Naïve Bayes and SVM classifiers for classifying Databank Accession Number sentences from online biomedical articles

Jongwoo Kim*, Daniel X. Le, and George R. Thoma
National Library of Medicine, 8600 Rockville Pike, Bethesda, MD 20894, USA

ABSTRACT

This paper describes two classifiers, Naïve Bayes and Support Vector Machine (SVM), to classify sentences containing Databank Accession Numbers, a key piece of bibliographic information, from online biomedical articles. The correct identification of these sentences is necessary for the subsequent extraction of these numbers. The classifiers use words that occur most frequently in sentences as features for the classification. Twelve sets of word features are collected to train and test the classifiers. Each set has a different number of word features ranging from 100 to 1,200. The performance of each classifier is evaluated using four measures: Precision, Recall, F-Measure, and Accuracy. The Naïve Bayes classifier shows performance above 93.91% at 200 word features for all four measures. The SVM shows 98.80% Precision at 200 word features, 94.90% Recall at 500 and 700, 96.46% F-Measure at 200, and 99.14% Accuracy at 200 and 400. To improve classification performance, we propose two merging operators, Max and Harmonic Mean, to combine results of the two classifiers. The final results show a measureable improvement in Recall, F-Measure, and Accuracy rates.

Keywords: Naïve Bayes, Support Vector Machine (SVM), databank, labeling, text classification, bibliographic information.

1. INTRODUCTION

The U.S. National Library of Medicine (NLM) maintains MEDLINE®, a heavily used bibliographic database of 17 million citations to the biomedical journal literature. Each citation consists of bibliographic information such as article title, author names, affiliations, grant numbers, grant support types, databank accession numbers, etc. While NLM receives most such citations in XML format directly from journal publishers, key bibliographic information is often missing, requiring manual entry. Databank Accession Number (DAN) [1] is typically one such missing item.

Databanks are databases/registries of genetic sequences, clinical trials, gene expression data, genomic DNA/protein sequences, small molecules, etc. There are several databanks such as GenBank, NCT, PDB, etc. and DAN is the registration number of a sequence (entry) in any of these databanks. DAN usually appears in a sentence together with other information such as databank names and/or words such as “deposit”, “submit”, etc. For the purpose of this article, we call this sentence a “DAN sentence”. An example of DAN sentences is “The confirmed nucleotide sequence of mouse preproET-1 cDNA was deposited into the GenBank database (accession no AB081657).” In this sentence, GenBank is the databank name and “AB081657” is a DAN. Indentifying a DAN sentence is a precursor to extracting the DAN by subsequent pattern matching.

To find DAN sentences manually, professional indexers have to carefully search an entire article since DAN sentences, although usually located in the first or last page of an article, can occur anywhere. The work is labor-intensive and often error-prone; hence our interest in an automated approach.

The automatic detection of DAN sentences may be formulated as a text classification/categorization problem and several algorithms are used for this purpose. In earlier work we developed a rule-based algorithm [2] to classify DAN sentences. The rules in the algorithm are based on three types of clue words (Databank names, words such as “deposit”, “submit”, and words such as “accession”, etc.) and DAN formats. Although the algorithm works well for DAN sentences with the clue words, it frequently generates under- or over-classification errors when these sentences do not contain the clue words. We therefore focus on machine learning approaches in our current work and choose two common algorithms, Naïve Bayes and SVM classifiers, to solve this problem.

*jongkim@mail.nih.gov; phone 1 301 435-3227; fax 1 301 402-0341:

Naïve Bayes classifier [3] is a widely used technique for text classification. Due to its simplicity, efficiency, and speed, it is widely used in classifying Web documents [4], spam emails [5], and other types of documents such as newsgroups, newswire articles, etc. [6]. SVM [7] is also commonly used to categorize newswire documents and Medical Subject Headings (Mesh) [8], Reuters-21578 collection (in which 12,902 stories fall into 118 categories) [9], and Web documents [10]. We therefore adapt both Naïve Bayes and SVM classifiers to identify DAN sentences, and then combine them using two merging operators to improve performance.

The paper is organized as follows. The definition of a DAN sentence is given in Section 2. The details of our method using the Naïve Bayes and SVM classifiers are presented in Section 3. Performance evaluation measures are shown in Section 4. We report experimental results in Section 5, and conclusions in Section 6.

2. DATABANK ACCESSION NUMBER (DAN) SENTENCE

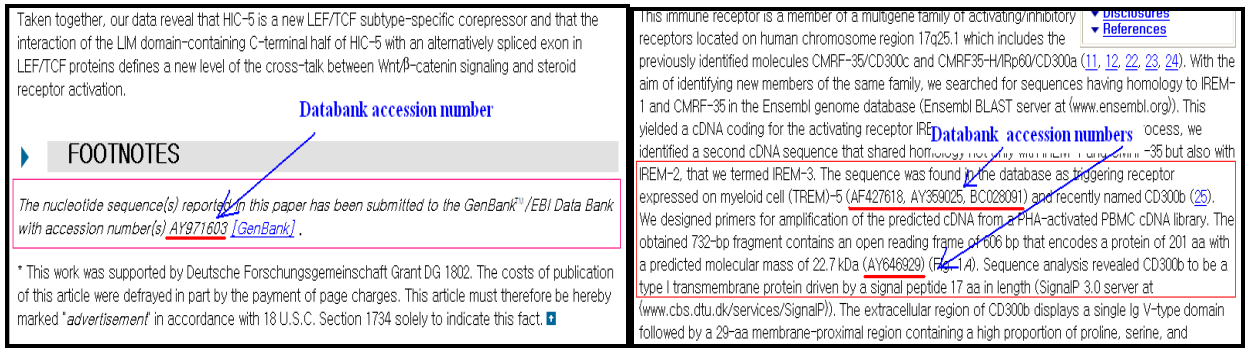
Each of the several databanks has its own distinct DAN format. Table 1 shows a list of databank names and their corresponding DAN formats. The first databank called "GenBank" has three different formats illustrated in the examples as "A12345", "AB123456", and "ABC12345". Other databanks also have their own DAN formats except for the ones in the last row: SwissProt, PIR, GDB, CSD, HGML and PREFSEQDB databanks follow free formats. We can therefore recognize a DAN sentence based on databank names and numbers that follow known formats.

Table 1. Databank names and Databank accession number formats.

Databank name	**Databank accession number format**	**Example**
GenBank [11]	[one-letter character]+[five-digit number], [two-letter character]+[six-digit number], [three-letter character]+[five-digit number]	A12345, AB123456 ABC12345
NCT (Clinical Trials) [12]	NCT+[eight-digit number]	NCT 12345678
GEO (Gene Expression Omnibus) [13]	{GEO, GDS, GSE, GPL, or GSM }+[any digit number]	GDS01, GSE1234567
ISRCTN [14]	ISRCTN+[eight-digit number]	ISRCTN 12345678
RefSeq (Reference Sequence) [15]	{AC, AP, NC, NG, NM, NP, NR, NT, NW, NZ, XM, XP, XR, YP, or ZP } + "_" + [six or nine-digit number]	AC_123456, AC_123456789
OMIM (Online Mendelian Inheritance in Man) [16]	OMIM+{ space, *,#,+,%, or ^} + {1,2,3,4,5, or 6} + [five-digit number]	OMIM ^123456,
PDB (Protein Data Bank) [17]	[one-digit number] + [three-digit Alphabet character or Arabic number]	1FA7
PubChem [18]	{PubChem, PubChem-Substance, PubChem-Compound, or PubChem-BioAssay} + [any digit number]	PubChem/12345, PubChem-Substance/ 123456
SwissProt, PIR, GDB, CSD, HGML, PREFSEQDB [1]	[Free Formats]	Free Formats

Figure 1 shows typical articles that contain several DAN sentences. Figure 1(a) shows a DAN sentence having a databank name "GenBank", a DAN "AY971603", and the word "submitted". It is clear that "AY971603" is a DAN because of the words "GenBank" and "submitted" in the sentence. This sentence is located at the end of the article. Figure 1(b) shows two DAN sentences with four DANs. But there are no databank names corresponding to the DANs in these sentences. In addition, these DAN sentences are located in the middle of the article which is not a usual place to search for DANs. However, there is a word "sequence" in the first sentence for three DANs (AF427618, AY359025, and BC028091) and the second sentence also has words such as "protein" and "molecular" for one DAN (AY646929).

Table 2 shows three Non-DAN sentences that have numbers that appear to follow DAN formats. The first sentence contains NIH Grants "AI065898" and "RR015563" that mimic a DAN format in GenBank. The second and third sentences also include year "2001" and page numbers "2488" and "2492" that appear to be a DAN in the PDB databank. However, there are no words suggesting DANs in these sentences. Clearly, words suggesting DANs are important to identify a DAN sentence. We use words that occur frequently in DAN and Non-DAN sentences as word features for classifying DAN sentences.



(a) (b)

Figure 1. Examples of Databank accession numbers (a) AY971603, (b) AF427618, AY359025, BC028091, and AY646929.

Table 2. Examples of Non-DAN sentences.

Sentences	**Numbers mimicking DAN formats**
1. This research was supported by NIH Grants **AI065898**, and **RR015563**.	AI065898, RR015563
2. Stossel TP, Condeelis J, Cooley L, et al. Filamins as integrators of cell mechanics and signalling. Nat Rev Mol Cell Biol. **2001**;2: 138	2001
3. C.E. Outten and T.V. O'Halloran, Femtomolar sensitivity of metalloregulatory proteins controlling zinc homeostasis, *Science* 292 (**2001**), pp. **2488**–**2492**.	2001, 2488, 2492

3. OUR APPROACH

3.1 Naïve Bayes classifier

Assume that we have a binary feature vector from a sentence $\boldsymbol{x}=(x_1, x_2, x_3, \ldots, x_m)$ where m is the dimension of the vector and $x_i = 0$ or 1 means absence or presence of the ith feature (feature refers to word, in our case) in the vector. Assume there are two classes C_r and C_n: relevant and non-relevant classes. In this paper, DAN sentences belong to C_r and Non-DAN sentences belong to C_n. The decision function can be written as

$$P(\boldsymbol{x}|C_r)\, P\,(C_r) > P\,(\boldsymbol{x}|C_n)\, P(C_n), \tag{1}$$

where $P(C_j)$ is the prior probability of C_j.

Assume that the features x_i in feature vector $\boldsymbol{x}=(x_1, x_2, \ldots, x_m)$ are stochastically independent. Let us define p_i as the probability of occurrence of a word (ith word) suitable as a feature in a sentence that is in a relevant class, and q_i as the probability of the word (ith word) in a non-relevant sentence. Then, $P(\boldsymbol{x}|C_j)$ can be rewritten as

$$P(\boldsymbol{x} \mid C_r) = \prod_{i=1}^{m} p_i^{x_i} (1 - p_i)^{1-x_i} \tag{2}$$

$$P(\boldsymbol{x} \mid C_n) = \prod_{i=1}^{m} q_i^{x_i} (1 - q_i)^{1-x_i} \tag{3}$$

where $p_i = P(x_i=1|C_r)$ and $q_i = P(x_i=1|C_n)$.

When we insert Equations (2) and (3) into Equation (1), take logs, and move the right term to the left, we have the following linear decision function $G(\boldsymbol{x})$:

$$G(\boldsymbol{x}) = \sum_{i=1}^{m} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} x_i + \sum_{i=1}^{m} \log \frac{(1-p_i)}{(1-q_i)} + \log \frac{P(C_r)}{P(C_n)} \tag{4}$$

When $G(\boldsymbol{x})$ is positive, $\boldsymbol{x}$ belongs to C_r. If not, $\boldsymbol{x}$ belongs to C_n. We use this equation to classify the DAN sentence. To normalize $G(\boldsymbol{x})$ output to a value from 0.0 to 1.0, we use the following equation $G_{NB}(\boldsymbol{x})$:

$$G_{NB}(\boldsymbol{x}) = \frac{1}{1+e^{-G(x)}}$$, where $G(\boldsymbol{x})$ is the expression in (4). (5)

3.2 SVM classifier

We use the LIBSVM [19, 20] library and use radial basis function (RBF) as the kernel function. In the case of parameters (C, γ), the library automatically sets its own parameters after its optimization process.
Given training vectors $\boldsymbol{x}_i \in R^n$, $i = 1, 2,\ldots,l$, in two classes $y_i \in \{1,-1\}$ (1 means relevant class and -1 means non-relevant class), C-support vector classification (C-SVC) tries to solve the following problem.

$$\min_{\boldsymbol{W},b,\xi} \frac{1}{2}\boldsymbol{W}^T\boldsymbol{W} + C\sum_{i=1}^{l}\xi_i \quad (6)$$

subject to $y_i(\boldsymbol{W}^T\phi(\boldsymbol{x}_i) + b) \geq 1-\xi_i$, $\xi_i \geq 0$, $i = 1, 2,\ldots, l$.

When $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$ is the kernel and $\phi(\boldsymbol{x}_i)$ is a function mapping $\boldsymbol{x}_i$ into a higher dimensional space, the decision function is

$$\operatorname{sgn}\left(\sum_{i=1}^{l} y_i\alpha_i K(\boldsymbol{x}_{i,}\boldsymbol{x}) + b\right)$$, where $0 \leq \alpha_i \leq l$. (7)

We use the following sigmoid function to convert the results (7) into a value from 0.0 to 1.0.

$$G_{SVM}(\boldsymbol{x}) = \frac{1}{1+e^{-t}}$$, where $t = \sum_{i=1}^{l} y_i\alpha_i K(\boldsymbol{x}_{i,}\boldsymbol{x}) + b$ (8)

3.3 Feature extraction

We use the following equation to select and extract word features more related to the relevant class [21]. In this equation, p_i and q_i are the same variables used in Section 3.1.

$$M(x_i) = |\log\frac{p_i(1-q_i)}{q_i(1-p_i)}| \quad \geq \quad t \quad (9)$$

When a feature candidate x_i satisfies the above criterion (greater than or equal to the threshold t), we choose x_i as one of the features in $\boldsymbol{x}=(x_1, x_2, x_3,\ldots, x_m)$. We collect several feature sets using different values of t in our experiment. In this paper, x_i stands for a word (a frequently occurring one) selected from sentences with and without DANs.

3.4 Merging operators for the Naïve Bayes and SVM classifiers

We use two operators to combine the results of these classifiers (Equations (5) and (8)) to compensate for errors in each classifier, and to improve the classification performance:

$$G_{Max}(\boldsymbol{x}) = Max\{G_{SVM}(\boldsymbol{x}), G_{NB}(\boldsymbol{x})\} \quad (10)$$

$$G_{Harmonic}(\boldsymbol{x}) = 2.0 \times G_{SVM}(\boldsymbol{x}) \times G_{NB}(\boldsymbol{x}) / (G_{SVM}(\boldsymbol{x}) + G_{NB}(\boldsymbol{x})) \quad (11)$$

Equation (10) shows that the $G_{Max}(\boldsymbol{x})$ operator chooses a maximum value among the results of the Naïve Bayes ($G_{NB}(\boldsymbol{x})$) and SVM ($G_{SVM}(\boldsymbol{x})$) classifiers for an input sentence $\boldsymbol{x}$. In Equation (11), the $G_{Harmonic}(\boldsymbol{x})$ operator estimates the *Harmonic Mean* of the results of these two classifiers.

4. PERFORMANCE EVALUATION MEASURES

We use four measures, Precision, Recall, F-Measure, and Accuracy, to evaluate the performance of the classifiers and the merging operators. The measures are expressed as follows:

$$Precision = TP/(TP+FP),$$

$$Recall = TP/(TP+FN),$$

$$F\text{-}Measure = 2\times Precision\times Recall/(Precision+Recall),$$

$$Accuracy = (TP+TN)/(TP+TN+FP+FN),`$$

where *TP*, *TN*, *FP* and *FN* stand for the numbers of "true-positives", "true-negatives", "false-positives", and "false-negatives", respectively.

5. EXPERIMENTAL RESULTS

The experiment consists of the following steps. First, we collect twelve sets of word features using Equation (9). Second, we compare the performance of the two classifiers, Naïve Bayes and SVM, for each set of word features. Third, we use merging operators to combine the results of these two classifiers to improve performance.

We collect 21,287 sentences from biomedical articles published in 2006, to train and test the classifiers. 2,632 of these sentences are DAN sentences (relevant class) and 18,655 sentences are Non-DAN sentences (non-relevant class). From the 2,632 DAN sentences, we randomly sample 1,316 sentences for training and reserve the remaining 1,316 sentences for testing. We use the same sampling method to select sentences (9,327 and 9,328) for training and testing from the 18,655 Non-DAN sentences.

To obtain word features for the classifiers, we collect 2,094 of the *most frequently occurring* words in these sentences as "general" features, using the criterion expressed in Equation (9), and use three "special" features as shown in Table 3. From these features, we estimate $M(x_i)$ of all word features using Equation (9), sort them in descending order, and collect twelve feature sets ranging from 100 to 1,200 words by decreasing the threshold *t*. In Table 3, the definition of p_i and q_i are the same as in Section 3.1. For example, in the case of the general word feature "accession", p_i=0.65, i.e., it is found in 65% of DAN sentences (relevant class) and 0% of Non-DAN sentences (non-relevant class).

Table 4 shows the performance of the Naïve Bayes and SVM classifiers for each set of word features. We use twelve sets of word features for experiments as shown in the first column of the table. In the case of the Naïve Bayes classifier, the highest Precision (second column) is 97.70% at 200 word features, Recall (third column) 93.91% at 200, F-Measure (fourth column) 95.77% at 200, and Accuracy (fifth column) 98.97% at 200. In the case of the SVM classifier, Precision (sixth column) shows the best performance 98.80 at 400 word features, Recall (seventh column) 94.90% at 500 and 700, F-Measure (eighth column) 96.46% at 200, and Accuracy (ninth column) 99.14% at 200 and 400 word features. The Naïve Bayes classifier shows the best performance at 200 word features in all four measures while the SVM classifier does not. The SVM classifier shows the best performance at 200 word features for F-Measure and Accuracy. When we compare the performance of the two classifiers, the SVM classifier performs a little better than the Naïve Bayes classifier in all four measures using several sets of word features.

The results of all four measures are important to evaluate the performance of these classifiers. Of the four measures, Recall is more important than the others. The better the Recall, the fewer the false-negative (under-classification) errors. In the case of false-positive (over-classification) errors, post processors (the next modules) have a chance to check for these errors automatically. However, false-negative errors cannot be checked and human indexers have to manually find the missing DAN sentences.

Therefore, we try to combine the results of the two classifiers using two merging operators (Equations (10) and (11)) to improve their performance, especially the Recall rate. Table 5 shows the performance of each classifier and operator with the best Recall rate. The Naïve Bayes classifier shows the best Precision rate (97.70%). $G_{Max}(\boldsymbol{x})$ operator shows the best Recall rate (95.36%), and $G_{Harmonic}(\boldsymbol{x})$ operator shows the best F-Measure rate (96.18%) and Accuracy rate (99.07%). As shown in Table 5, both $G_{Max}(\boldsymbol{x})$ and $G_{Harmonic}(\boldsymbol{x})$ operators increase Recall, F-Measure, and Accuracy rates over those resulting from the Naïve Bayes and SVM classifiers.

Table 3. Some word features and corresponding p_i and q_i.

Feature type	**Feature**	$\boldsymbol{p_i}$	$\boldsymbol{q_i}$
Special	Databank Name (GenBank, PDB, etc.)	0.9445288	0.0081475
	Deposit Word (deposited, submitted, etc.)	0.6276595	0.0019296
	Accession Word (accession, access, etc.)	0.6580457	0.0010720
General	accession	0.6512158	0.0006432
	deposited	0.4886010	0.0004280
	foundation	0.1675220	0.0043680
	coordinates	0.2051670	0.0003210
	numbers	0.1626130	0.0022510
	structure	0.1846500	0.0039660

Table 4. Performance of Naïve Bayes and SVM classifiers for each feature set.

	Naïve Bayes					SVM				
Number of word features	**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**Accuracy (%)**		**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**Accuracy (%)**	***C*, *γ***
100	96.57	92.24	94.36	98.63		98.13	91.93	94.93	98.78	0.125, 0.0078125
200	**97.70**	**93.91**	**95.77**	**98.97**		98.72	94.29	**96.46**	**99.14**	0.5, 0.5
300	97.31	93.53	95.38	98.88		98.02	94.60	96.28	99.09	0.5, 0.125
400	97.31	93.53	95.38	98.88		**98.80**	94.22	96.45	**99.14**	8.0, 0.03125
500	97.38	93.53	95.42	98.89		94.83	**94.90**	94.86	98.73	128, 0.000122
600	97.31	93.68	95.46	98.90		98.41	94.44	96.39	99.12	2,048, 0.000122
700	97.16	93.84	95.47	98.90		94.33	**94.90**	94.61	98.66	128, 0.0000305
800	97.23	93.46	95.30	98.86		98.41	94.29	96.31	99.10	2048, 0.0000305
900	97.23	93.61	95.38	98.88		98.33	94.29	96.27	99.09	8192, 0.000305
1,000	97.31	93.68	95.46	98.90		98.33	94.22	96.23	99.08	8, 0.125
1,100	97.31	93.68	95.46	98.90		98.33	94.52	96.39	99.12	8, 0.03125
1,200	97.23	93.53	95.34	98.87		98.10	94.67	96.36	99.11	32, 0.00195

Table 5. Performance of Naïve Bayes and SVM Classifiers and Max and Harmonic operators.

Classifier or Operator	**Number of word feature**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**Accuracy (%)**	**Number of False-Negative**	**Number of False-Positive**	**Number of Total Errors**
Naïve Bayes	200	**97.70**	93.91	95.77	98.97	80	29	109
SVM	500	94.83	94.90	94.86	98.73	67	68	135
$G_{Max}(\boldsymbol{x})$	1,000	96.83	**95.36**	96.09	99.04	**61**	41	102
$G_{Harmonic}(\boldsymbol{x})$	700	97.50	94.90	**96.18**	**99.07**	67	**32**	**99**

Tables 6 and 7 show examples of false-negative and false-positive errors made by the two classifiers, respectively. In Table 6, although the first sentence contains a DAN "DQ022369", the Naïve Bayes classifier does not classify it as a DAN sentence. Also in Table 6, the SVM classifier could not correctly classify the third sentence as a DAN sentence ("AF241848" is a DAN). Table 7 shows examples of false-positive errors. Although they do not contain DANs, all four sentences are misclassified as DAN sentences because of suggestive words such as "OMIM", "Online Mendelian Inheritance in Man", "amino acid sequence", "amino acid sequence" and "pDB".

Table 6. Examples of false-negative (under-classification) errors made by the Naïve Bayes and SVM classifiers.

Classifier	**Sentence**
Naïve Bayes	**1**. Differential display generated rabbit Fn1 cDNA clone sequence (477 bp,Ac #**DQ022369**,this study).
	2. Ribbon diagrams of ubiquitin (**1UBI**),Urm1 (**2AX5**),MoaD (**1FMA** chainD),and ThiS (**1F0Z**) are located on the right.
SVM	**3**. Extensive analysis of the sequence **AF241848** that contains promoter area of RFP2 was performed.
	4. Deletions for TR2 and TR2A included bp 2754 to 3323 (**DQ360502**) and bp 345279 to 346423 (**NC_005139**), respectively.

Table 7. Examples of false-positive (over-classification) errors made by the Naïve Bayes and SVM classifiers.

Classifier	Sentence
Naïve Bayes	**1**. McKusick,V A (**2000**) Online Mendelian Inheritance in Man, OMIM, Bethesda, MD, Available at www.ncbi.nlm.nih.gov Prevent Blindness America Skokie,IL Available at http://www.preventblindness.org Accessed December 1,**2004**.
	2. Hydropathy plot of the mouse SCD1 amino acid sequence and the design of the epitope-tagged **SCD1** constructs.
SVM	**3**. Representation of the region of **IE62** containing ORF66-directed phosphorylation sites,with the amino acid sequence in single-letter code and the position of each serine or threonine residue indicated with the residue number above.
	4. The PCR products were digested with NotI/NsiI,and then ligated into an intermediate vector,**pDB25**,which contains the NotI/BclI fragment of pTN201 in pET28A,for ease of cloning.

6. CONCLUSIONS

In this paper we describe our use of two classifiers, Naïve Bayes and SVM, to classify sentences that contain Databank Accession Numbers in online biomedical articles, as s preliminary step to identifying these numbers. We collect words that occur most frequently in DAN and Non-DAN sentences as word features for the classifiers. To find the optimum number of word features, we collect twelve sets of word features with different sizes to train and test the classifiers. Both classifiers show relatively good performance in all sets, although the SVM classifier shows a little better performance in several sets. The Naïve classifier shows the best performance when the number of word features is 200 in all four measures. However, the SVM classifier does not show the best performance in all four measures in any set. This classifier shows the best performance for Precision at 400 word features, Recall at 500 and 700, F-Measure at 200, and Accuracy at 200 and 400. The best Recall rate of the Naïve Bayes classifier is 93.91% and that of the SVM classifier is 94.90%. We use two merging operators to combine results of the Naïve Bayes and SVM classifiers to improve performance, especially for the Recall rate. The merging operators do improve performance, as seen in the results for Recall (95.36%), F-Measure (96.18%), and Accuracy (99.07%) rates. As future work, we intend to find additional methods of collecting sets of word features and different merging operators to further improve performance.

ACKNOWLEDGMENT

This research was supported by the Intramural Research Program of the National Institutes of Health, National Library of Medicine, and Lister Hill National Center for Biomedical Communications.

REFERENCES

[1] "Technical Memorandum 347: Databank Accession Numbers," National Institutes of Health, National Library of Medicine, October, 1993.

[2] J. Kim, D. X. Le, and G. R. Thoma, "Automatic Extraction of Bibliographic Information from Biomedical Online Journal Articles Using s String Matching Algorithm," *Proceeding of 19th IEEE Symposium on Computer-Based Medical Systems*, pp. 905-910, 2006.

[3] D. D. Lewis, "Naïve (Bayes) at Forty: The Independence Assumption in Information Retrieval," *ECML*, The Tenth European Conference on Machine Learning, pp.4-15, 1998.

[4] A. McCallum and K. Nigam, "A Comparison of Event Models for Naïve Bayes Text Classification," *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, pp.577, 1998.

[5] D. Madigan, "Statistics and the war on spam," *Statistics*: *A Guide to the Unknown*, 4th Ed. (R. Peck, G. Casella, G. Cobb, R. Hoerl, D. Nolan, R. Starbuck and H. Stern, eds.), Thomson Brooks/Cole, Belmont, CA, pp.135–147, 2005.

[6] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the Poor Assumptions of Naïve Bayes Text Classifiers," *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 616-623, 2003.

[7] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998.
[8] T. Joschims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, Chemnitz, DE, pp.137-142, 1998
[9] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. "Inductive learning algorithms and representations for text categorization," In *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, Washington, pp. 148–155, 1998.
[10] E. Gabrilovich and S. Markovitch, "Text Categorization with Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive with C4.5", In *ICML'04* pp. 321- 328, 2004.
[11] http://www.ncbi.nlm.nih.gov/Genbank/
[12] http://clinicaltrials.gov/
[13] http://www.ncbi.nlm.nih.gov/geo/
[14] http://isrctn.org/
[15] http://www.ncbi.nlm.nih.gov/RefSeq/
[16] http://www.ncbi.nlm.nih.gov/Omim/
[17] http://www.pdb.org/
[18] http://pubchem.ncbi.nlm.nih.gov/
[19] C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[20] M. Johnson, SVM.NET, 2008. Software available at http://www.matthewajohnson.org/index.html.
[21] S. Sohn, W. Kim, D. C. Comeau, and W. J. Wilbur, "Optimal Training Sets for Bayesian Prediction of MeSH Assignment," *Journal of the American Medical Informatics Association*, Vo. 15, No. 4, pp.546-553, 2008.