Security Level Classification of Confidential Documents Written in Turkish

Erdem Alparslan and Hayretdin Bahsi

National Research Institute of Electronics and Cryptology-TUBITAK, Turkey
{ealparslan,bahsi}@uekae.tubitak.gov.tr

Abstract. This article introduces a security level classification methodology of confidential documents written in Turkish language. Internal documents of TUBITAK UEKAE, holding various security levels (unclassified-restricted-secret) were classified within a methodology using Support Vector Machines (SVM's) [1] and naïve bayes classifiers [3][9]. To represent term-document relations a recommended metric "TF-IDF" [2] was chosen to construct a weight matrix. Turkic languages provide a very difficult natural language processing problem in comparison with English: "Stemming". A Turkish stemming tool "zemberek" was used to find out the features without suffix. At the end of the article some experimental results and success metrics are projected.

Keywords: document classification, security, Turkish, support vector machine, naïve bayes, TF-IDF, stemming, data loss prevention.

1 Introduction

In recent years, protecting secure information became a challenge for military and governmental organizations. As a result, well defined security level contents and rules are more preferable than in the past. Each piece of information has its own security level. Correct detection of this security level may lead to apply correct protection rules on information.

Document classification aims to assign predefined class labels to a new document that is not classified [2]. An associated classification framework provides training documents with existing class labels. Therefore, supervised machine learning algorithms are fitting as a solution to classification problems. Well-known machine learning tasks such as Bayesian methods, decision trees, neural networks and support vector machines (SVM) are widely used for classification [3].

Classification accuracy of textual data is highly related to preprocessing tasks of training and test data. [4] These tasks become more difficult in processing unstructured textual data than in structured data. Unstructured nature of data needs to be formatted in a relational and analytical form. In this study TF-IDF (term frequency-inverse document frequency) is preferred to represent text based contents of documents. TF-IDF representation holds each word stem as an attribute for classification; and each document represents a separated classification event.

P. Daras and O. Mayora (Eds.): UCMedia 2009, LNICST 40, pp. 329–334, 2010.

Another important task of formatting textual data is stemming. In this study of Turkish documents, stemming tasks are more difficult than in other studies based on English or in other Latin-based languages. Turkic languages involve diverse exceptional derivation rules. Therefore stemming of Turkish terms provides some unstable rules varying from structure to structure.

Each of the distinct terms mentioned in our text document set is a dimension of this TF-IDF representation. Hence this representation leads to a very high dimensional space, more than 10000s dimensions. It is mainly noted that feature selection tasks are critical to make the use of conventional learning methods possible, to improve generalization accuracy, and to avoid "overfitting" [2].

Recent studies on document classification are performed on text datasets, especially on news stories in English [2][6]. We previously performed a classification of Turkish news stories and obtained a classification accuracy of 90%. [5]

In this study, internal documents of TUBITAK UEKAE (National Research Institute of Electronics and Cryptology) are classified into three classification levels: “secret, restricted and unclassified” by using support vector machines and naïve bayes algorithms.

2 Experimental Settings

In this study, 222 internal documents of TUBITAK UEKAE (National Research Institute of Electronics and Cryptology) are used to develop a framework which has an ability to classify documents according to their security levels by using support vector machines and naïve bayes algorithms.

First, all of 222 internal documents are classified into correct security levels (secret, restricted, unclassified) according to the general policies of TUBITAK UEKAE with the help of an expert. (The numbers of secret, restricted and unclassified documents are 30, 165 and 27 respectively.) Then these classified documents are converted into UTF-8 encoded txt based file format. Training and test documents have totally about 2.5 millions of words except stopping words. All the documents are grouped and arranged in a relational database structure.

These 2.5 million words are unstemmed Turkish words. A comprehensive stemmer library zemberek [8] is used to find out roots of unstemmed words. Zemberek gives us all the possible stemming structures for a term. Our stemming system selects the structure that has the biggest probability of semantic and morphologic patterns of the Turkish language. Table-1 shows some stemming examples of zemberek.

Table 1. Stemming Examples of Zemberek [8]

word	root	suffixes
getirilebilmesi	getir	il + ebil + me + si
yayımladığı	yayım	la + dığ(k) + ı
göstergelerin	gösterge	ler + in
endeksteki	endeks	te + ki
işlevin	işlev	in

Performing the stemming process we obtained approximately 9000 distinct terms. These 9000 terms may cause a high dimensionality and a time consuming

classification process. As we mentioned above SVM's are able to handle high dimensional TF-IDF matrix with the same accuracy; however the time complexity of the classification problem is also an important aspect of our study. Therefore a χ^2 statistics with a threshold (100) was performed to select important features of classification. By the feature selection, the size of selected features is reduced from ~9000 to ~2000. This is known as the corpus of classification process.

The final task performed for the preprocessing phase was constructing a TF-IDF value matrix of all the features for all documents. The application was calculating TF-IDF values for each feature-document pair in the corpus.

Preprocessing software of this study was developed in JAVA, an open-source powerful software development language, and data were stored on PostgreSQL, an open-source professional database.

3 Results and Discussions

SVM-multiclass [7], Joachim's new multiclass support vector machine implementation, was executed on 3 different train and test sets with standard linear kernel and learning parameters. A leave-one-out cross validation was performed to confirm classification accuracy and parameter selection. All the documents have been grouped randomly into 3 train/test set pairs as in table 2.

Table 2. Test sets applied to SVM and NB algorithms

	doc set 1	doc set 2	doc set 3
number of train docs	145	163	136
number of test docs	77	59	86
total	222	222	222

Parameter c of SVM, the trade-off between training error and margin [10] is defined as a high value like 1000. As shown in figure 1, classification accuracy varies respect to the parameter c. The best fitting c value for all these 3 document sets is 1000.

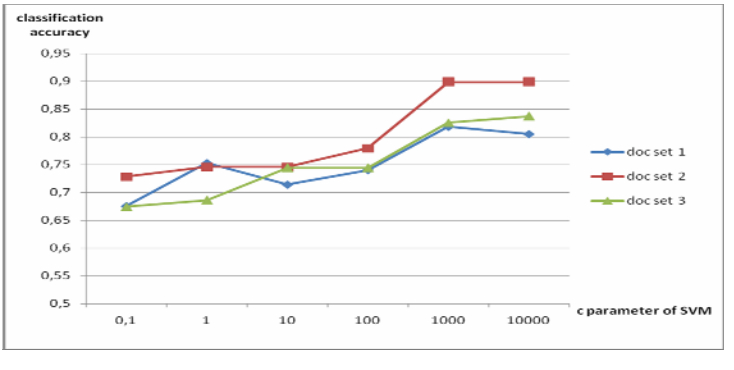

Fig. 1. Classification accuracy respect to the parameter c

Support vector machines and naïve bayes algorithms were performed on all 3 document sets which were randomly selected from 222 documents. Accuracy rates of both naïve bayes and support vector algorithms for 3 document sets are similar as shown in figures 2 and 3. Hence explaining the results of only one document set (ex doc set 2) may supply us the overall inferences for all 3 train/test pairs.

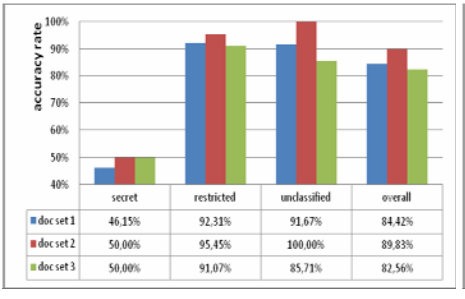

Fig. 2. SVM results for different doc. sets

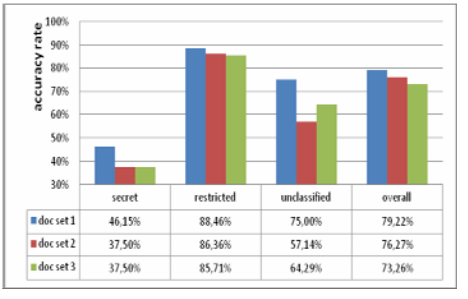

Fig. 3. NB results for different doc. sets

Performing classification algorithms on **document set 2** the tables 3, 4 and 5 summarize overall accuracy rates for SVM, naïve bayes and sub-classification based classification by using SVM respectively. Rows in tables 3, 4 and 5 represent the actual/real class labels of documents and columns refer to the given class labels by classification algorithms on document set 2. For example in table 3 we infer that document set 2 provides 8 secret documents. (4+3+1) And SVM classifier assigned 4 of them as secret, 3 as restricted and 1 as unclassified. Hence the accuracy rate for secret documents in document set 2 is 50%. (4/8)

For 59 test documents, pure security level classification results were in a satisfactory level, but may be improved as shown in table 3.

Table 3. Classification with SVM of doc set 2

Actual \ Predicted	Secret	Restricted	Unclassified	ACCR
Secret	4	3	1	50%
Restricted	0	42	2	95,5%
Unclassified	0	0	7	100%
Overall Accuracy Rate:	89,83% (53/59)			

Table 4. Classification with NB of doc set 2

Actual \ Predicted	Secret	Restricted	Unclassified	ACCR
Secret	3	4	1	37,5%
Restricted	4	38	2	86,3%
Unclassified	0	3	4	57,1%
Overall Accuracy Rate:	76,27% (45/59)			

Regarding support vector classification results, we noticed that *restricted* documents are very well classified with 2 misclassified out of 44 documents. On the other hand SVM classifier is not very effective to detect *secret* labeled documents.

Another well known classifier widely used in document classification is naïve bayes classifier. In this study we used Weka as a naïve bayes classifier tool. The

overall results obtained in naïve bayes classification are less preferable than in support vector classification as shown in table 4. Accuracy rates of naïve bayes classifier for all classes are lower than of support vector classification table.

In this study, we are classifying randomly selected test documents into 3 classes. It is a very interesting result that the misclassified document numbers of support vector classification are mostly matching with the misclassified document numbers of naïve bayes classification for all 3 document sets. For example in set 2, there are totally 6 documents which are misclassified in support vector classification. (DocIds: 1-11-12-13-14-25) 5 of these 6 documents are also misclassified in naïve bayes classification. (DocIds: 1-11-12-13-14) That means, classification errors in both two classification algorithms are nearly the same.

This study also aims to state a relation between class labels and sub-class labels implicating parent label. Sub-classification areas are other distinctive properties of documents like document type, area or format. Detecting class labels of internal documents of an organization depends on some interaction rules of sub-classes. For example, document area (military, private company, government) and document type (study report, travel report, meeting report, procedure) are sub-classes of security level classification (secret, restricted, unclassified) of TUBITAK UEKAE.

Some of rule based subclass – class interactions can be defined as:

If area: **military** and type: **spec. document** then level: **secret**
If area: **military** and type: **procedure** then level: **restricted**
If area: **government** and type: **travel** then level: **restricted**
If area: **general** and type: **tech. guide** then level: **unclassified**

All of 59 test documents are classified with support vector machines according to their area (military, private company, government, general etc.) and their type (study report, travel report, meeting report, procedure etc.) respectively. The results of two SVM sub-classifications are merged for each test document according to the subclass-class interaction rules mentioned above. Finally we obtained a success matrix as follows:

Table 5. SVM Classification results performing sub-classification logic

Actual \ Predicted	Secret	Restricted	Unclassified	ACCR
Secret	3	3	2	37,5%
Restricted	0	42	2	95,5%
Unclassified	0	1	6	85,7%
Overall Accuracy Rate:	76,27% (45/59)			

We expected that sub-classification based classification may increase security classification accuracy. Because we believe that learning the type and the area of a document is easier and much effective than learning its security level. But in this

study, for this document set accuracy level of sub-classified solution is lower than the conventional solutions.

4 Conclusion and Future Work

In this study we have classified internal Turkish documents of TUBITAK UEKAE (a military-governmental organization) using support vector machines and naïve bayes classifiers. Obviously, support vector machines are more preferable than naïve bayes classifiers for text classification. It is also noticeable that subclass-class interaction based classification is no more successful than the conventional classification methods. Finally we highlight that the classification framework suggested in this study can be used to detect and prevent loss of confidential data of organizations via web platform as a data loss prevention solution.

The document set retrieved from the internal documents of TUBITAK UEKAE obviously provides "restricted" documents many more than "secret" and "unclassified" documents because of the military-governmental nature of the organization. In the future we aim to weight the instances of underrepresented "secret" and "unclassified" instances by using re-weighting techniques.

Another issue is about the structured representations of textual data. In this study, SVM's are trained with a TF-IDF form of data. Bag of words representation of textual data will also be used to train SVM algorithms in the future.

Finally we aim to extend our classification framework by using semi-supervised classification methods with unlabeled documents and machine learning techniques by means of knowledge engineering to obtain more accurate results from sub-classification issues.

References

1. Cortes, C., Vapnik, V.: Support-vector Networks. Machine Learning 20, 273–297 (1995)
2. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398. Springer, Heidelberg (1998)
3. Feldman, R., Sanger, J.: Text Mining Handbook. Cambridge University Press, Cambridge (2007)
4. Han, J.W., Kamber, M.: Data Mining Concept and Techniques, 2nd edn. (2007)
5. Alparslan, E., Bahsi, B., Karahoca, A.: Classification of Turkish News Documents Using Support Vector Machines. INISTA (2009)
6. Cooley, R.: Classification of News Stories Using Support Vector Machines. In: IJCAI Workshop on Text Mining (1999)
7. http://svmlight.joachims.org/
8. http://code.google.com/p/zemberek/
9. Eyheramendy, S., Lewis, D., Madigan, D.: On the Naive Bayes Model for Text Categorization (2003)
10. Ageev, M., Dobrov, V.: Support Vector Machine Parameter Optimization for Text Categorization. In: International Conference on Information Systems Technology and its Applications (2003)