Word Classification: An Experimental Approach with Naïve Bayes

Wei Ding
ding@cs.umb.edu
University of Massachusetts
Boston
Boston, MA 02125

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

Srikanth Kotagiri
Srikanthk@gmail.com
University of Houston-Clear
Lake,
Houston, TX 77058 USA

Abstract

Word classification is of significant interest in the domain of natural language processing and it has direct applications in information retrieval and knowledge discovery. This paper presents an experimental method using Naïve Bayes for word classification. The method is based on combing successful feature selection techniques on Mutual Information and Chi-Square with Naïve Bayes for word classification. We utilize the advances in feature-selection techniques in information retrieval and propose an efficient method to select key features for term identification and classification. We evaluate the method using real-world texts taken from the Wall Street Journal news articles. The experimental results proved that the method is fairly effective and competitive for word classification.

1. Introduction

Word classification is of the great interest in the domain of natural language processing (NLP) [3, 5, 9, 10, 12, 15, 18]. The knowledge and information embedded in the literature and textual repositories are extremely massive, especially with the increasing dependence on the *Internet* use for information sharing and storage. This drives a great need to discover useful and significant knowledge (*knowledge discovery*) and extract information and relations (*information extraction*) to benefit all fields of human knowledge [2, 22].

Most of the effective knowledge discovery and information extraction techniques include an essential component for word classification and term disambiguation [3, 4, 7, 9, 12, 13, 16, 19, 20, 30]. In other domains, for example, in biomedical fields, significant amount of research projects have been devoted to problems related to biomedical terms identification and classification [4, 6, 17]. Moreover, in NLP, each meaning of a word is called a *sense*, and word sense disambiguation (WSD) is used to determine which sense of a word should be adopted for each instance of a word in given contexts [4]. Hence, this problem can be casted as sense classification. Furthermore, the word prediction task is one of the direct applications of word classification [3, 10, 11]. The word prediction task targets on predicting the most suitable word in a given context to offer a correct word to the user for text completion. Thus, word completion utilities can save the user some keystrokes, especially in enhancing the communication rate of people who have difficulties to type quickly [3, 10].

In a word classification task, we want to assign the correct word into a given context based on prior knowledge. For example, in a sentence "*Word classification is one of the important {tasks vs. targets} in natural language processing NLP*", we want to determine which word in {*tasks*, *targets*} is the correct word in this given context. Thus, the word classification task is viewed as multiple candidate words are to be classified to determine the most correct one in a given context.

One of the immediate applications of word classification is the task of context-sensitive spelling-error correction, or *malapropisms* [5, 16]. According to this problem, a misspelled variant of the original word is a correct word and belongs to the language [5, 16]. For example, the misspelling of the word *quite* as *quiet* is a context-sensitive spelling error. Since *quiet* is a valid word in English, a traditional spell-checkers is not able to discover this spelling error. Thus, the function of the context-sensitive spelling correction is to choose, for an instance of a word in text, for example, *quite*, as its correct spelling from its confusion set {*quite, quiet*}.

In this paper, we present an experimental method using Naïve Bayes and feature selection techniques

Mutual Information (*MI)* and Chi-Square (X^2) for word classification. The feature selection techniques of *MI* and X^2 are utilized to select the key features in the contexts of the target words. We combine these advances in feature selection techniques, from information retrieval, with the power of Naïve Bayes to assign the correct word in a given context. We evaluate the proposed method using real-world texts from Linguistic Data Consortium [1] taken from the Wall Street Journal. The experimental results have proven that the method is fairly effective and competitive for word classification.

2. Related Work

In [11], a comprehensive review of prior related work in word prediction and classification is presented. Fazly in [11] also presented a collection of experiments on word prediction applied to word completion utilities. The implemented and evaluated algorithms in [11] were based on word unigrams and bigrams, and based on syntactic features like POS tags in the syntactic predictors, and combination.

Among the other related interesting work is the approach presented in [10]. That approach attempted to learn the contexts in which a word tends to appear, using expressive and rich set of features. The features are introduced in a language as information sources. The method also augmented local context information by global sentence information. The evaluation of the method in this paper is very similar to what has been presented in [10].

Naïve Bayes has also been used in a web-page classification system [12]. The Bayesian prior probabilities of term counts and frequencies were used for the learning and classification of web pages.

The feature extraction and selection techniques used in this work, *MI* and X^2, have been used successfully in information retrieval (IR) and text categorization (TC) [13, 15, 23].

Commonly Confused Words
{for, four}, {life, like}, {buy, by}, {may, many}, {were, where}, {back, bank}, {done, down}, {from, form}, {loss, lot}, {real, rate}, {sale, same}, {avoid, agree}, {fall, fast}, {offer, office}, {range, rather}, {since, sources}, {should, share}

Table 1. List of confusion pairs.

3. An Experimental Approach with Naïve Bayes

Our approach for word classification is based on representing each word using a feature vector, and then using Bayesian learning to acquire the prior knowledge and probabilities of the word. The constructed prior knowledge is then employed to determine, from a confusion set, the correct word in a given context. For example, let the confusion set in a given context be {*weak*, *week*} then we want to use word classification to determine which word of {*weak*, *week*} is the most appropriate in that given context.

In a given context (e. g., [… w_3 w_2 w_1 $\{\underline{w_x,w_y}\}$ u_1 u_2 u_3…]), we want to classify between the words w_x and w_y, to choose the correct one in this context. In this work, we follow the majority of researchers and assume that the confusion sets are predetermined [9, 10, 16]. Each confusion-set contains two or more of the mostly confused words in the language. For example, Table 1 contains a list of confusion sets (confusion pairs) used in this work and other similar work in literature.

Now we can summarize the problem as follows. Let $c = \{ w_n \ldots w_2, w_1 \_\_ u_1, u_2, \ldots, u_n\}$ be the context of the word classification task, where n is an integer number represents the size of a context window (for example, 3, 5, and 10). The words $w_1, w_2, \ldots, w_n$ and $u_1, u_2, \ldots, u_n$ are the words that appear immediately before and after the target word. Also let $f = \{w_x , w_y\}$ be the confusion set for this case to be predicated in the blank between words w_1. Our approach relies on Bayesian learning to classify whether w_x or w_y is the correct word in that context. Each word in the confusion set is represented as a projection on the feature vector that is composed from the prior data. In the following section, we describe the feature extraction process, and then we talk about the learning and the prediction steps.

3.1. Feature Selection

We use word features to represent the target words in the classification process. But the words in the context of the underlined word are not used directly as features. Instead, we select features only from those words having high *discriminating* capabilities among the various classes of words. This way we can discard those *noisy* surrounding words and improve the classification quality. These *word features* are used to represent each instance (occurrence) of the words in the classification process. The method then uses Naïve Bayesian approach with some already labeled terms (*annotated with class labels*) used as prior knowledge. The Bayesian classifiers will then be used to classify new instances based on the prior knowledge and probabilities. One of the contributions of this work is the way we select features for learning and classification. A great deal of research has been dedicated to feature selection in data mining and machine learning, for example in *text categorization* research; see [13, 15, 23]. Feature selection is an important aspect in the efficiency of the learning in classification methods in general.

Let a training text T be given. We extract from T all the occurrences of the confusion set words w_x and w_y. Each occurrence is extracted along with its context of surrounding words to make one *training* example of the form [… w_3 w_2 w_1 $\underline{w_x}$ u_1 u_2 u_3 …] or [… w_3 w_2 w_1 $\underline{w_y}$ u_1 u_2 u_3 …]. Thus, we have now two sets of examples for learning (for w_x and w_y) extracted from T. We convert each example into a feature vector as follows. The given context words are used as features in some of the related work [7, 19, 20]. In this research, however, we do not use word features directly from the context; instead, we select features only from certain words with high *discriminating* capabilities between the two confused words (w_x and w_y). These features are used to represent each example in the learning and classification. We use the confusion words occurrences extracted from the training text T as labeled training examples.

For a given confusion pair $\{w_{x,}\ w_y\}$, let us assume we have two classes C_1 and C_2 of examples/occurrences extracted from the training text T. Let C_1 contains the examples of w_x and their contexts, and C_2 includes the examples of w_y with their contexts. Then we collect and compile all the context words $W = \{w_1, w_2, \ldots, w_m\}$ from the sets C_1 and C_2. Now, each such context word $w_i \in W$ may occur in contexts from C_1 or C_2 or both with different frequency distributions. That is, if a word $w_j \in W$ is collected from C_1 then with high frequency, then this indicates that w_j is most likely occurs in the neighborhoods of w_x and not w_y, and vice versa. Now, if a context word $w_i \in W$ appears in the context of a classification example, we would like to be able to determine to what extent the existence 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 (for example, the highly discriminating words) as features. We utilize feature selection techniques like *MI* and X^2 [15, 23] to select the highly discriminating context words from W. *MI* and X^2 were used effectively for feature selection in text categorization and information retrieval [13, 15, 23], but not extensively applied for language prediction or classification problems.

We define the four counts a, b, c, and d as follows. From the training examples, we calculate four frequency counts 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, mutual information MI is defined as:

$$MI = \frac{N*a}{(a+b)*(a+c)} \qquad (1)$$

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

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

Again, N is the total number of examples in C_1 and C_2. Let us discuss an example on using the X^2 for feature selection. We calculate the X^2 value for each $w_i \in W$. Then we choose the k top $w_i \in W$ words with the highest X^2 values as features. For example, if $k=10$, then each example in the learning stage is represented by a vector of 10 entries, such that, the first entry represents the word with the highest X^2 value, the second entry represents the word with the second highest X^2 value, and so on. Then for a given training example, the feature vector entry is set to 1 if the corresponding feature word appears in that training example, and set to 0, otherwise. We call this type of feature vectors *binary* feature vectors. The other type of feature vectors is the *non-binary* feature vectors. The *non-binary* vector is obtained from the binary vector by replacing each 0 with $-\boldsymbol{v}$ and each 1 with $+\boldsymbol{v}$, where $\boldsymbol{v}$ is the X^2 (or *MI*) value of the corresponding context word. Thus, if we want to utilize the 20 most discriminating words as features to represent each example, then feature vector size will be 20. Consider the following example, let $W = \{w_{1,}\ w_2, \ldots, w_m\}$ be the set of all context words.

Context words w_i	X^2
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 2. Context words with the highest X^2 values.

We compute X^2 for each $w_i \in W$ and sort the words W according to their X^2 values in descending order as in Table 2. The top context words having the highest ten X^2 values appear in Table 2. These ten words will be used to compose the feature vectors for the classification process. For example, the following *binary* feature vector [0, 1, 1, 0, 0, 0, 1, 0, 0, 0] represents an example containing the 2nd, 3rd and 7th feature words (*process, sample,* and *generate)* in the given context. The *non-binary* feature vector for the same example will look like: [-3.9, 3.6, 3.2, -2.6, -1.9, -1.9, 1.8, -1.8, -1.6, -1.4]. This implies that, if the window size is five, three of the 10 feature words are occurring within the surrounding five words of the word to be classified.

Let us look into the X^2 feature selection technique in little more details. The objective of X^2 is to select from two classes C_1 and C_2 of the examples on the most discriminating word features. A good such feature is the one that is highly associated with C_1 but not with C_2 or vice versa. X^2 uses the co-occurrence counts *a*, *b*, *c*, and *d* with Equation 2 to compute X^2 value for each feature, such that the feature with highest X^2 value will be the best in discriminating C_1 from $C_{2.}$ The X^2's formula gives most weight to *a* and *d* (the numerators in Equation 2), where *a* represents the association between the word feature and class C_1 and the value *b* represents the association between w_i and class C_2 (*i.e.*, how many times w_i occurs in C_2).

3.2. Learning and Classification

After constructing all feature vectors from the prior text, we used the Naïve Bayesian method for the classification task. In applying Naïve Bayes, we followed the general procedure by assuming the probabilistic model of the training examples [10]. Naïve Bayes was applied into many classification and disambiguation tasks like NLP problems, for example, word sense disambiguation [20, 19, 7, 15]. We briefly introduce Naïve Bayes here and describe the experimental settings with it, for more details you can refer to [15, 18]. Let $W = \{w_1, w_2, \dots, w_n\}$ be the context, $C = \{c_1, c_2, \dots, c_m\}$ be the confusion set that contains the alternative (*candidate*) words for the classification task. The decision rule of the Naïve Bayes is as follows:

$$c^* = \underset{k}{\text{argmax}}\ P(c_k|W) = \underset{k}{\text{argmax}}(P(c_k) \,.\, \prod_{i=1}^{n} P(w_i \mid c_k)) \tag{3}$$

such that $P(c_k\ |W)$ is the conditional probability of the confusion set word c_k appears in the context W. This decision rule selects $c^* \in C$ as the correct word in the given context W. The probabilities $\mathrm{P}(c_k)$ and $\mathrm{P}(w_i|c_k)$ are computed from the training text T. Notice here that Naïve Bayesian Network assumes that the context words $w_1, w_2, \dots, w_n$ are conditionally independent. There is one issue here is that the probability $\mathrm{P}(w_i|c_k)$ may, very well, be a very small value or zero, so we use a smoothing technique to avoid this problem. There are a number of smoothing techniques proposed in the literature, for example, for more details on smoothing see [16, 8]. Chen *et al.* (1998) [8] presents a comprehensive review about the smoothing techniques.

4. Evaluation and Results

We have implemented the experimental Bayesian approach for word classification with both *MI* and X^2 techniques for feature selection. We apply our method to a large-scaled real world data corpus and conducted 5,790 experiments. In this section, we describe the datasets, the experimental design, and the final experimental results.

4.1 Datasets

The text corpus for learning and classification is from the ACL dataset obtained from Linguistic Data Consortium (LDC) (www.ldc.upenn.edu). The dataset contains the real news stories of 1987-1991 taken from the Wall Street Journal (WSJ) [1]. The large data corpus includes roughly 1,200,000 English words.

4.2 Experimental Design and Settings

30 pairs of confusion words are randomly selected for testing. Each confusion pair, for example, {*agree, avoid*}, provides two classes for training and classification. We use 5-fold cross validation, such that, we segment the data into 5 equal sized partitions. During each run, one of the partitions is chosen for testing, while the rest of them are used for training. We repeat the procedure 5 times, and in each run, different partition is used for testing—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 experimental texts are preprocessed as follows:

(1) **Case Conversion**: We change all the letters into lower case.

(2) **Word Stemming**: all words converted to their stems using *Porter's* stemming algorithm [21].

(3) **Stopword Removal**: we removed all the function words (*stopwords)* like *'the'*, *'of'*, *'in'*, *'for'*, *'on'*, etc.

We use 3 feature-vector sizes: 100, 500, and 1000. Two different window sizes, which are 10 and 15 words preceding the word to be classified, are used to construct feature vectors. In addition, if a sentence cannot provide sufficient preceding words with respect to a window size, we use two different settings: either stop at the beginning of the sentence or exceed beyond the current sentence to collect words from a previous sentence. Two types of feature vectors are used in our experiments: *binary* feature vectors use 1 and 0 to record the presence and absence of context words, and *non-binary* feature vectors use real values of *MI* and X^2, respectively.

For performance metrics, we use *accuracy* and *precision.* Accuracy is used to evaluate the ratio of the number of correct classifications to the total number of all classification. Because it is more important to correctly predict a right word to be typed by a user, we use precision to evaluate the ratio of the number of true positive predications and the total number of predictions.

4.3 Results and Discussion

We conducted 193 different experiment settings for every 30 confusion pairs using two different feature selection techniques, *MI* and X^2, different vector sizes, binary and non-binary vectors, varying window sizes, and the combinations of word-stemming/non-word-stemming, stopword/non-stopword in text preprocessing steps. For the purpose of comparison, we summarized experiment results by Ginter and Hatzivassiloglou et al. in [16, 17] in Table 3.

Method	**Accuracy range**
New Method – *weighted* [16]	0.82– 0.86
New Method - *Unweighted* [16]	0.81– 0.82
Naïve Bayes [16]	0.77– 0.84
RIPPER [17]	0.75
C4.5 [17]	0.77
Naïve Bayes [17]	0.77

Table 3. Experiment results from Ginter's and Hatzivassiloglou's studies.

Table 4 lists the 30 confusion pairs, the distribution of each word class in the data corpus, and the number of instances used for training and testing.

In Table 5, we report the best 10 results obtained in the experiments with the highest average values on accuracy and precision. Our feature selection method using Naïve Bayesian method out-performs the accuracy rates reported in [17] and match the best resulted done in [16]. In particular, the best parameter setting in our experiments uses X^2 for feature selection with size k=1,000 non-binary feature-vectors, 10 window size for feature selection, non-word-stemming, non-stop-word removal, and non-exceeding current sentence.

We observe that the feature selection X^2 gives the best performance on word prediction. In fact, of the 193 experiment settings, the best performance of *MI* is ranked at 20th with both accuracy and precision values close to 0.75, which is 1% less than the best performance of X^2.

From these results we submit that the proposed feature- selection method is fairly effective and competitive.

5. Conclusion

This paper presents an experimental approach for word classification. The experimental results have shown that the method is effective in classifying words using surrounding context words as features. The feature-selection method *Chi-square* used in this work has proved to be the most efficient method on word classification with respect to the real-world corpus used in the experiments. In the future directions of this research, we would like to investigate a number of new aspects to improve the approach; for example, we can examine other feature selection techniques.

Reference

[1] ACL data set, University of Pennsylvania, Linguistic Data Consortium LDC, http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93T1.

[2] 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.

[3] H. Al-Mubaid. A Learning-Classification Based Approach for Word Prediction. International Arab Journal on Information Technology IAJIT, Vol.4 No.3, July 2007.

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

[5] H. Al-Mubaid and K. Truemper, "Learning to Find Context-Based Spelling Errors," in Triantaphyllou E. and Felici G. (Eds), Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques, Massive Computing Series, Springer, Heidelberg, Germany, pp. 597-628, 2006.

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

No.	Class 1 [C_1]	Class 2 [C_2]	C_1 # instances	C_2 # instances	$C_1 + C_2$ # instances	Training (80%)	Testing (20%)
1	agree	avoid	726	871	1597	1278	319
2	back	bank	1501	1501	3002	2402	600
3	building	business	1501	1501	3002	2402	600
4	buy	by	1501	1501	3002	2402	600
5	class	close	1126	1351	2477	1982	495
6	done	down	769	923	1692	1354	338
7	estimated	earnings	1501	1501	3002	2402	600
8	fall	fast	1103	919	2022	1618	404
9	find	fast	1104	920	2024	1619	405
10	for	four	1501	1501	3002	2402	600
11	found	force	1501	1501	3002	2402	600
12	from	form	1501	1501	3002	2402	600
13	least	leave	1361	1134	2495	1996	499
14	life	like	1501	1501	3002	2402	600
15	loss	lot	1501	1488	2989	2392	597
16	may	many	433	361	794	635	159
17	offer	office	1501	1501	3002	2402	600
18	opportunity	order	764	917	1681	1345	336
19	out	own	1501	1501	3002	2402	600
20	products	price	1501	1501	3002	2402	600
21	range	rather	1103	1060	2163	1370	793
22	real	rate	1501	1501	3002	2402	600
23	sale	same	1501	1501	3002	2402	600
24	should	share	1501	1501	3002	2402	600
25	since	sources	1501	1366	2867	2293	574
26	sold	same	1501	1501	3002	2402	600
27	trade	target	1147	956	2103	1682	421
28	were	where	1501	1501	3002	2402	600
29	when	what	1501	1501	3002	2402	600
30	yield	years	1501	1501	3002	2402	600
Total			39656	39284	78940	62800	16140

Table 4. The set of confusion pairs selected for our experimental evaluations.

Experiments						Accuracy (Average)	Precision (Average)
Feature selection	vector size	window size	word stemming	stopword removal	exceed current sentence		
X^2	1000	10	N	N	N	0.84	0.85
	1000	10	N	Y	N	0.84	0.84
	1000	10	N	N	Y	0.83	0.84
	1000	10	Y	N	Y	0.83	0.83
	500	10	N	N	N	0.83	0.83
	1000	10	N	Y	Y	0.82	0.83
	500	10	N	Y	N	0.83	0.83
	1000	10	Y	N	N	0.82	0.83
	1000	10	Y	Y	Y	0.82	0.82
	1000	10	Y	Y	N	0.82	0.82

Table 5. Results of experiments with the highest values on accuracy and precision. The top 10 results all use non-binary feature vectors.

[7] R. Bruce and J. Wiebe, “Word-Sense Disambiguation Using Decomposable Models,” in Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94), USA, pp. 139-145, 1994.

[8] S. Chen and J. Goodman, “An Empirical Study of Smoothing Techniques for Language Modeling,” Technical Report TR-10-98, Harvard University, Cambridge, Massachusetts, 1998.

[9] Y. Even-Zohar and D. Roth, “A Classification Approach to Word Prediction,” in Proceedings of the NAACL'00, USA, pp. 124-131, May 2000.

[10] Y. Even-Zohar, D. Roth, and D. Zelenko, “Word Prediction and Clustering,” The Bar-Ilan Symp. on the Foundations of Artificial Intelligence, June 1999.

[11] A. Fazly, “The Use of Syntax in Word Completion Utilities,” Master Thesis, University of Toronto, Canada, 2002.

[12] V. Fernandez, R. Unanue and S. Herranz and A. Rubio, “Naive Bayes Web Page Classification with HTML Mark-Up Enrichment,”: Proc.of ICCGI '06, 2006.

[13] G. Forman, “An Extensive Empirical Study of Feature Selection Metrics for Text Classification,” JMLR, vol. 3, no. 1, pp. 1289-1305, 2003.

[14] 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.

[15] W. A. Gale, K. W. Church, and Yarowsky D., “A Method for Disambiguating Word Senses in a Large Corpus,” Computers and the Humanities, vol. 26, no. 1, pp. 415-439, 1992.

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

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

[18] C. Manning and H. Schutze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, Massachusetts, 1999.

[19] T. Pedersen, “Search Techniques for Learning Probabilistic Models for Word Sense Disambiguation,” in Proceedings of the AAAI Spring Symposium on Search Techniques for Problem Solving Under Uncertainty and Incomplete Information, Palo Alto, CA, 1999.

[20] T. Pedersen and R. Bruce, “Knowledge Lean Word Sense Disambiguation,” in Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98), Madison, WI, 1998.

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

[22] 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.

[23] 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.