Various Features with Integrated Strategies for Protein Name Classification

Budi Taruna Ongkowijaya, Shilin Ding, and Xiaoyan Zhu

State Key Laboratory of Intelligent Technology and Systems (LITS),
Department of Computer Science and Technology, Tsinghua University,
Beijing, 100084, China
wwx01@mails.tsinghua.edu.cn, dingsl@gmail.com
zxy-dcs@tsinghua.edu.cn

Abstract. Classification task is an integral part of named entity recognition system to classify a recognized named entity to its corresponding class. This task has not received much attention in the biomedical domain, due to the lack of awareness to differentiate feature sources and strategies in previous studies. In this research, we analyze different sources and strategies of protein name classification, and developed integrated strategies that incorporate advantages from rule-based, dictionary-based and statistical-based method. In rule-based method, terms and knowledge of protein nomenclature that provide strong cue for protein name are used. In dictionary-based method, a set of rules for curating protein name dictionary are used. These terms and dictionaries are combined with our developed features into a statistical-based classifier. Our developed features are comprised of word shape features and unigram & bi-gram features. Our various information sources and integrated strategies are able to achieve state-of-the-art performance to classify protein and non-protein names.

1 Introduction

Biomedical literature has become a vast dataset that is in urgent requirement for automatic knowledge discovery to improve the effectiveness and efficiency of knowledge use. Nowadays, Named Entity Recognition (NER) is proved to be fundamental in information extraction and understanding in biomedical domain. Based on the method, the NER system can be roughly split into three categorizes: rule-based methods [1-2], dictionary-based methods [3], and statistical-based methods [4-7], although there are also combination of dictionary-based and rule-based method [8].

Dictionary based method is intuitive and effective in building annotated corpus, but it is in direct correlation with the completeness of dictionaries and fails to handle inconsistency in naming. Rule-based method relies on a set of expert-derived rules and has a high precision. But, it is domain-specific and usually hard to maintain and adapt to other areas. Statistical-based method is an alternative to those dictionary and rule based methods. This method is more flexible in environment adaptation but needs a large annotated corpus.

In this paper, we investigate the extent to which different feature sources and strategies contribute towards the task of classifying protein name and non-protein name. The classification task is the second task after named entity has been identified.

G. Chen et al. (Eds.): ISPA Workshops 2005, LNCS 3759, pp. 213 – 222, 2005.

Separating NER into two tasks provides a more accurate and efficient system because strategies and the relevant sources used in classification task is different than in identification task. Another reason for considering the classification task independently is that information extraction needs not to be limited to protein-protein interaction. Other types of information extraction also require name recognition. By only adjusting its feature sources environment, system architecture can be modified to classify another type of name. In addition, because classification task exploits various features and strategies, it can improve the performance of the entire process.

This paper is organized as follows. Section 2 briefly introduces feature sources for classification. Section 3 presents the idea of integrated strategies and approaches in detail. Section 4 describes our experimental results. Conclusion is presented in section 5.

2 Feature Sources for Protein Name Classification

To classify a protein name, both internal and external information should be considered [7]. Internal information is the information within the named entity, while the external information is information outside the named entity like nearby words and context occurrences. In addition, we also present our own feature sources including Fuzzy Word Shapes, Unigram and Bi-gram which contribute a lot to the classification task.

2.1 Internal and External Information

Compared to external information, internal information is a stronger factor to distinguish named entity. This feature can be collected using the most commonly occurred words from biomedical corpora. Words like “protein”, “kinase”, “alpha”, “receptor” and “factor” usually indicate the possible presence of protein names. These words are described as functional terms (f-terms) features, which we borrowed from Fukuda *et al* [1]. In addition, suffix and prefix are also good indications of the presence of protein and non-protein names, like “-ase” in “alkaline phosphatase”.

External information is provided in case of failure in extracting features from internal information. This information has been used for the task of word sense disambiguation (WSD), and is called contextual information. Observed on many researches, words that are close-by in location tend to have stronger predictive power for WSD. Therefore we include the external information by limiting the distance of the words. In addition we limit the external information to only nouns and adjectives.

2.2 Our Additional Sources

To improve the poor performance of above information when dictionary inquiries failed, unigram and bi-grams features which are calculated based on their statistical probability of training data to predict how strong they are related to protein name are used. These features are aimed to provide statistical information which is rarely captured in previous features. These unigram and bi-gram features significantly boost up our system performance.

In other hand, we introduce another surface feature, named *fuzzy word shape* features, to provide additional word shape information. This fuzzy word shape features implement simple fuzzy set [9] for computing the confidence score. Fuzzification process brings ascender/descender information, position of digit, position of capital, number of intersection in center words, number of vowels in word, number of consonants in word. Each character which appears in [bdfhkl] will be counted as ascender and each character appears in [gjpqy] will be counted as descender. Other characters than those which are in range of [a-z] will be counted as middle character. For each special character which appears in our shape focus, we calculate according to their position (Pos_{begin} , Pos_{middle} , and Pos_{end}) in word. Pos_{begin} is a first character based on the category. For instance: In category "ascender", then Pos_{begin} is the first position of ascender in the given name. The definition of Pos_{middle} and Pos_{end} is similar to Pos_{begin} :

$$Pos_{begin} = \underset{c_i \in type}{Min}\,(Pos(c_1), Pos(c_2), ..., Pos(c_n)) \tag{1}$$

$$Pos_{end} = \underset{c_i \in type}{Max}\,(Pos(c_1), Pos(c_2), ..., Pos(c_n)) \tag{2}$$

$$Pos_{middle} = \underset{c_i \in type}{Avg}\,(Pos(c_1), Pos(c_2), ..., Pos(c_n)) \tag{3}$$

where $Pos(Z)$ is a function returning position of character Z in word starting at 0. *Type* is our special character type {ascender, descender, capital, digit, symbol}. In addition, geometric features which calculate number of ascender, descender, middle, digit, symbol and intersection in word are also presented. After being extracted, these features are then normalized relatively to the length of the word. Finally, we got f in $[0.0..1.0]$. For calculating number of intersection, we use Table 1 to indicate number of intersection for each character.

Table 1. Intersection number in character

a	b	c	d	e	f	g	h	i	j	k	l	m
3.0	2.5	1.0	3.0	2.0	2.0	3.0	3.0	1.0	1.0	3.0	1.0	5.0
n	o	p	q	r	s	t	u	v	w	x	y	z
3.0	2.0	3.0	3.0	2.0	1.0	1.0	3.0	2.0	3.0	2.5	3.0	1.0

To clarify our word shape feature extraction, we take the word "Interleukin 2" as example. Table 2 shows our word shape feature representation of "Interleukin 2".

There are three areas in Figure 1 (Begin, Middle, and End). Once the special character hits on the beginning of the word, it will get a high score in the "Begin" area. Similar rules are applied to "Middle" and "End" area. These positional rules are applied for special characters which are defined using knowledge of protein nomenclature principles. Theses principles reveal that positional information of special cha-

Table 2. Word Shape Features for word "Interleukin 2"

Type Features	Pos	Val	Type Features	Pos	Val
Num Capital	-	0.07	Pos Capital	Begin	1.00
Num Vocal	-	0.38		End	0.00
Num Consonant	-	0.46		Middle	0.00
Num Ascender	-	0.23	Pos OtherChar	Begin	0.00
Num Descender	-	0.00	(not in range [a-z])	End	0.68
Num Middle	-	0.61		Middle	0.32
Num Digit	-	0.07	Pos Ascender	Begin	0.70
Num OtherChar	-	0.07		End	0.22
Num Intersection	-	0.52		Middle	0.76
Pos Digit	Begin	0.00	Pos Descender	Begin	0.00
	End	1.00		End	0.00
	Middle	0.16		Middle	0.00

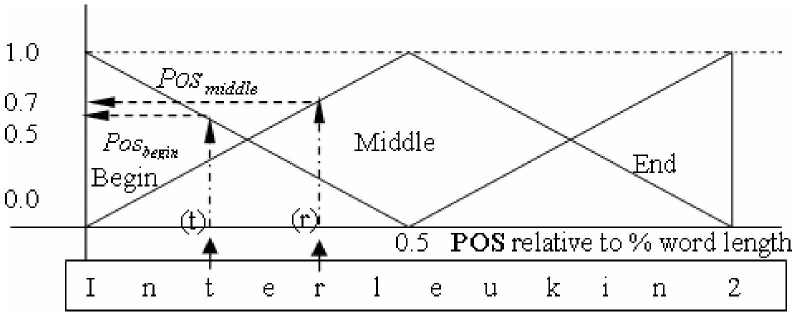

Fig. 1. Fuzzy Membership Position for 't' and 'r' in "Interleukin 2"

racters (ex. capital, digit, dash) has a special contribution to the classification task. Rules on position can be formulized into formula as follows:

$$PosNew_{begin} = FMember(Pos_{begin}, -\infty, 0.0, 0.5) \tag{4}$$

$$PosNew_{end} = FMember(Pos_{end}, 0.5, 1.0, \infty) \tag{5}$$

$$PosNew_{mddle} = FMember(Pos_{middle}, 0.0, 0.5, 1.0) \tag{6}$$

where Pos_{begin}, Pos_{end}, and Pos_{middle} are computed from formula (1, 2, and 3). In "Middle" case, function $FMember(pos, left, middle, right)$ returns a value of ~1.0 if the "pos" value is near to middle value; otherwise it will return a value of ~0.0 if "pos" value is near to "left" or "right". This function is a simple implementation of fuzzy membership function. After applying these fuzzy rules, we have position features

PosNew which provides better representation of features. The word shape feature extraction result can be seen on Table 2.

3 Methods

Stated in previous section, we try to incorporate the advantages of three methods and various features in our integrated strategies. In rule-based method, terms and knowledge of protein nomenclature are used. In dictionary-based method, rules to preserve dictionary list are used. For statistical-based method, SVM classifier which has been proved outstanding in biomedical domains for NER systems is used. Through the integrated strategies, a high performance can be achieved for classification task. Consider classification as a second task, which a score from identification task will be propagated into this task, hence a high confidence score is eagerly needed on this classification task.

3.1 Construction of Dictionaries

Ten dictionaries are constructed for classification tasks. They are one f-terms dictionary, one suffixes/prefixes dictionary, two external feature dictionaries (left context words dictionary and right context words dictionary), one in-context words dictionary, one protein names dictionary, one –in words with negative ending dictionary from NLProt (Mika *et al* [4]), two unigram dictionaries and two bi-grams dictionaries.

F-terms are taken into normalization by lower casing. This dictionary is manually collected on many papers using knowledge of experts. Words which are positively tied to classify class of protein names and non-protein names are used. Some of our f-term dictionary is shown in Table 3:

Table 3. Example of our f-term dictionary lists

Example of f-term dictionary lists					
factor~	receptor~	site~	vitamin~	region~	cell~
system~	sequence~	virus~	messenger~	element~	portion~
events~	system~	state~	motif~	particle~	kinase~
activit~	promot~	pathway~	complex~	protein~	enzym~

Morphological features as suffix and prefix are considered as important terminology cue for classification and have been widely used in biomedical domain. Similar to Zhou *et al* [5], we use statistical method to get the most frequent suffixes and prefixes from training data as candidates. Then, each of those candidates is sorted using formula below:

$$Morph - score(X) = (IN(X) - OUT(X)) / (IN(X) + OUT(X)) \tag{7}$$

where IN(X) is number of candidate X appearing in protein names and OUT(X) is number of candidate X in non-protein names. Then we manually selected the candidates over a threshold using expert knowledge.

Our external features dictionaries are taken from left context and the other is taken from right context. Both dictionaries are collected from training data as candidates which are limited only in adjective and noun words. Tokenization rules described in section 3.2 are used to extract these candidates. For these features we limit the number of words from environment to only 5 from left and 5 from right.

In-context features dictionary is similar to the work of Lee *et al* [6]. The most right 3 words from name in the training data are collected as candidates. These candidates are normalized using tokenization rules which are described in section 3.2.

Protein names dictionary is also collected from training data as candidates. This dictionary is merged with protein name dictionary we have extracted from SWISSPROT. Both dictionaries (generated from training data and SWISSPROT) are tokenized using tokenization rules in section 3.2

For unigram and bi-gram dictionaries, we apply differently with other dictionaries which we have described above. We only filter out the stop words and normalize white space in candidates. Tokenization step using tokenization rules on the candidates is not applied because of consideration on original shape information of candidates. For unigram we have two dictionaries. One contains protein names and the other does not contain protein names. Similar to our unigram dictionary, our bi-gram dictionaries are constructed.

3.2 Curating Dictionary

We curate all words in dictionary and provide a protein names curate-dictionary using tokenization rules. These tokenization rules consider the variability in writing such as hyphen, white space, capital, bracket, slash, numeral digit and special word like alpha, beta, gamma, kappa, etc. An example of tokenization process is shown below:

Sentence: IL-2 gene expression and NF-kappaB activation through CD28 requires reactive oxygen production by 5-lipogenase

Tokens: [il] [<N>] [gene] [express] [nf] [<M>] [b] [activation] [cd] [<N>] [require] [reactive] [oxygen] [production] [<N>] [lipogenase]

3.3 Simple Dynamic Matching

Our simple matching based on regular expression is implemented to search sub-string matching in a word or a sentence. This matching algorithm uses dynamic programming technique and is more flexible. It tries to search all combination which matches a source word/sub-word with a destination word/sub-word. The '~' symbol implements the '*' symbol in Regular Expression (RE) on Finite Automata, and means there can be none or some of characters to fill the '~' symbol. The details of this RE matching can be seen in example below:

Examples of match word:	Details of our RE models:	Curation process:
"inter~" → Interleukin	Observed word :	
"inter~in~ → Interleukin	"Interleukin-2"	Interleukin-2
"~ase" → kinase	Valid RE:	→ interleukin <N>
"~cept~" → receptor	inter~in~<N>	

Our experiment shows that using this model is better than using relaxed string matching algorithm. The result of relaxed string matching can be seen in Table 4, which contains *sm*(x) function.

3.4 System Design

Because we only classify protein and non protein names, we employ one vs. rests classifier which is the basic model of SVM. For our external information, we use 10 tokens (5 from left environment, and 5 from right environment) which have been already filtered using our tokenization rules. For environment tokens, we put a weight value based on their distance to our "observed name". On features which are related to dictionaries, only f-terms and suffixes/prefixes features are extracted using our simple matching algorithm, the rests are using exact matching. Fuzzy word shape features are extracted using procedure described in section 2.2.

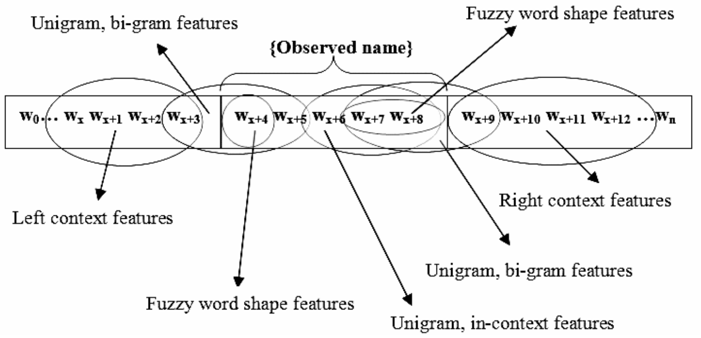
Unigram, bi-gram features
{Observed name}
Fuzzy word shape features
w0 ... wx wx+1 wx+2 wx+3
wx+4 wx+5 wx+6 wx+7 wx+8
wx+9 wx+10 wx+11 wx+12 ... wn
Left context features
Right context features
Unigram, bi-gram features
Fuzzy word shape features
Unigram, in-context features

Fig. 2. Sequence words based on their feature extraction type

Figure 2 shows from which words these features are extracted. For example, *fuzzy word shape* features are extracted from first word and the last two words in the name. For those features which are related with environment (<left-contexts, right-contexts, in-context, fuzzy word shapes, unigrams, bi-grams>) we employ our weighting feature method that consider the distance between the name and the target word. After all features have been extracted, we assign a value based on the following formula:

$$feature_{type_i} = \begin{cases} X & \text{, if exist/computable} \\ 0 & \text{, otherwise} \end{cases} \tag{8}$$

where $feature_{type_i}$ is a type of features, and i refers to the specific element in each type. There are only 3 types. The first is feature which is related with dictionary. The X value for this type is 1.0 if the word is inside the word list in the dictionary, otherwise 0.0. The second type is fuzzy word shape feature. The X value for this type is discussed in 2.2. The last is unigram/bigram feature type. These features are taken into SVM classifier to train our model.

4 Experimental Setup

The experiments were conducted using Genia Corpus 3.02 developed by University of Tokyo. We use SVM ^{light} developed by Joachims, T. as our classifier. We are reporting all of our experiment results which influence us to design such features extraction models. Each step of our research is shown in following table:

Table 4. Experiment in adjusting and adding features on 25% Genia corpus

Experiments	Acc	Prec	Rec	F-sco
Baseline -> ft+in+lf+rf	83.57	77.41	69.88	73.45
ft+sp	77.45	73.60	97.71	83.96
ft+sp+in+lf+rf	84.27	78.71	70.42	74.33
ft+sp+in+lim(lf+rf)	84.54	80.23	69.29	74.36
ft+sp+tok(in)+lf+rf	84.38	79.54	69.64	74.26
ft+sp+tok(in+lf+rf)	85.46	79.56	74.11	76.74
ft+sp+tok(in+lim(lf+rf))	86.10	81.14	74.31	77.57
ft+sp+tok(in+lim(fil1(lf+rf)))	85.89	80.41	74.56	77.37
ft+sp+tok(in+lim(lf+rf))+fws1	86.08	81.50	73.72	77.42
ft+sp+tok(in+lim(fil1(lf+rf)))+fws1	87.45	81.86	78.64	80.22
ft+sp+tok(in+lim(fil2(lf+rf)))+fws1	86.85	81.21	77.21	79.16
fws3	79.59	82.46	84.08	83.26
ft+sp+in+lim(fil1(lf+rf))+fws3	87.07	85.30	72.54	78.40
ft+sp+tok(in)+lim(fil1(lf+rf))+fws3	88.25	84.23	78.35	81.18
ft+sp+in+tok(lim(fil1(lf+rf)))+fws3	88.06	82.91	79.48	81.16
ft+sp+tok(in+lim(fil1(lf+rf)))+fws3	88.84	84.93	79.63	82.19
fws3+ix	79.61	82.49	84.08	83.28
ft+sp+tok(in+lim(fil1(lf+rf)))+fws3+ix	88.89	85.03	79.68	82.27
sm(ft)+sp+tok(in+lim(fil1(lf+rf)))+fws3+ix	87.90	84.83	76.23	80.30
ft+sp+fws3+ix	84.20	85.89	88.34	87.10
ft+sp+tok(in+lim(fil1(lf+rf)))+fws3+ix+ug	95.84	94.12	92.96	93.54
ft+sp+tok(in+lim(fil1(lf+rf)))+fws3+ix+bg	91.93	89.53	84.99	87.20
ft+sp+tok(in+lim(fil1(lf+rf)))+fws3+ix+ug+bg	96.74	94.85	95.08	94.96
sm(ft)+sp+tok(in+lim(fil1(lf+rf)))+fws3+ix+ug+bg	96.60	95.00	94.44	94.72
ft+sp+tok(in+lim(fil1(lf+rf)))+fws3+ix+ug+bg+dic	94.86	96.21	87.55	91.68

Table 4 experiments are important sources. We analyze those experiments to formulate decision for our features model. Within our expectation, the introduction of unigram and bigram features can greatly improve the performance by 7.8%. That the single use of bigram is of little effect may be due to the simplicity of our smoothing algorithm. However, the fuzzy word shape features boosted the classification task slightly. We attribute this phenomenon to the interference of other features.

Table 5. Description of symbol in Table 4

Description	Functions
ft = fterms	ug = unigram features
sp = suffixes/prefixes	bg = bi-gram features
in = inword features	tok(x) = tokenize x features
lf = left context features	lim(x) = limited window size (-5 and +5)
rf = right context features	fil1(x) = limited to noun and verb in extraction on dictionary features
fws1 = word shape features on the last name	
fws3 = word shape features on the first word and last two words of name.	fil2(x) = limited to noun and verb both in extraction on dictionary features and feature extraction process
ix = -in negative features from NLProt	
dic = dictionary of protein names	sm(x) = using relaxed string matching for x

By looking at the system performance, our method without using any protein names dictionary as features performs better than using protein names dictionary as features. For this phenomenon, the only reason is the SVM classifier. Because in training task all of the protein names in dictionary are available, therefore SVM classifier which has the tendency to be outfitted to training samples has tied too much to protein name dictionary features than other features. However, in the testing task, protein names dictionary covers less instances of testing samples than it would do in training task. The possibility that no entry of our protein name dictionary appears in testing samples is also relatively high. For this reason, we also extract our protein names from various resources such as SWISSPROT.

Compared to other systems that have been developed, our system achieves better performance by integrating all of the relevant features. Torii *et al* [7] used name-internal and contextual features and implemented a context-based method in their classification task. Their system got f-score 91.00% on the protein class (while our system achieves 98.23% without the dictionary). Lee *et al* achieved 88.90% in their system which is combined with positional features, suffixes, orthographical characteristics and outside context. Therefore, it is clear that our integrated strategies with the relevant features are of great importance to the classification task.

5 Conclusions

In this paper, we present various feature sources with integrated strategies which achieve high performance to classify protein names. We introduce our new fuzzy word shape features, unigram and bi-gram features combined with all advantages of rule-based, dictionary-based, and statistical-based method. We have shown that our model is robust and capable of covering disadvantages from other models using fuzzy word shape features and our statistical based features. It is reasonable because noise words are pre filtered during features extraction. With additional source features, it is capable to cover leak of tokenization rules and also provide shape for words which are not in dictionaries. In case of words in list of dictionary, unigram and bi-gram features based on statistics will strengthen the information for classifying the name. Formulating more flexible fuzzy word shape positional model will be an attractive project. Besides, formulizing good smoothing method is promising for our statistical-based unigram and bi-grams features.

Acknowledgments

We thank to the members of the GENIA project at the University of Tokyo for making the corpus available. This work was supported by Chinese Natural Science Foundation under grant No.60272019 and 60321002.

References

1. K. Fukuda, T. Tsunoda, A. Tamura and T. Takagi, "Toward Information Extraction: Identifying Protein Names from Biological Papers", Proceedings of the 3rd Pacific Symposium on Biocomputing (PSB'1998), 3:705-716, 1998.
2. M. Narayanaswamy, K.E. Ravikumar, and K. Vijay Shanker. 2003. A Biological Named Entity Recognizer. In Proc. Of PSB 2003.8
3. A Simple and Practical Dictionary-based Approach for Identification of Protein in Medline Abstracts Sergei Egorov, PhD, Anton Yuryev, PhD, and Nikolai Daraselia, PhD (2004, American Medical Informatics Association).
4. Sven Mika and Burkhard Rost, “Protein names precisely peeled off free text.”
5. GuoDong Zhou, Jie Zhang, Jian Su, Dan Shen, and ChewLim Tan. “Recognizing names in biomedical texts: a machine learning approach”. Bioinformatics Vol. 20 no. 7. 2004, pages 1178-1190.
6. Ki-Joong Lee, Young-Sook Hwang and Hae-Chang Rim, “Two-Phase Biomedical NE Recognition based on SVMs”. Proceeding of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, pp. 33-40.
7. Manabu Torii, Scahin Kamboj, K. Vijay-Shanker, “Using name-internal and contextual features to classify biological terms”.
8. S.Mukherhea, et al. “Enhancing a biomedical information extraction with dictionary mining and context disambiguation. “. IBM J. RES. & DEV. VOL 48 No. 5/6
9. L.A. Zadeh. “Fuzzy sets.” Inform. Contr., 8, 574-591 (1965)
10. Nabota C, Collier N, Tsujii J. “Automatic term identification and classification in biology text”, Proc Natural Language Pacific Rim Symposium 1999:369-75