However, the “presenilin-1” token style never appears in the training data. It therefore can’t be recognized.

We observe the system and find the reasons for error recognition and unable recognition. The reason is often that the combination of token is complex or the training data is not included due to the data sparseness problem. Therefore, if “presenilin-1” is converted into a class type, as shown in Figure 3, the training data can contain the pattern combinations of it, after the training data and test data are converted into a class type. The system will be able to successfully identify it.

Examples :
“LRP6”→ [Abbreviation]

“presenilin-1”→ presenilin [Frequent]1 → presenilin [Frequent] [Number] → [Frequent_Type2][Frequent][Number]

“ SAF-A” → SAF[Frequent]A → SAF[Frequent][Single_Letter] → [Short_Abbreviation][Frequent][Single_Letter]

Fig. 1 An example of protein name converted into token type

Traing set		Testing set	
Protein name	Token type	Protein name	Token type
PTHrP	[Abbreviation]	dkk1	[Abbreviation]
Wnt	[Short_Proper_Case]	Akt	[Short_Proper_Case]
GTPase	[Frequent_Type1]	RNase	[Frequent_Type1]
dismutase-1	[Long_Frequent_Type1][Frequent][Number]	presenilin-1	[Frequent_Type2][Frequent][Number]

Fig. 2 Protein names converted into token type

Traing set		Testing set	
Protein name	Class type	Protein name	Class type
PTHrP	[Abbreviation]	dkk1	[Abbreviation]
Wnt	[Regular]	Akt	[Regular]
GTPase	[Regular]	RNase	[Regular]
dismutase-1	[Bioregular][Delimiter][Single]	presenilin-1	[Bioregular][Delimiter][Single]

Fig. 3 Protein names converted into class type

On selective conversion, the advantage is that the system can filter out most of the non-protein names in token type. Therefore it can minimize the interference of recognition in class type. After converting into class type, multi-word protein name recognition results could become much better.

4.2 Core word features

Protein name may be composed of single word abbreviation or multiple short description words. Most of the protein names may have a core word and other short-terms are around the core word. In our study, the uni-gram probability model Eq. (2) and Eq. (3) are used to calculate the frequency of occurrence of core words. Then, high-frequency core words are probably protein names. Where c is the meaning of the count.

$$P(W_i)=P(W_1,W_2,W_3..W_i)=P(W_1)P(W_2)P(W_3)..P(W_i) \qquad (2)$$

$$P(W_i)=c(W_i)\Big/\sum_i c(W_i) \qquad (3)$$

4.3 Contextual features

If we can successfully find the position of protein name core word, the frequency of occurrence of context can be calculated by the bi-gram probability model Eq. (4) and (5). The possible protein name fragments can be also found out. Then protein name boundaries can successfully be recognized.

$$P(W_i)=P(W_1,W_2,W_3..W_i)=P(W_1)P(W_2|W_1)P(W_3|W_2)..P(W_i|W_{i-1}) \qquad (4)$$

$$P(W_i)=c(W_{i-1},W_i)\Big/c(W_{i-1}) \qquad (5)$$

4.4 Positive and negative features and selected boundaries

There will be lot of errors as only contextual feature is used to recognize protein name. Therefore, we used the training data of corpus to calculate the probability of the protein names and non-protein names Eq. (6) and (7). After calculating the probability, if the protein name probability is greater than the non-protein name probability, the reliability of this rule could be set to be high. On the contrary, be set to be low. Thus, the weak rule will be excluded via Eq. (8).

$$P_p(W_{i+1}|W_i) \qquad (6)$$

$$P_n(W_{i+1}|W_i) \qquad (7)$$

$$P_p(W_{i+1}|W_i)>P_n(W_{i+1}|W_i) \qquad (8)$$

Figure 4 shows an example of the calculation of the probability. P_p is the probability function of protein name, P_n is the probability function of non-protein name. When the protein name probability is greater than the non-protein name probability, the condition will be true and the system will involve this rule in estimation.

4.5 Rule filter

In our system, we define some rules to increase the recognition rate by filtering out the words which is the known non-protein name. For example, words are with a capital letter at the beginning followed by more than three lower case letters (e.g. There, According, Although), units (e.g. mH, nM, pH, pI, mM), disease drug or compound (e.g. RU486, AIDS, ATP, RNA, DNA), punctuation mark (e.g. ",", ",", ":"), consecutive numbers (e.g. 1998,2001,2012), chemical formulas (e.g. CaCl2, NH2, Ca2, HCl and Mg2), Names (e.g. Kim et al.).

4.6 Using dictionary to recognize the abbreviation and proper name

In general, a complete bioinformatics corpus contains words of many different biological fields. These words are only partially related to proteins name, and the performance of protein name recognition often depends on the abbreviations or uppercase at the beginning of the words. Therefore, after we have completed the above method, we use Uniprot (Universal Protein Resource) database to create a dictionary and strengthen recognition for these proper nouns, abbreviations or capitalized at the beginning of word by using the dictionary method.

5. Results

5.1 Corpus and evaluation methods

In order to objectively assess the performance, we choose two corpuses (YAPEX corpus and GENIA corpus 3.02) for
cross-validation in our experiment. YAPEX consists of 200 MEDLINE abstracts, 99 abstracts for training data and 101
abstracts for testing data. GENIA corpus 3.02, in which the related protein name tag contains more than 17,295 test data.

In this study, we use precision Eq. (9), recall Eq. (10), and F-score Eq. (11) to evaluate the performance. In pattern recognition and information retrieval, precision (or known positive predictive value) is the fraction of retrieved instances that are relevant, recall (or known as sensitivity) is the fraction of relevant instances that are retrieved.

$$precision = \frac{T_p}{T_p + F_p} \qquad (9)$$

$$recall = \frac{T_p}{T_p + F_p} \qquad (10)$$

$$F - score = \frac{2 * precision * recall}{precision + recall} \qquad (11)$$

Figure 5 shows a comparison of our method with other methods in the YAPEX and GENIA corpus. Our approach F-score can achieve 69.5% without any dictionary. In addition, F-score can achieve 80.6% by using a dictionary to recognize abbreviations and uppercase at the beginning the words.

5.2 Hierarchy usage

Figure 6 shows the change in performance of the recognition system after using hierarchy token. In Figure 6, we can see that Precision of S. Tatar's system only has the reduction of 1.4% after using hierarchy token while Recall has been increased by 58.5%. In addition, in our system, Precision has been reduced by 2.8%, Recall is increased by 9.7%.

P_p (protein-tyrosine | phosphatase)

= P_p ([Frequent_Type2][Frequent][Lower_Case] | [Long_Frequent_Type1])

=2/16 =0.1250

P_n (protein-tyrosine | phosphatase)

= P_n ([Frequent_Type2][Frequent][Lower_Case] | [Long_Frequent_Type1])

=0/18 =0

Fig. 4 An example of the calculation of the probability

Corpus	Methods	Precision	Recall	F-score
YAPEX	YAPEX	62%	59.9%	61.0%
YAPEX	Tatar S et al.(bigram)	63.3%	71.8%	67.3%
YAPEX	Our method unused dictionary	67.7%	71.9%	69.7%
YAPEX	Our method used dictionary	84.9%	75.1%	79.7%
GENIA V3.2	Our method unused dictionary	65.6%	73.1%	69.1%
GENIA V3.2	Our method used dictionary	85.4%	76.3%	80.6%

Fig. 5 The results of various experiments

Method	Hierarchy	Number of token/class type	Precision	Recall	F-score
Tatar S et al.(Bi-gram)	No	21 token types	64.7%	13.3%	22.1%
Tatar S et al.(Hierarchy + bi-gram)	Yes	21 token types + 5 class types	63.3%	71.8%	67.3%
Our method(N-gram+ rule)	No	22 token types	70.5%	62.2%	66.09%
Our method(N-gram+ rule)	Yes	22token types + 7 class types	67.7%	71.9%	69.7%

Fig. 6 The results of hierarchy usage

After comparing both of two methods, we found that there is a slight variation in recall values since we used other rule-base method on pre-processing. The purpose is to make the performance more nearly stable. The purpose of using hierarchy toke is to recognize more complex patterns. In summary, after using hierarchical token, the performance of recognition can be indeed improved.

5.3 Analysis

Based on the above experimental results, we observed precision value and recall value. We analysis the reasons of the erroneous recognition and unable recognition. We find out the following types of problems:

1. Non-protein name is identified as a protein name: This situation will result in decreasing of precision values after the non-protein name has been converted to token. The token combination is too similar with protein name combination; if this rule can be excluded, precision will increase, but recall will decrease. Therefore, the system will take the highest F-score value as the balance.

2. Protein name is identified as a non-protein name: This situation will result in decreasing of recall values. It usually occurs when the training data has not appeared or the contextual features are too common or not enough. In addition, the context of core word may be weak token (e.g. [Lower case]). In the method 3.4 (Positive and negative features and selected boundaries), the system will exclude weak rule when it compares protein name probability to non-protein name probability. However, there are some the protein names may be included in a weak rule. Some of the protein name can't be recognized. It therefore causes boundary errors.

6. Conclusions

Feature selection and feature classification determine the performance of bioinformatics text recognition system. In our approach, we plan several feature selection methods and extend feature classification method. We also define more token types so that the protein name can achieve more detailed classification. And the rule base method is used to strengthen recognition performance. We compare our experimental results with other experimental results. The recognition performance can be reaching higher.

In feature selection, we found that the common difference between the protein name and the general terms is the word type feature, POS feature and lexical feature. In this study, word-type features have been resolved through the feature classification method. On the POS feature and lexical feature, we do analysis through following methods:

1. Add more information or dictionary, it makes more explicit on POS or lexical feature.

2. Add the paragraph features and the full text features, or choose other more distinctive features.

3. Add more recognition algorithm on paragraph feature, full text feature, or other feature.

This study currently adopts probability method, rule base method, and dictionary method. The ideas and approaches are more intuitive and simple and it can achieve good performance. Based on the results of known and unknown protein name recognition, it can also achieve good efficacy.

However, the drawback is that it only adjusts the parameters for the feature with larger impact and will have a better performance. The adjustments have slight efficacy for the features with small impact.

According to these conclusions, we hope to be able to improve small impact feature in the future, and develop more features algorithms to achieve a more complete protein name recognition system.

References

[1] Y.C. Cheng, Extracting Protein/Gene Names from the Biological Literatures, Master Thesis, Computer Science and Information Engineering, National Central University, Taiwan, 2005.

[2] Kazuhiro Seki, and Javed Mostafa, "A Hybrid Approach to Protein Name Identification in Biomedical Texts," Information Processing & Management, Vol. 41, No. 4, 2005, pp. 723–743.

[3] H.C. Wang, and T.J. Zhao, "A Hybrid Strategy to Protein Name Recognition," Intelligent Control and Automation, 2008, pp. 627-632.

[4] S.H. Ding, PNRS-Protein Name Recognition System Using Dictionary-Based Pattern Search Mining, Master Thesis, Computer Science and Information Engineering, National Chi Nan University, Taiwan, 2008.

[5] Kazuhiro Seki, and Javed Mostafa, "A Probabilistic Model for Identifying Protein Names and their Name Boundaries," IEEE Computer Society Bioinformatics Conference, 2003, pp. 251-258.

[6] K. Fukuda, T. Tsunoda, A. Tamura, T. Takagi, "Toward Information Extraction: Identifying Protein Names from Biological Papers," Pac Symp Biocomput, 1998, pp.707-718.

[7] Kristofer Franzen, et al., "Protein Names and How To Find them," International Journal of Medical Informatics, Vol. 67, No. 1, 2002, pp. 49-61.

[8] G.D. Zhou, et al., "Recognizing Names in Biomedical Texts: a Machine Learning Approach," Bioinformatics, Vol. 20, NO. 7, 2004, pp. 1178-1190.

[9] C.K. Chang, G-Norm Automated Pop-up PubMed Assistant Based on Two-phase Gene Normalization Approach, Master Thesis, Computer Science and Information Engineering, National Cheng Kung University, Taiwan, 2007.

[10] C.H. Hsieh, Retrieving Cancer Biomarker Related Evidence Sentences from Biomedical Literature, Master Thesis, Computer Science and Information Engineering, National Cheng Kung University, Taiwan, 2008.

[11] Z.F. Ju, M.C. Zhou, F. Zhu, “Identifying Biological Terms from Text by Support Vector Machine,” IEEE Industrial Electronics and Applications, 2011, pp. 455-458.

[12] Martin Gerner, Goran Nenadic, Casey M Bergman, “LINNAEUS: a Species Name Identification System for Biomedical literature,” BMC Bioinformatics, Vol. 11, No. 1, 2010, pp. 1471-2105.

[13] Manabu Torii, et al., “BioTagger-GM: a Gene/Protein Name Recognition System,” Journal of the American Medical Informatics Association, Vol. 16, No. 2, 2009, pp. 247-255.

[14] Martijn J. Schuemie, et al., “Evaluation of Techniques for Increasing Recall in a Dictionary Approach to Gene and Protein name Identification,” Journal of Biomedical Informatics ,Vol. 40, No. 3, 2007, pp. 316-324.

[15] Tatar, Serhan, and Ilyas Cicekli, “Two Learning Approaches for Protein Name Extraction,” Journal of Biomedical Informatics, Vol. 42, No. 6, 2009, pp. 1046-1055.

[16] John Kontos and Polyxene Kasda, and Hellas Athens. “Text Mining and Image Anomaly Explanation with Machine Consciousness,” Advances in Computer Science: an International Journal, Vol. 2, No. 5, 2013, pp. 35-39.

[17] Jenny Finkel, et al., “Exploring the Boundaries: Gene and Protein Identification in Biomedical Text,” BMC Bioinformatics, 2005, 6(Suppl 1): S5.

[18] K.H. Chen, “Organization and Extraction for Information,” Journal of Library and Information Studies, Vol. 12, 1997, pp. 127-141.