# Context-based Term Disambiguation in Biomedical Literature

**Ping Chen**
University of Houston –Downtown
Houston, Texas 77002
chenp@uhd.edu

**Hisham Al-Mubaid**
University of Houston – Clear Lake
Houston, Texas 77058
hisham@uhcl.edu

**Abstract**

The huge volumes of unstructured texts available online drives the increasing need for automated techniques to analyze and extract knowledge from these repositories of information. Resolving the ambiguity in these texts is an important step for any following analysis tasks. In this paper, we present a new method for one type of ambiguity resolving -- term disambiguation. The method is based on machine learning and can be viewed as a context-based classification approach. In our experiments we apply it to gene and protein name disambiguation. We have extensively evaluated our method using around 600,000 *Medline* abstracts and three different classifiers. The results show that our technique is effective in achieving impressive accuracy, precision, and recall rates, and outperforms the recently published results on this problem. The paper includes the details of the method and the experimental design. We plan to apply our technique to the general domain of word sense disambiguation in the future.

## 1. Introduction

Users' knowledge of words in a language constitutes the *Lexicon*, which includes lots of "character string to real entity" mappings. Memorization of these mappings proves to be hard for human beings. This is why in many natural languages a word is often used to represent multiple entities or meanings, which relieves the task of memorization but makes language interpretation harder. Each meaning of a word is called a sense. In natural language processing (NLP) word sense disambiguation (WSD) is used to determine which sense of a word should be adopted for each instance of a word. WSD methods are usually evaluated manually, which is very labor-intensive and time-consuming. In this paper we focus on disambiguation of terms used in a specific domain. Term disambiguation is very important and can help readers greatly. On the other hand, a specific domain can serve as a small-scaled testbed and let us focus on essential issues of disambiguation. After we gain enough insights, we can expand to the general area of WSD.

### 1.1 Selection of the biomedical domain

The existing volumes of biomedical texts available electronically are gigantic and grow at very high and unprecedented rates [1][7][9][13]. Massive wealth of knowledge is embedded in these documents and needs to be discovered and extracted. Thus there is a great need for efficient and effective machine learning, text mining and NLP techniques to analyze these texts for the advancement of the science [17][23]. Moreover, considerable amounts of research have been put into the problem of named entity recognition (NER) and disambiguation in biomedical literature [2][8]. Such a task is very important for extracting and disambiguating biological entities, which is critical for the development of bioinformatics systems (e.g., discovering associations among the various biological entities, such as gene-disease associations).

### 1.2 Selection of Gene/Protein for Term Disambiguation

In the biomedical literature, resolving ambiguity between genes and proteins is very important 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. A common example of gene and protein name ambiguity (discussed in [7][8][14]) is:

– "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 a gene. Gene and protein are often closely correlated and their disambiguation sometimes is difficult even for biomedical experts. So authors often have to explicitly put "protein" or "gene" before or after the protein or gene names. These explicitly disambiguated protein or gene names can be used for automatic evaluation of a disambiguation technique.

In this paper we present a new context-based term disambiguation method. We applied our method to the gene and protein name disambiguation and conducted extensive experiments using over 600,000 *PubMed* abstracts published from 2001 to 2004 containing more than *200,000* instances of ambiguous gene and protein names. The rest of the paper is organized as follows. In the next section we discuss the recently published work on this problem. Section 3 explains our method. Section 4 describes the experiments and results. Finally, Section 5 provides conclusion and future work.

## 2. Related Work

The applications of text mining and machine learning techniques in the biomedical domain were successful and encouraging. Large number of methods were proposed for various bioinformatics problems such as information

extraction, extracting gene and protein interactions, named entity recognition (NER), association rules discovery and extracting relationships among various biological entities [1][2][4][10][15][18][22][23]. However, the research work invested in biological term disambiguation [11] is not enough compared to the magnitude of this problem as biological entities are highly ambiguous and their disambiguation is often a prerequisite for any following text mining tasks [21].

The gene-protein disambiguation can be processed as a classification problem. In [7][14] authors use *SVMs* to train the term classifiers, and assign different weights to contextual words (*features)*. The method in [8] is also based on classification where three learners, *Naïve Bayes*, *Ripper* and *C4.5*, were used for training and testing. Major contribution of our method is its unique way of selecting the features of the ambiguous terms and building feature vectors. Our method achieves better recall, precision and accuracy rates, and experiment results will be presented in Section 4.

## 3. The Proposed Method

Using a word's context (surrounding words) for disambiguation is an effective technique [7][8][14]. In our method, instead of directly using all of a word's surrounding words, we only select certain words with high "discriminating" capabilities as features. By this way we can discard those "noisy" surrounding words and improve the disambiguation quality. These features will be used to represent each instance of the terms in the training and testing phases. The proposed method is based on the common machine learning approaches to train classifiers with labeled examples. We use author-disambiguated terms as labeled training examples. The classifiers will then be used to disambiguate unseen and unlabeled examples in the testing phase. The next subsection explains the feature selection technique in our method.

### 3.1 Feature Selection

Feature selection is a key issue in machine learning, and quality of features will greatly impact the performance of a machine learning technique. Our experiments shown in table 3 and 4 illustrate that how important it is to select appropriate features and assign right weights to these features. A lot of research has been devoted to feature selection [5][6][24][25].

Let us assume that we have two classes $C_1$ and $C_2$ of labeled examples extracted from biomedical texts for training. And $C_1$ contains examples of *gene* names and their contexts, whereas $C_2$ includes examples of *protein* names with their contexts. The *name* which represents a protein or a gene is the *term* to be disambiguated 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 follows (Note: "protein" and "gene" supplied by authors for disambiguation have been deleted before selecting a term's context words):

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

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) surrounding the *term*, and *n* is called the *window size*. We extract all the context words $W = \{w_1, w_2, \ldots, w_m\}$ from the examples in the sets $C_1$ and $C_2$. (From now on we use $w_i$ to represent these surrounding words instead of $p_i$ and $f_i$). Now, each such context word $w_i$ ($w_i \in W$) may occur in contexts of either or both of $C_1$ and $C_2$ with different frequency distributions. We want to determine that to what extent an occurrence of $w_i$ in an ambiguous example suggests that this example belongs to $C_1$ or $C_2$. Thus, we only 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 feature selection techniques like *mutual information* (MI) and *chi-square* ($X^2$) ([6][24]) to select the highly discriminating context words from $W$.

Let us first define the notions of *a, b, c,* and *d:* from the training examples, we calculate four 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:

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

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

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

When using the *MI* technique for feature selection, we calculate *MI* values for all $w_i \in W$, then we choose the top *k* words with the highest *MI* values as features in this term's *feature vectors*. In our experiments we tested on *k* values of 10, 20, and 30, 40, and 200. For example, if *k = 10*, then each training example is represented by a vector of 10 entries such that the first entry represents the word with the highest *MI* value, and 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 a value $V^+$ if the corresponding feature word occurs in that training example and set to a different value $V^-$ otherwise. Thus, if we want to utilize the 20

most discriminating words as features to represent each term, then feature vector size will be 20.

Consider the following example, let W = {$w_1$, $w_2$, …, $w_m$} be the set of all context words. We compute *MI* for each $w_i \in W$ and sort the words W according to their *MI* values in descending order as in Table 1.

| **Context words** $w_i$ | **MI** |
|---|---|
| *activate* | 3.9 |
| *process* | 3.6 |
| *sample* | 3.2 |
| *deliver* | 2.6 |
| *inhibit* | 1.9 |
| *went* | 1.9 |
| *generate* | 1.8 |
| *smear* | 1.8 |
| *diagnose* | 1.6 |
| *clear* | 1.4 |
| … | … |

**Table 1 Words with the top MI values**

Table 1 shows 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 terms to be disambiguated.

For example, the following feature vector:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $V^-$ | $V^+$ | $V^+$ | $V^-$ | $V^-$ | $V^-$ | $V^+$ | $V^-$ | $V^-$ | $V^-$ |

represents an example containing the 2nd, 3rd and 7th feature words (*i.e.*, *process*, *sample*, and *generate)* around the term within certain window size. Additionally, if the window size *w=4*, then that example will 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 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 translates to a "$V^+$" in the seventh entry of the feature vector.

Selecting only features with high discriminating capabilities results in better disambiguation quality than existing methods as shown in our experiment 1. Choosing appropriate values for $V^-$ and $V^+$ can improve the disambiguation quality further as shown by our experiment 2. In experiment 1 we simply set $V^-$ as 0 and $V^+$ as 1, we can achieve the accuracy rate of 0.8 - 0.9, which is already better than existing methods. In experiment 2, we set $V^-$ as -*MI* and $V^+$ as *MI*, and we achieved the accuracy rate of around 0.99. For example, if we use 0 and 1 for $V^-$ and $V^+$, the feature vector, in the previous example, will be as follows if we use 0 and 1 for $V^-$ and $V^+$:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $0^-$ | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

If we set $V^-$ = -MI and $V^+$ = MI, the vector will be:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| -3.9 | 3.6 | 3.2 | -2.6 | -1.9 | -1.9 | 1.8 | -1.8 | -1.6 | -1.6 |

Similarly we can build feature vectors with $X^2$ values, which is equally effective as shown by our experiments.

### 3.2 Learning and Classification

After we generate feature vectors using the top words selected by *MI* or $X^2$, we can use any classification method to classify instances and find out the classes they belong to. In our experiments we used *Support Vector Machines* (SVM) [3], Decision Table (a decision-rule based learning algorithm), and Naïve Bayes. All these three methods are widely applied and accepted in the field of machine learning.

To be more specific, we construct one feature vector for each instance of the term to be disambiguated. Then we divide these vectors into one class of positive vectors and one class of negative vectors. A classification method is trained on these two classes and produces a classifier (model). And we use this classifier to classify data samples in testing set and calculate accuracy, recall and precision rates. More formally, we describe our approach with the following algorithm:

---

**Algorithm**

*Input*: text document set *D*, ambiguous term *t*

*(Here we assume t has two senses, $t_1$ and $t_2$, a more general case is discussed later)*

1. Extract from set *D* all instances of $t_1$ *and* $t_2$ that have been disambiguated by authors and their contexts, and save them to $S_1$ and $S_2$ respectively.
2. Construct the set $W = \{w_1, w_2, \ldots, w_m\}$ of neighborhood words from $S_1$ and $S_2$ after removing context words supplied by authors for disambiguation, *i.e.*, *"gene"* or *"protein"*.

   $W = S_1 \cup S_2$ – author-inserted disambiguating features
4. Compute *MI* and $X^2$ values for each word $w_i$ in *W*.
5. Sort the set *W* based on *MI* and $X^2$ respectively in descending order.
6. Construct feature vectors for all extracted gene/protein instances based on the top *n* words in *W*. *n* is the size of feature vector.
7. Use these feature vectors as input to a classification method for creating classifiers.
8. Use generated classifiers to classify testing samples for evaluation.

---

So far we have discussed how to disambiguate terms with two senses, which is called two-way classification. For terms with more than two senses, our technique can be

used to disambiguate the class of sense 1 and the class of non-sense 1 first, then within the class of non-sense 1 classify the class of sense 2 and the class of non-sense 2. By repeating this process, terms with more than two senses can be classified and disambiguated accordingly.

## 4. Experiments

### 4.1 Data Collection

We retrieved our data from *Medline* using the *E-Utilities* in the *PubMed* interface [13]. Medline database is considered the main source of data for bioinformatics research as it contains huge collection of biomedical documents (about ~14 millions research abstracts, dated back to 1950) and its wide coverage of medical fields [7][8][10][17][18][23]. We carried out some preprocessing steps on the downloaded *Medline* texts:

1) Converting all words to *lowercase*.
2) Removing *stopwords*: removing all common function words like "is", "the", "in", etc.
3) Performing word *stemming* (converting each word to its root, *e.g., activated→activate, activates→activate*) using *Porter* stemming algorithm [16].

After the text is preprocessed, we extract the gene and protein name occurrences that are already disambiguated by the authors, that is, each gene/protein name occurrence should be preceded or followed by the word "*gene*" or "*protein*". To recognize a gene or protein we have downloaded and compiled gene and protein name dictionaries from three databases *SwissProt, Tremble, LocusLink* [12][19][20]. Thus with a simple and efficient search algorithm we extracted all such *unambiguous* gene and protein name instances taking the formats:

….gene *gene-name*…
…*gene-name* gene….
…protein *protein-name*….
…*protein-name* protein….

giving that the complete *gene-name* or *protein-name* is also found in the gene or protein dictionary. Moreover, we extract with each instance its context of 10 preceding and 10 following words (*i.e.*, window size is 10) which enables us to test on windows of any size up to 10.

### 4.2 Experiments

We conducted several experiments with different vector size and feature selection methods. In these experiments, we used 2-fold cross-validation when the data is divided into 50%/50% for training/testing, whereas, 5-fold cross-validation is used with 80%/20% for training/testing. To be more specific, in 5-fold cross-validation, we divided the data into 5 equal folds, then we use 4 folds (80% of the data) to train the classifier (learning phase) and the remaining fold (20%) will be used to test the classifier. This process is repeated 5 times; each time, one of the 5 folds is used for testing and the remaining 4 folds for training. 2-fold cross-validation is performed similarly.

For performance metrics, in each experiment, we record the accuracy, precision, and recall [11]. In our experiments, we use gene class as positive class and protein as negative class.

For the purpose of comparison we summarized experiment results from [7] and [8] in Table 2.

| Method | Accuracy range | Comments |
|---|---|---|
| New Method - *weighted* [7] | 82.37–86.12 | *with and without collocations, and different context sizes (n).* |
| New Method - *unweighted* [7] | 81.21 – 82.47 | *with and without collocations, and different context sizes (n).* |
| Naïve Bayes [7] | 77.38–84.44 | *Different smoothing techniques.* |
| RIPPER [8] | 74.59% | |
| C4.5 [8] | 76.61% | |
| Naïve Bayes [8] | 76.57% | |

**Table 2: Experiment results from** [7] **and** [8]

**Experiment 1:** In Table 3 we show our experiments using 20,000 *Medline* abstracts from year 2002. Within these abstracts we found 6106 occurrences of gene/protein names. We set $V^- = 0$ and $V^+ = 1$ in the feature vectors. We performed 5-fold cross validation with SVM, and get good accuracy, precision and recall rates. With different sizes of vectors, our technique shows a very good and stable performance supporting the effectiveness of the method. We notice in this experiment that $X^2$ gives better results than MI and the best results in this experiment are obtained with $X^2$ and vectors of size 40.

**Experiment 2:** In this experiment, we collected all medical abstracts available in *Medline* published from 2001 to 2004. There are totally around 600,000 abstracts. From these abstracts we extracted about 200,000 unambiguous gene and protein name instances. These instances were explicitly disambiguated by authors with "*gene*" or "*protein*" immediately before or after each instance. We set $V^-$ as $-MI$ or $-X^2$, and $V^+$ as $MI$ or $X^2$ in the feature vectors. We performed 2-fold and 5-fold cross-validation and recorded the average of the testing cycles using three classification methods, SVM, Decision Table and Naïve Bayes. The vector size for both $MI$ and $X^2$ is fixed to 20, and the results are shown in Table 4. The accuracy, precision and recall rates are approaching or exceeding 99%, which indicate that our method is

highly effective. Also this experiment shows that quality of feature vectors is very important.

## 5. Conclusion and Future Work

We presented a new context-based technique for term disambiguation and applied it to gene/protein name disambiguation in biomedical documents. The experiments clearly show that our method is effective and the strength of the method comes from its unique way of selecting and encoding the features of the target terms into the feature vectors. The two most relevant techniques produced best accuracy in the range of ~0.80 – ~0.86, while our method constantly achieved the accuracy of ~0.99 with different document sets and experiment settings.

In future, we would like to conduct experiments on all available Medline abstracts and apply our method to the general field of word sense disambiguation.

## 6. References


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

[4] C. Creighton, S. Hanash. Mining gene expression databases for association rules. Bioinformatics, 19-1:79--86, 2003.

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

[6] 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. 2000.

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

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

[9] 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.

[10] D. Hristovski, J. Stare, B. Peterlin, S. Dzeroski (2001). Supporting discovery in medicine by association rule mining in Medline and UMLS. Proceeding of MedInfo Conf., London, England, 2001, 10(2), 1344-1348.

[11] H. Liu, S.B. Johnson, C. Freidman. Automatic Resolution of Ambiguous Terms Based on Machine Learning and Conceptual Relations in the UMLS. Journal of the American Medical Informatics Association Vol. 9(6), 2002.

[12] LocusLink at NCBI: www.ncbi.nlm.nih.gov/LocusLink/

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

[14] 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.

[15] M. Palakal, M. Stephens, S. Mukhopadhyay, R. Raje, S. Rhodes. A Multi-level Text Mining Method to Extract Biological Relationships, Proceedings of IEEE Computer Society Bioinformatics (CSB) Conference, 2002, pp. 97--108, appeared in August 2002.

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

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

[18] P. Srinivasan, and B. Libbus, Mining MEDLINE for Implicit Links between Dietary Substances and Diseases. ISMB 2004 and in Bioinformatics (Supplement)

[19] Swissport home: www.ebi.ac.uk/swissprot/

[20] Tremble: http://www.ebi.ac.uk/trembl/

[21] O. Tuason, L. Chen, H. Liu, J. A. Blake, C. Friedman. Biological nomenclatures: A source of lexical knowledge and ambiguity. Pacific Symposium on Biocomputing 9:238-249(2004)

[22] M. Weeber, et. Al., Generating hypothesis by discovering implicit Associations in the literature: a case report of a search for new potential therapeutic uses of thalidomide, *.J.Am Med. Inform.Assoc*. vol.10, 252-259, 2003.

[23] 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.

[24] Y. Yang, J.P. Pedersen . A comparative study on feature selection in text categorization. *The 4th Intl Conf. on Machine Learning*, 1997

[25] Z. Zheng, R. Srihari. Optimally combining positive and negative feature for text categorization, ICML'2003 Workshop on Learning from Imbalanced Data Sets, 2003.


| Method | | Training set size (80%) | Testing set size (20%) | Accuracy | Precision | Recall | incorrect instances |
|---|---|---|---|---|---|---|---|
| $X^2$ | 20 fields | 4885 | 1221 | 85.0% | 86.3% | 96.7% | 183 |
| | 30 fields | 4885 | 1221 | 85.5% | 87.0% | 96.2% | 177 |
| | 40 fields | 4885 | 1221 | 86.4% | 87.9% | 96.3% | 166 |
| MI | 20 fields | 4885 | 1221 | 80.4% | 81.1% | 98.4% | 239 |
| | 30 fields | 4885 | 1221 | 80.2% | 81.2% | 98.0% | 242 |
| | 40 fields | 4885 | 1221 | 80.2% | 81.2% | 97.9% | 242 |

**TABLE 3:** Result of 5-fold cross validation experiment using SVM and vectors of 20, 30 or 40 fields

| Year | Classifier | Method | | Training set size | Testing set size | Accuracy | Precision | Recall | Incorrect instances |
|---|---|---|---|---|---|---|---|---|---|
| 2001 | SVM | $X^2$ | 2-fold | 24654 | 24654 | 99.30% | 98.45% | 100% | 173 |
| | | | 5-fold | 39447 | 9861 | 99.30% | 98.44% | 100% | 69 |
| | | MI | 2-fold | 24654 | 24654 | 99.76% | 100% | 99.46% | 59 |
| | | | 5-fold | 39447 | 9861 | 99.93% | 100% | 99.83% | 7 |
| | Naïve Bayes | $X^2$ | 2-fold | 25593 | 25593 | 99.1% | 99.1% | 99.1% | 220 |
| | | | 5-fold | 40950 | 10236 | 99.0% | 99.0% | 99.0% | 242 |
| | | MI | 2-fold | 25593 | 25593 | 99.2% | 99.3% | 99.2% | 187 |
| | | | 5-fold | 40950 | 10236 | 99.2% | 99.3% | 99.2% | 186 |
| | Decision Table | $X^2$ | 2-fold | 27306 | 27306 | 100% | 100% | 100% | 2 |
| | | | 5-fold | 43690 | 10922 | 100% | 100% | 100% | 0 |
| | | MI | 2-fold | 27306 | 27306 | 100% | 100% | 100% | 0 |
| | | | 5-fold | 43690 | 10922 | 100% | 100% | 100% | 0 |
| 2002 | SVM | $X^2$ | 2-fold | 29337 | 29337 | 99.54% | 98.99% | 100% | 136 |
| | | | 5-fold | 46939 | 11735 | 99.55% | 98.99% | 100% | 54 |
| | | MI | 2-fold | 29337 | 29337 | 99.55% | 100% | 99.00% | 134 |
| | | | 5-fold | 46939 | 11735 | 99.43% | 100% | 98.74% | 67 |
| | Naïve Bayes | $X^2$ | 2-fold | 24654 | 24654 | 99.3% | 99.2% | 99.3% | 207 |
| | | | 5-fold | 39447 | 9861 | 99.2% | 99.1% | 99.2% | 226 |
| | | MI | 2-fold | 24654 | 24654 | 99.0% | 99.2% | 99.0% | 232 |
| | | | 5-fold | 39447 | 9861 | 99.0% | 99.2% | 99.0% | 225 |
| | Decision Table | $X^2$ | 2-fold | 25593 | 25593 | 100% | 100% | 100% | 0 |
| | | | 5-fold | 40950 | 10236 | 100% | 100% | 100% | 0 |
| | | MI | 2-fold | 25593 | 25593 | 100% | 100% | 100% | 0 |
| | | | 5-fold | 40950 | 10236 | 100% | 100% | 100% | 0 |
| 2003 | SVM | $X^2$ | 2-fold | 27306 | 27306 | 99.82% | 99.59% | 100% | 50 |
| | | | 5-fold | 43690 | 10922 | 99.82% | 99.59% | 100% | 20 |
| | | MI | 2-fold | 27306 | 27306 | 99.76% | 100% | 99.46% | 66 |
| | | | 5-fold | 43690 | 10922 | 99.90% | 100% | 99.77% | 11 |
| | Naïve Bayes | $X^2$ | 2-fold | 29337 | 29337 | 99.5% | 99.5% | 99.5% | 146 |
| | | | 5-fold | 46939 | 11735 | 99.5% | 99.5% | 99.5% | 145 |
| | | MI | 2-fold | 29337 | 29337 | 99.1% | 99.1% | 99.1% | 261 |
| | | | 5-fold | 46939 | 11735 | 99.2% | 99.2% | 99.2% | 238 |
| | Decision Table | $X^2$ | 2-fold | 24654 | 24654 | 100% | 100% | 100% | 0 |
| | | | 5-fold | 39447 | 9861 | 100% | 100% | 100% | 0 |
| | | MI | 2-fold | 24654 | 24654 | 100% | 100% | 100% | 0 |
| | | | 5-fold | 39447 | 9861 | 100% | 100% | 100% | 0 |
| 2004 | SVM | $X^2$ | 2-fold | 25593 | 25593 | 98.55% | 96.84% | 100% | 372 |
| | | | 5-fold | 40950 | 10236 | 99.37% | 98.61% | 100% | 64 |
| | | MI | 2-fold | 25593 | 25593 | 99.76% | 100% | 99.45% | 63 |
| | | | 5-fold | 40950 | 10236 | 99.87% | 100% | 99.69% | 13 |
| | Naïve Bayes | $X^2$ | 2-fold | 27306 | 27306 | 99.0% | 98.9% | 99.0% | 315 |
| | | | 5-fold | 43690 | 10922 | 99.0% | 98.9% | 99.0% | 318 |
| | | MI | 2-fold | 27306 | 27306 | 99.0% | 98.9% | 99.0% | 315 |
| | | | 5-fold | 43690 | 10922 | 99.4% | 99.4% | 99.4% | 178 |
| | Decision Table | $X^2$ | 2-fold | 29337 | 29337 | 100% | 100% | 100% | 0 |
| | | | 5-fold | 46939 | 11735 | 100% | 100% | 100% | 0 |
| | | MI | 2-fold | 29337 | 29337 | 100% | 100% | 100% | 0 |
| | | | 5-fold | 46939 | 11735 | 100% | 100% | 100% | 0 |

**TABLE 4:** Results of 2-fold and 5-fold cross validation experiment using SVM, Decision Table and Naïve Bayes with vector of size 20