2006 IEEE Congress on Evolutionary Computation
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Context-Based Technique for Biomedical Term Classification

Hisham Al-Mubaid <hisham@uhcl.edu>
University of Houston-Clear Lake, Houston, TX 77058 USA

***Abstract*—The existing volumes of biomedical texts available online drive the increasing need for automated techniques to analyze and extract knowledge from these information repositories. Recognizing and classifying biomedical terms in these texts is an important step for developing efficient techniques for knowledge discovery and information extraction from the literature. This paper presents a new technique for biomedical term classification in biomedical texts. The method is based on combing successful feature selection techniques (*MI, X^2*) with machine learning (SVM) for biomedical term classification. We utilize the advances in feature selection techniques in IR and use them to select the key features for term identification and classification. We evaluated the method using *Genia* 3.0 corpus with about 3,000 to more than 34,000 biomedical term instances. The technique is effective, achieving impressive accuracy, precision, and recall; and with *F-score* approaching ~90%, the method is superior or very competitive with the recently published results.**

I. INTRODUCTION

The research productions in the biomedical and bioinformatics fields are very high, and the resulting data and literature volumes are extremely huge and still growing in very high rates [1, 5, 7, 8]. For example, *MedLine* contains about 14 million abstracts [22]. The knowledge embedded in the literature is really massive which drives a great need to discover and extract more useful and significant knowledge, information, and relations to benefit the field. There is a need for effective techniques from machine learning, text mining, and natural language processing (NLP) to analyze these texts and extract significant knowledge for the advancement of the science [1, 10, 13]. In the past two decades, significant amount of research projects have been devoted to problems related to biomedical terms and entity names including named entity recognition (NER), term identification and classification, and gene/protein name disambiguation in biomedical literature [2, 3, 6]. These tasks are really important for knowledge extraction systems as resolving terms and entity names (*gene names, protein names, disease names, chemical compounds,…etc*) is a necessary step for developing efficient computational bioinformatics systems and tools [11]. For example, in the problem of bio-terms disambiguation, resolving ambiguity between genes and proteins is especially critical due to their importance in biological and medical fields. On the other hand, their disambiguation is difficult since many proteins and genes have identical names, and only few instances are explicitly disambiguated by authors in the biomedical texts. A common example of gene and protein name ambiguity (*discussed in* [5, 6, 9]) can be seen in the following two sentences:

– "*By UV cross-linking and immunoprecipitation, we show that SBP2 specifically binds selenoprotein mRNAs both in vitro and in vivo*."
– "*The SBP2 clone used in this study generates a 3173 nt transcript (2541 nt of coding sequence plus a 632 nt 3' UTR truncated at the polyadenylation site).*"

The term *SBP2* in the first sentence is a protein, whereas in the second sentence *SBP2* is used as a gene name.
In term identification and classification task we want to recognize the term and determine its boundaries in a given biological text, and then classify it into the correct class. For example, in this sentence: '*p53 protein suppresses mdm2 expression'* in an article on human signal transduction [25], the term identification step will find the term boundaries for the two entities of interest (*p53 protein* and *mdm2*). Then, the classification step determines that the first entity (*p53 protein*) is a protein, while the second entity, "*mdm2",* is classified as a gene. Notice here that the first term is already classified/disambiguated by the author while the second term (*mdm2)* is ambiguous. This paper presents a new method to identify and classify technical terms in biomedical texts. The proposed method is based on machine learning and can be viewed as a word classification task. We utilize the advanced in feature selection techniques in *Information Retrieval* (IR) and use them (*MI* and X^2) to select the key features in the contexts of the target terms. The method was evaluated extensively with a large number of experiments and achieved impressive performance, the details in the Sections II and III.

Related work

Most of the biomedical term identification and recognition techniques target certain specific entities (mostly gene and protein names) and this way term identification and term classification are integrated as one task [25]. A number of machine learning and statistical based approaches have been proposed for this task in the past two decades [17, 24, 25]. For example, Morgan et. al. (2003) [17], used HMMs based on local context and simple orthographic and case variations and reported F-measure of 75% for the recognition of *Drosophila* gene names. Moreover, in [14] Shen et. al. (2003) used POS tags and noun heads as features and achieved F-scores of 16.7% to 80% depending

0-7803-9487-9/06/$20.00/©2006 IEEE

TABLE 1
Results of the *JNLPBA-2004* competition of Bio-Entity recognition: (*recall/precision/F-score*) results of each one of the participating systems and the baseline (BL), taken from Kim et. al. (2004) [24].

	1978-1989 set	**1990-1999 set**	**2000-2001 set**	**S/1998-2001 set**	**Total**
[Zho04]	75.3/69.5/72.3	77.1/69.2/72.9	75.6/71.3/73.8	75.8/69.5/72.5	76.0/69.4/72.6
[Fin04]	66.9/70.4/68.6	73.8/69.4/71.5	72.6/69.3/70.9	71.8/67.5/69.6	71.6/68.6/70.1
[Set04]	63.6/71.4/67.3	72.2/68.7/70.4	71.3/69.6/70.5	71.3/68.8/70.1	70.3/69.3/69.8
[Son04]	60.3/66.2/63.1	71.2/65.6/68.2	69.5/65.8/67.6	68.3/64.0/66.1	67.8/64.8/66.3
[Zha04]	63.2/60.4/61.8	72.5/62.6/67.2	69.1/60.2/64.7	69.2/60.3/64.4	69.1/61.0/64.8
[Rös04]	59.2/60.3/59.8	70.3/61.8/65.8	68.4/61.5/64.8	68.3/60.4/64.1	67.4/61.0/64.0
[Par04]	62.8/55.9/59.2	70.3/61.4/65.6	65.1/60.4/62.7	65.9/59.7/62.7	66.5/59.8/63.0
[Lee04]	42.5/42.0/42.2	52.5/49.1/50.8	53.8/50.9/52.3	52.3/48.1/50.1	50.8/47.6/49.1
BL	47.1/33.9/39.4	56.8/45.5/50.5	51.7/46.3/48.8	52.6/46.0/49.1	52.6/43.6/47.7

on the class (overall F-score 66.1%; the protein class F-score was 70.8%), and reported that POS tags (obtained by a tagger trained on the biomedical domain) proved to be among the most useful features. A number of approaches employed SVM for term identification and recognition. For example, Kazama et. al. [16] trained SVMs on the *Genia* corpus for multi-class classification. They annotated the training data class label and with *B*, *I*, and *O* labels to indicate that a term is *beginning, inside*, or *outside* the term (for example, the label 'B-Gene' indicates that the word is in the beginning of a gene name). They used position-dependent features (POS, prefix, suffixes, …etc.) [16]. Moreover, Takeuchi and Collier [18] used head-noun features in combination with orthographic features the reported performance was encouraging (F-score of 74.2% for ten classes).

The *JNLPBA-2004* competition [19] included eight systems for the Bio-Entity recognition task [24]. The competition was an open challenge, and the participants were allowed to use whatever techniques and data resources they like. However, the systems were evaluated using a common evaluation methodology and a common datasets. Four types of classification models where used in these eight systems: SVMs, HMM, MEMM, and CRFs. The overall results (Table 1) showed the *recall* ranges from 50.8% to 76.0%, *precision* from 43.6% to 69.4%, and *F-score* from 47.7% to 72.6% [19, 24]. The work presented in this paper is most similar to this competition [19, 24] as we focus on biomedical term classification rather than recognition. We assume that the terms/entity names are already recognized and labeled and we attempt to classify them into their correct classes.

The rest of the paper is organized as follows. Next section explains our method. Section III describes the experiments and discusses the results. Finally, Section IV presents the conclusion and future work.

II. THE METHOD

A number of previous related methods use the words around the terms of interest as features for term identification or classification [5, 6, 9]. In our method, we also use word features to represent the biomedical terms, but the words in the context of the term are not used directly as features. Instead, we select as features only those words having high '*discriminating*' capabilities between the various classes of terms. These *word* features are used to represent each instance (example) of the terms in the training and testing. The method then uses machine learning (SVMs) to train classifiers with labeled (*training*) examples. So, some already labeled terms (*annotated with class labels*) are used as labeled training examples. The classifiers will then be used to classify unseen and unlabeled examples in the testing (*classification)* phase. One of the contributions of this work is the way we select features for learning and classification.

A. Feature Selection

Feature selection is a key issue in the efficiency of the learning and classification of such methods as the one presented here. A lot of research work has been devoted to feature selection in machine learning and data mining, particularly in *text categorization* research, see for example [26, 4, 15].

Assume that we have two sets C_1 and C_2 of labeled examples extracted from biomedical texts. Let C_1 contains examples of biomedical term instances and their contexts from one category/class (C_1), whereas C_2 includes examples with their contexts from another biomedical category (C_2). We want to classify terms from C_1 and C_2 into their correct classes. The *term*, which belongs to either C_1 or C_2, is what is to be classified in this case, and the words preceding and following the term are its *context words*. So each example in the set C_1 or C_2 can be represented as:

$$p_n \ldots p_3\ p_2\ p_1 \langle term \rangle f_1\ f_2\ f_3 \ldots f_n$$

where the words $p_1, p_2, p_3, \ldots, p_n$ and $f_1, f_2, f_3, \ldots, f_n$ are the preceding and following words (context words) surrounding the *term*, and *n* is called the *window size* (*w*). From the examples in C_1 and C_2, we extract all these *p* and *f* context words into the set *W* such that: $W = \{w_1, w_2, \ldots, w_m\}$ Now, each such context word $w_i \in W$ may occur in contexts from either C_1 or C_2 or both with different frequency distributions. We want to determine that if we see a context word w_i in an ambiguous example to what extent this occurrence of w_i suggests that this example belongs to C_1 or C_2. Thus, we select those words w_i from *W* which are highly associated with either C_1 or C_2 (the highly discriminating words) as features. We utilize and apply the feature selection techniques *mutual information* (MI) and *chi-square* (X^2) ([4, 15]) to select the highly discriminating context words from *W*. These feature selection techniques, *MI* and X^2, were used successfully for feature selection in text categorization and information retrieval [26, 4, 15, 16]. We explain, in the rest of this section, how we use *MI* and X^2 to determine which context words from *W* to be selected as features.

Let us first define the notions of *a, b, c,* and *d:* From the training examples, we calculate four numeric values *a*, *b*, *c*, and *d* for each context word $w_i \in W$ as follows:

a = number of occurrences of w_i in C_1
b = number of occurrences of w_i in C_2
c = number of examples of C_1 that do not contain w_i
d = number of examples of C_2 that do not contain w_i

Then, the *mutual information* (MI) is defined as:

$$\text{MI} = \frac{N*a}{(a+b)*(a+c)}$$

where *N* is the total number of examples in C_1 and C_2.
And Chi-Square (X^2) is computed as:

$$X^2 = \frac{N*(ad-cb)^2}{(a+c)*(b+d)*(a+b)*(c+d)}$$

again *N* is the total number of examples in C_1 and C_2.
When using the *MI* technique for feature selection, we calculate *MI* values for each $w_i \in W$, then we choose the top *v* words $w_i \in W$ with the highest *MI* values as features in this term's *feature vectors*. In our experiments we tested on *v* values of 10, 20, 30, 50, and 100. For example, if *v=10*, then each training example is represented by a vector size of 10 entries (thus, *v: vector size)* such that the first entry represents the word with the highest MI value, the second entry represents the word with the second highest MI value, and so on. Then for a given training example, the feature vector entry is set to 1 if the corresponding feature word occurs in that training example and set to 0 otherwise.

TABLE 2
Words with the top MI values

Context words w_i	MI
activate	1.92
process	1.90
sample	1.87
deliver	1.86
inhibit	1.68
went	1.56
generate	1.48
smear	1.33
diagnose	1.33
clear	1.27
…	…

Consider the following example, let $W = \{w_1, w_2, \ldots, w_m\}$ be the set of all context words. We compute *MI* for each $w_i \in W$ and sort the words w_i according to their MI values in descending order as shown in Table 2. Table 2 contains the top 10 context words with the highest MI values. These 10 words will be used to compose the feature vectors for training or testing examples of the term to be disambiguated. For example, the following feature vector:

0	1	1	0	0	0	1	0	0	0

represents an example containing the 2nd, 3rd and 7th feature words (*i.e., process, sample,* and *generate)* occurring in the context of the term in that example within certain window size. Additionally, if the window size *is 4*, then that example may look like:

— *generate* — — <*the-term*> — *sample* — *process*

That is, three of the 10 feature words are occurring within the window of size 4 of the *term*. In this case, window size is 4 and the vector size is 10. Notice also that, we do not encode positional information of the feature words in the feature vector. For example the word '*generate'*, occurred as third preceding word in the term context but it is translated to a '1' in the seventh entry of the feature vector.

B. Learning and Classification

As we have seen, we generate feature vectors from the training examples using the top words selected using *MI* or X^2. Then, we use a well-established learning technique *Support Vector Machines* (SVM) [20, 21] to train classifiers with the training vectors. SVM is an inductive learning technique for two-class classification. Significant theoretical and empirical justifications exist in the literature to support SVM [20, 21]. In our method, for each class of biomedical terms, we construct one feature vector for each training example. Then we take two classes at a time and apply SVM to train on these two classes and a classifier (model) is produced. The classifier will then be used in the testing/classification phase to classify testing instances. We use SVM-*light* (svmlight.joachims.org*)* with the default parameters except that we adjust the cost factor (*j* parameter) by which training errors on positive examples outweigh errors on negative examples (*default j=1*).

TABLE 3
The 36 terminal classes of *Genia 3.0.*

Class names		
amino_acid_monomer	DNA_domain_or_region	atom
peptide	DNA_family_or_group	carbohydrate
protein_N/A	DNA_molecule	lipid
protein_complex	DNA_substructure	virus
protein_domain_or_region	RNA_N/A	mono_cell
protein_family_or_group	RNA_domain_or_region	multi_cell
protein_molecule	RNA_family_or_group	body_part
protein_substructure	RNA_molecule	tissue
protein_subunit	RNA_substructure	cell_type
nucleotide	other_organic_compound	cell_component
polynucleotide	organic	cell_line
DNA_N/A	inorganic	other_name

TABLE 4
The set of class pairs selected for our evaluations and includes 30 class pairs.

No.	Class 1 [C_1]	C_1 instances	Class 2 [C_2]	C_2 instances	$C_1 + C_2$ instances	Training 80%	Testing 20%
1	amino_acid_monomer	780	protein_domain_or_region	990	1770	1418	352
2	peptide	518	RNA_molecule	557	1075	861	214
3	DNA_substructure	106	protein_substructure	127	233	187	46
4	carbohydrate	97	protein_substructure	127	224	180	44
5	nucleotide	236	RNA_family_or_group	332	568	455	113
6	mono_cell	222	nucleotide	236	458	367	91
7	polynucleotide	258	RNA_family_or_group	332	590	473	117
8	mono_cell	222	polynucleotide	258	480	385	95
9	other_artificial_source	207	polynucleotide	258	465	373	92
10	inorganic	255	RNA_family_or_group	332	587	471	116
11	atom	340	RNA_family_or_group	332	672	539	133
12	lipid	2357	virus	2117	4474	3580	894
13	lipid	2357	multi_cell	1745	4102	3283	819
14	multi_cell	1745	virus	2117	3862	3091	771
15	cell_component	662	tissue	678	1340	1073	267
16	protein_family_or_group	8247	virus	2117	10364	8292	2072
17	protein_family_or_group	8247	tissue	678	8925	7141	1784
18	protein_family_or_group	8247	lipid	2357	10604	8484	2120
19	protein_family_or_group	8247	atom	340	8587	6871	1716
20	protein_molecule	21511	virus	2117	23628	18903	4725
21	protein_molecule	21511	tissue	678	22189	17752	4437
22	DNA_domain_or_region	7810	virus	2117	9927	7943	1984
23	DNA_domain_or_region	7810	tissue	678	8488	6792	1696
24	DNA_domain_or_region	7810	lipid	2357	10167	8135	2032
25	other_organic_compound	4081	peptide	518	4599	3680	919
26	cell_type	7021	nucleotide	236	7257	5806	1451
27	cell_type	7021	lipid	2357	9378	7503	1875
28	cell_type	7021	peptide	518	7539	6032	1507
29	cell_line	3846	lipid	2357	6203	4963	1240
30	cell_line	3846	peptide"	518	4364	3492	872
Total		**142,638**		**30481**	**173119**	**138525**	**34594**

III. EXPERIMENTS AND RESULTS

We implemented and evaluated the proposed method with a large number of experiments using the data from the *Genia 3.0* corpus. In this section, we describe the datasets, the experimental design, and then we discuss the results.

A. Dataset

The data for training and testing are taken from the *Genia* corpus version 3.0 [23]. This corpus is used as benchmark in most of the biomedical term/entity name related problems [18, 19, 24]. The *Genia* corpus was developed at *University of Tokyo* and constructed from *Medline* [22] by querying using terms *'human'*, *'blood cells'* and '*transcription factors'*. From this search process, 2,000 abstracts were selected for the corpus. The identified terms, in these selected documents, were hand annotated with 36 classes/types, these classes are shown in Table 3. The corpus contains a total of 75,108 term occurrences.

B. Experimental Design

We selected for testing 30 pairs of classes such that we want each pair to contain two classes having equal or close number on instances. These classes are shown in Table 4. The total number of term instances in these 30 class pairs is 173,119 instances of which 34,594 instances were used for the testing (Table 4). We used 5-fold cross validation, such that, we divided the data into 5 equal folds and repeated each experiment five times. Each time we leave one fold (20% of the data) out for testing and use the remaining 4 folds (80%) for training. In the text preprocessing step, the training and testing texts were preprocessed as follows: (1) We changed all the letters into lower case (2) Word stemming: all words converted to their stems using *Porter's* stemming algorithm [12]. (3) *Stopword* removal: we removed all the function words (*stopwords)* like *'the', 'of', 'in', 'for', 'on', ...etc.* For performance metrics we use *accuracy*, *precision, Recall,* and *F1-score.*

C. Results and Discussion

Firstly we conducted a variety of experiments using feature selection techniques *MI* and X^2 to compare their performance. In these experiments we changed window size *w,* with varying vector size *v* as well. We initially experimented with a smaller dataset of 15 class pairs, shown in Table 5. The total number of instances in these 15 class pairs is 20,900 instances and number of testing instances is 4,164; Table 5 contains the numbers of instances in each class. Before we delve into the details of the results we present little more details about the experimental procedure. Consider the first class pair in Table 5 [*amino_acid_monomer*, *protein_ domain_or_region*]. The first class (*amino_acid_monomer)* includes 780 annotated terms from this class, whereas the second class (*protein_domain_or_region)* contains 990 annotated terms, and the total is 1,770 terms (Table 5). Of these 1,770 instances, 80% (1,418 instances; 4 folds) were used for training and the remaining 20% (354 instances; 1 fold) are used for testing. This step is repeated 5 times by changes the training/testing folds. We record accuracy, precision, and recall for each round, and then we take the average accuracy, precision, and recall of the five rounds. This procedure is done for each one of the 15 pairs. Finally we take the *microaverage* of accuracy, precision, and recall for all of the 15 testing pairs. Table 6 shows the results of the first set of experiments (using the data in Table 5) in which we changed the window size *w*, vector size *v* with the two feature selection techniques *MI* and X^2. In this table, we notice that using windows size *w=3*, and vector of size *v=30* with X^2 for feature selection produces the highest *accuracy* (75.17%) and F_1 (75.55%) results, while the best *precision* (81.48%) was produced with *MI* when *w=3* and *v=30*. In the second set of experiments on the data of Table 5 (the smaller dataset), we examined the performance with three text preprocessing steps: word *stemming, stopword* removal, and converting all letters into lowercase; the results are in Table 7. Table 7 indicates that word stemming and stopword removal did improve the performance but only slightly. Table 8 contains the results when the preprocessing steps are done in combinations, window size *w=10*, X^2 for feature selection, and two different vector sizes *v=20* and *v=30*. These results in Table 8, indicate that using all the three preprocessing steps with *v=30* gives the best results: 62.23% accuracy and 64.07% F_1 score. Next, we conducted the main experiments using the main and larger dataset (Table 4) that includes 30 class pairs and more than 173,000 term instances. The results are in Table_9 when no preprocessing is done. We notice that these results are significantly higher than the results on the first set of class pairs. We examined only X^2 (and not *MI*) because X^2 was giving better results than *MI* in most cases. This performance with accuracy ranging between 82.87% to 85.07% and F_1 from 88.23% to 90.24% (Table 9) is higher that the results of the similar work as reported earlier in this paper (Section I & Table 1). Next, we examined the performance after the preprocessing steps; we firstly applied the preprocessing steps one at a time and the results are in Table 10. Then, combinations of preprocessing steps were applied in the third set of experiments and the results are in Table 11.

These results clearly demonstrate that our technique for biomedical term classification produces impressive performance results proven by a large number and variety of experiments. For example, each line in tables 9, 10, and 11 represents the average *accuracy/precision/recall/F1* of five experiments (*5-fold CV)*, each experiment is done on 34,594 testing instances. Moreover, we notice that the strength of the proposed method lies mostly on the feature selection technique and the learning/classification process. We have seen that the preprocessing steps (Table 10 and Table 11) did not improve the performance results of Table_9 significantly, because the strength comes from the way the features selected and the way training/testing is done rather than from preprocessing steps. Moreover, the discussion about the related work (Section I) provides

some results from the published similar work [14][16][18][19][24] which indicate that our method outperform most of these methods. For example, Tables 9, 10, and 11 show that our method is capable of giving F1 scores close to or exceeding 90% whereas the best F1 in Table 1 is 73.8% [24]. Hence, we can conclude that the strength of the method (and the contribution of this work) comes from the unique combination of the successful feature selection techniques (*MI, X^2*) with one of the best performers in machine learning (*SVM*) for this task. To the best of our knowledge, these results are among the best published performance results for this particular task.

TABLE 5
The smaller dataset of class pairs, selected for the preliminary testing, and contains 15 class pairs.

No.	Class 1 [C_1]	C_1 instances	Class 2 [C_2]	C_2 instances	$C_1 + C_2$ instances	Training 80%	Testing 20%
1	amino acid monomer	780	protein domain or region	990	1770	1418	352
2	peptide	518	RNA molecule	557	1075	861	214
3	DNA substructure	106	protein substructure	127	233	187	46
4	carbohydrate	97	protein substructure	127	224	180	44
5	nucleotide	236	RNA family or group	332	568	455	113
6	mono cell	222	nucleotide	236	458	367	91
7	polynucleotide	258	RNA family or group	332	590	473	117
8	mono cell	222	polynucleotide	258	480	385	95
9	other artificial source	207	polynucleotide	258	465	373	92
10	inorganic	255	RNA family or group	332	587	471	116
11	atom	340	RNA family or group	332	672	539	133
12	lipid	2357	virus	2117	4474	3580	894
13	lipid	2357	multi cell	1745	4102	3283	819
14	multi cell	1745	virus	2117	3862	3091	771
15	cell component	662	tissue	678	1340	1073	267
	Total	10362		**10538**	**20900**	**16736**	**4164**

TABLE 6
Results of the first set of experiments using different feature selection (*f.s.)* techniques, window size (*w)*, and vector sizes (*v*).

Experiment			A	P	R	F1
f.s.	*w*	*v*				
MI	3	10	54.63	26.39	41.37	32.22
	3	20	55.86	56.84	44.08	49.65
	3	30	57.23	**81.48**	47.12	59.71
X^2	3	10	69.07	70.79	69.83	70.31
	3	20	71.93	73.71	70.36	72.00
	3	30	**75.17**	76.09	**75.01**	**75.55**
MI	5	10	54.59	25.34	44.38	32.26
	5	20	54.78	41.5	44.82	43.10
	5	30	55.48	66.62	43.23	52.43
X^2	5	10	58.91	64.27	51.13	56.95
MI	10	10	54.66	28.12	44.55	34.48
	10	20	55.02	55.55	45.37	49.95
	10	30	55.09	60.54	45.52	51.97
X^2	10	10	57.66	66.08	46.63	54.68
	10	20	52.57	67.26	9.15	16.11
	10	30	53.96	72.74	11.72	20.19

f.s.: feature selection, w: window size, v: vector size, A: accuracy, P: precision, R: recall, and F1: F_1-score.

TABLE 7
Results of the second set of experiments using X^2 for feature selection, and the three preprocessing steps, one at a time.

Experiment			A	P	R	F1
Preprocessing	***w***	***v***				
Stemming (Porter's)	10	20	57.42	58.54	54.03	56.19
	10	30	58.12	61.66	48.85	54.51
Stopword removed	10	20	54.08	89.45	8.75	15.94
	10	30	54.25	81.98	10.30	18.30
Convert to lowercase	10	20	52.57	67.26	9.15	16.11
	10	30	53.96	72.74	11.72	20.19

TABLE 8
Results of using combinations of preprocessing steps. In these experiments *w=10,* and *f.s.* is X^2.

Experiment		A	P	R	F1
Preprocessing	***v***				
lowercase + stopword removed	20	60.09	68.12	51.88	51.88
	30	60.09	68.12	51.88	51.88
lowercase + word stemming	20	56.80	56.28	55.93	55.93
	30	57.69	61.10	48.92	48.92
stopword removed + word stemming	20	57.78	64.25	47.73	47.73
	30	59.10	66.77	50.52	50.52
lower case + stopword removed + word stemming	20	57.42	63.27	47.04	47.04
	30	62.23	65.86	64.07	64.07
Average		**58.90**	**64.22**	**52.25**	**52.25**

TABLE 9
Results of the main/larger dataset, using *w=5,10; v = 20, 30; and feature selection (f.s) is* X^2 *[no preprocessing steps]*

Experiment			A	P	R	F1
f.s.	*w*	*v*				
X^2	5	20	84.53	85.20	94.25	89.50
	5	30	85.07	85.67	95.33	90.24
	10	20	82.87	84.28	92.56	88.23
	10	30	84.30	85.54	93.08	89.15

TABLE 10
Results using the main/larger dataset, *w=5; v=20, 30; and feature selection (f.s) is* X^2 with preprocessing steps (one at a time).

Experiment			A	P	R	F1
Preprocessing	***w***	***v***				
Word stemming	5	20	84.14	84.17	95.86	89.63
	5	30	84.38	84.71	95.16	89.63
Stopword remvd	5	20	83.85	85.22	92.42	88.67
	5	30	84.20	85.74	92.42	88.96
lowercase	5	20	84.07	87.37	90.37	88.85
	5	30	84.56	74.94	78.25	76.56

TABLE 11
Results of the set of experiments with the main dataset, using combinations of preprocessing steps. In these experiments window size *w=5,* vector size *v=20, 30,* and *f.s.* is X^2.

Experiment		A	P	R	F1
Preprocessing	***v***				
lowercase + stopword removed	20	84.34	86.27	93.18	89.59
	30	84.65	86.39	93.58	89.84
lowercase + word stemming	20	84.17	84.19	95.83	89.63
	30	84.33	84.66	95.16	89.60
stopword removed + word stemming	20	84.21	84.68	95.36	89.71
	30	84.36	84.89	95.20	89.75
lower case + stopword removed + word stemming	20	84.19	84.82	95.09	89.66
	30	84.38	84.91	95.12	89.72
Average		**84.33**	**85.10**	**94.81**	**89.69**

IV. CONCLUSION AND FUTURE WORK

We have presented an approach for biomedical term classification. The experimental results showed that the method is effective in classifying biomedical terms using few surrounding context words as features. We borrowed from the IR and TC domains two successful feature selection techniques (viz. *mutual information* and *Chi-square*), and proved with a variety of experiments the effectiveness of the approach. The strength of the method comes from the unique combination of successful feature selection techniques with one of the top machine learners (SVM) into the biomedical term classification problem. The experimental results clearly demonstrated that the approach is very impressive. In the future endeavor of this research, we plan to explore more feature selection techniques into this task like *information gain (IG).* We also plan to evaluate the method with more datasets, and investigate the possibility of using the concept of *information content (IC)* in the training and classification process.

ACKNOWLEDGEMENT

This work was supported by the Institute for Space Systems Operations (ISSO), February 2005.

REFERENCE

[1] L. A. Adamic, D. Wilkinson, B. A. Huberman, E. Adar. A literature based method for identifying gene-disease connections, *IEEE Computer Society Bioinformatics Conference,* 2002.

[2] M. Andrade, A. Valencia, Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families, *Bioinformatics,* Vol.14, 1998.

[3] H. Al-Mubaid and P. Chen. Biomedical Term Disambiguation: An Application to Gene-Protein Name Disambiguation. *In IEEE proceedings of ITNG-06*, 2006.

[4] L. Galavotti, F. Sebastiani, M. Simi. Experiments on the use of feature selection and negative evidence in automated text categorization. The 4th European Conf. on Research and Advanced Technology for Digital Libraries *ECDL-00*, 2000.

[5] F. Ginter, J. Boberg, J. Jarvinen, T. Salakoski, New Techniques for Disambiguation in Natural Language and Their Application to Biological Text. *JMLR,* 5, 2004.

[6] V. Hatzivassiloglou, P. A. Dubou´e, A. Rzhetsky, Disambiguating proteins, genes, and RNA in text: A machine learning approach, *Bioinformatics*, vol. 17, 2001.

[7] L. Hirschman, J.C. Park, J. Tsujii, L. Wong, C.H. Wu. Accomplishments and challenges in literature data mining for biology, *Bioinformatics,* Vol. 18, 2002.

[8] Medline: accessed using Entrez PubMed Interface: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi

[9] T. Pahikkala, F. Ginter, J. Boberg, J. Jarvinen, T. Salakoski, Contextual weighting for Support Vector Machines in literature mining: an application to gene versus protein name disambiguation *BMC Bioinformatics* 2005.

[10] P. Srinivasan, Text mining: generating hypotheses from MEDLINE, *Journal of the American Society for Information Science and Technology*, v.55 n.5, p.396-413, March 2004

[11] N. Uramoto, H. Matsuzawa, T. Nagano, A. Murami and H. Takeuchi. A text-mining system for knowledge discovery from biomedical documents, IBM Systems Journal, Vol. 43, 2004.

[12] M.F. Porter. An algorithm for suffix stripping. Program, 14:130–137, 1980.

[13] J. D. Wren, R. Bekeredjian, J.A. Stewart, R.V. Shohet, H.R. Garner, Knowledge discovery by automated identification and ranking of implicit relationships, *Bioinformatics*, Vol.20, No.3, 2004.

[14] Shen, D., J. Zhang, G. Zhou, J. Su, and C. Tan. *Effective Adaptation of Hidden Markov Modelbased Named Entity Recognizer for Biomedical Domain*. In: Proceedings of *NLP in Biomedicine, ACL 2003*. 2003. Sapporo, Japan. p. 49-56.

[15] Y. Yang, J.P. Pedersen . A comparative study on feature selection in text categorization. In Jr. D. H. Fisher, editor, *The 4th International Conf on Machine Learning*, pp. 412-420, 1997

[16] Kazama, J., T. Makino, Y. Ohta, and J. Tsujii. *Tuning support vector machines for biomedical named entity recognition*. In: Proceedings of *Workshop on NLP in the Biomedical Domain, ACL 2002*. 2002. Philadelphia. p. 1-8.

[17] Morgan, A., A. Yeh, L. Hirschman, and M. Colosimo. *Gene Name Extraction Using FlyBase resources*. In: Proceedings of *NLP in Biomedicine, ACL 2003*. 2003. Sapporo, Japan. p. 1-8.

[18] Takeuchi, K. and N. Collier. *Bio-medical Entity Extraction using Support Vector Machines*. In: Proceedings of *NLP in Biomedicine, ACL 2003*. 2003. Sapporo, Japan. p. 57-64.1 8

[19] JNLPBA-04 workshop: http://www.genisis.ch/~natlang/JNLPBA04/ • *shared task homepage:* http://research.nii.ac.jp/~collier/workshops/JNLPBA04st.htm

[20] V. Vapnik. The nature of statistical learning theory. Springer, New York, USA, 1995.

[21] B. E. Boser, I. Guyon, V. Vapnik. A training algorithm for optimal margin classifiers. In COLT, pages 144–152, 1992.

[22] Medline: accessed using Entrez PubMed Interface: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi

[23] Kim JD, Ohta T, Tateisi Y, Tsujii J. GENIA corpus—a semantically annotated corpus for bio-textmining. Bioinformatics: 19 (Suppl. 1) 2003.

[24] Kim, J-D, Tomoko, O., Yoshimasa, T., Tateisi, Y. and Collier, N. (2004). Introduction to the Bio-Entity Recognition Task at JNLPBA. In the Proc. of the Intl Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-04).

[25] Miachel Krauthammer and Goran Nenadic, "Term Identification in the Biomedical Literature". Journal of Biomedical Informatics, 2004.

[26] G. Forman. An Extensive Empirical study of feature selection metrics for text classification. JMLR, 2003.