Extraction of Key Phrases from Biomedical Full Text with Supervised Learning Techniques

Extraction of Key-phrases from Biomedical Full-text with Supervised Learning Techniques

Yanliang Qi
Information System Department
New Jersey Institute of Technology
University Heights, Newark, NJ 07102
yq9@njit.edu

Artun I. Yagci
Information System Department
New Jersey Institute of Technology
University Heights, Newark, NJ 07102
iay2@njit.edu

Min Song
Information System Department
New Jersey Institute of Technology
University Heights, Newark, NJ 07102
min.song@njit.edu

ABSTRACT

Key-phrase extraction plays useful a role in the research area of Information Systems (IS) such as digital libraries. Short metadata like key phrases could be beneficial for searchers to understand the concepts of documents' concept. This paper evaluates the effectiveness of different supervised learning techniques on biomedical full-text: Naïve Bayes, linear regression, SVMs (reg1/2), all of which could be embedded inside an IS for document search. We use these techniques to extract key phrases from PubMed. We evaluate the performance of these systems using the well-established holdout validation method. The contributions of the paper are comparison among different classifier techniques, and a comparison of performance differences between full-text and abstract. We conducted experiments and found that SVMreg-1 improves the performance of key-phrase extraction from full-text while Naïve Bayes improves from the abstracts. These techniques should be considered for use in information system search functionality. Additional research issues also are identified.

Keywords

Key-phrase extraction, Naive Bayes, Linear Regression, SVM, classifier

INTRODUCTION

In recent years, there has been a tremendous increase in the number of biomedical documents in the digital libraries that provide users (researchers, readers) with access to the scientific and technical literature of those biomedical documents (articles or abstract). For example, the PubMed digital library (a free search engine for accessing the MEDLINE database of biomedical research articles) currently contains over 18 million citations from various types of biomedical documents published in the past several decades (www.pubmed.gov). With the rapid expansion of the number of biomedical documents, the ability to effectively determine the relevant documents from a large dataset has become increasingly difficult for users. As it is a challenging task for a reader to examine complete documents to determine whether the document would be useful, short semantic metadata like key-phrases would be an alternative for a reader to understand the concept of the document. Key phrases are increasingly used as brief descriptors of text document content. However, not all of the biomedical documents in digital libraries have key phrases, so readers have to read through the documents to determine whether they are relevant to their research.

Therefore automatically presenting key phrases from a document has become an important task in the biomedical domain.

Automatic key-phrase extraction can be defined as the process of extracting key phrases from a document that an author (or a professional indexer) is likely to assign to that document (El-Beltagy 2006). Consequently, automatic extraction makes it feasible to generate key phrases for a large number of full-text documents that do not have manually assigned key phrases. It also reduces the cost and time spent manually assigning key phrases to documents (Zhang et al. 2005). Key-phrases, short semantic metadata, are useful for various purposes including summarizing as well as search engine optimization. Using key phrases for full-text documents can vary: when they are presented on the first page of the document, the goal is summarization, which enables the users to quickly determine the concept of the document; when they are entered in a search engine query box in a digital library, the goal is to enable the users to make the search more precise (Turney 2000a). Therefore, they play an important role in document descriptions and document search in digital libraries, e.g., PubMed.

Traditionally, key-phrases are assigned manually to documents by authors or professional indexers. The indexers often choose key phrases from a predefined control vocabulary: Medical Subject Heading (MeSH). Authors usually choose key phrases to present their work in a certain way, or to maximize its chance of being noticed by particular searchers. However, issues with this manual assignment of key-phrases are (1) it is a time consuming process, and (2) it requires knowledge of subject matter and also entails an updated control vocabulary list (Kumar et al. 2008; Witten et al. 1999). Automatic key phrase extraction can be a good practical alternative.

Key-phrases can be automatically generated in two ways: (1) key-phrase assignment (controlled-vocabulary-indexing-based), which is assigning key-phrases from a controlled vocabulary to documents or (2) key-phrase extraction (free-term-indexing-based), which is identifying and selecting the most descriptive phrases in that document (Dumais et al. 1998).

In domain-specific control-indexing, key-phrases are chosen from a controlled vocabulary such as the MeSH terminology list (Medelyan et al. 2006). The MeSH provides a consistent way to assign phrases to biomedical documents that have the same concept. However the downsides are that the lists are expensive to build and maintain, so they are not always up to date, and potentially useful phrases are ignored if they are not in the list (Jones et al. 2003).

In the free-term indexing, the text of a document is analyzed and its most appropriate phrases are identified and associated with the biomedical document (Witten et al. 1999). This means that the selection of key-phrases does not depend on a controlled vocabulary, e.g., MeSH, but rather chooses key phrases of the document.

Both of the approaches use machine learning methods that are the branch of Artificial Intelligence concerned with the design of algorithms that allow machines to improve their performance over time by learning from data (Abu-Nimeh et al. 2003), and require for training purposes a set of documents with key-phrases which are already attached. Currently, several key phrase extraction techniques have been proposed based on different machine-learning techniques, e.g., KEA (Witten et al. 1999), and GenEx (Turney 2000a; Witten et al. 1999).

This paper focuses on a key-phrase extraction task from biomedical documents with supervised learning techniques, which is a machine learning technique for learning a function from training data. The tasks we discuss in this paper are to take a biomedical document as an input and automatically generate a list (in no particular order) of key-phrases as an output and compare the performance of different algorithms. We also evaluate the effectiveness of automatically extracted key-phrase in terms of how many author-assigned key-phrases are correctly identified (Pala et al. 2008; Turney 1997). For this evaluation, we use Naïve Bayes, linear regression, and SVMs (SVMreg-1 and SVMreg-2) classifiers on biomedical full-texts and the abstracts of the documents by comparing the output key-phases with author-assigned key-phrases from PubMed. We evaluate the effectiveness of extracted key-phrases in terms of how many author assigned key-phrases are correctly identified by using a holdout validation method. The primary contribution of the paper is (1) comparison among different classifier techniques, and (2) a comparison of performance differences between full-text and abstract. In brief, our results show that Naïve Bayes performs better than the other three on abstract only, while technique SVMreg-1 performs better than others on full-text. Linear regression performs worse than others in most of the cases.

The rest of this paper is structured as follows; Section 2 discusses the systems and learning algorithms for key phrase extraction. Section 3 explains the settings and data collection, Section 4 presents the evaluation experiment and results, while Section 5 discusses contributions and presents future research direction.

THE SYSTEMS AND LEARNING ALGORITHMS FOR KEY PHRASE EXTRACTION

Naturally, key-phrase extraction from a text document is considered a classification task: each phrase is classified either as a key-phrase or non key-phrase class. To the best of our knowledge, most published literature covers only supervised tasks (Huang et al. 2006; Turney 2000a; Witten et al. 1999). In machine learning terminology, the key phrase list in a text document is used as an example, and the learning problem is to find a mapping from the examples to the two classes (key-phrase and non-key-phrase) (Frank et al. 1999). In the context of key phrase extraction, these are simply phrases that have been identified as being either key-phrases or not. Once the learning method has generated the model given the training data, it can be applied to unlabeled data. The training documents are used to adjust the key-phrase extraction algorithms to attempt to maximize system performance (Turney 1997). The main difficulties arise from how the system correctly identifies whether a phrase in a document is either a key-phrase or not (Turney 1999; Turney 2000b).

In general, the key-phrase extraction process can be achieved in the following four steps (El-Beltagy 2006):

1: Extract candidate phrases and their number of occurrence: a candidate phrase is defined as any sequence of words within the input document that is not separated by punctuation marks or stop words; then the common suffixes are removed from each candidate phrase by applying a stemmer algorithm, such as the Porter or the Lovins stemmers.

2: Filter candidate phrases. In order to reduce the number of candidate phrases, a number of rules may be applied such as a filtration rule, e.g., a certain number of times a phrase occurs in a text document can be considered a candidate phrase.

3: Calculate the weight of candidate phrases: the weight is calculated to enable ranking by applying linguistic or/and statistical techniques on domain text such as TFIDF, C/NC-value. TFIDF weighting is the most common statistical method of measuring the weight of a candidate phrase in a document (Zhang et al. 2005). TFIDF value has been used in the candidate phrase extraction step by some of the well-known key phrase extraction systems such as KP-Miner, or KEA. C/NC-value is the other method for calculating the weight of candidate phrases used in the biomedical domain, introduced by Frantzi et al (Frantzi et al. 1998). The C/NC-value method combines linguistic (linguistic filter, part-of-speech tagging and stop-list) and statistical analysis (frequency analysis, C/NC-value) to enhance the common statistical techniques (e.g. TFIDF) by making sensitive to a particular type of multi-word term. C-value enhances the common statistical measure of frequency of occurrence for phrase extraction. NC-value gives a method for the extraction of phrase context words and the incorporation of information from phrase context words to the extraction of terms (Frantzi et al. 1998).

4: Refine the results and generate a final key-phrase list. Once the weight calculation step has been performed, a number of key-phrases are listed in rank order of phrases.

In recent years, more effective systems have been developed to improve the performance of a key phrase extraction by integrating data mining techniques (decision tree, Naïve Bayes, SVMs, etc). For instance, KP-Miner (El-Beltagy 2006), improves TFIDF by introducing two factors (provide higher weights for terms whose length is greater than one and for terms that appear somewhere in the beginning of the document); LAKE (D'Avanzo et al. 2004) relies on the linguistic features of a document in order to perform key-phrase extraction through the use of a Naïve Bayes classifier as the learning algorithm and TFIDF term weighting with the position of a phrase as a feature; KPSpotter (Song et al. 2003), uses the information gain measure to rank the candidate phrases based on a TFIDF and distance feature. Following are brief description of the state-of-the art key phrase extraction systems: KEA and GenEx.

Turney was the first to approach the task of key-phrase extraction as a supervised learning problem. Turney proposes a system named Extractor, using frequency-based and part-of-speech information as features, and decision tree and a generic algorithm (called Genex) as classifiers (Piramuthu et al. 2006; Piramuthua et al. 2009; Turney 1997; Turney 1999; Turney 2000b). Extractor uses a set of heuristic and generic algorithms to identify the phrases that are most likely to map to those of the author's. The Genex system has two components, the Genitor genetic algorithm and the Extractor key-phrase extraction algorithm. Extractor takes a document as an input and extracts a list of key-phrases as an output. The output of Extractor is controlled by numerical parameters.

KEA is another efficient algorithm for extracting key-phrases from text documents. KEA system runs in two stages: (a) Training and (b) Extraction. During the training stage, it creates a model for identifying candidate phrases, using a set of training documents in which the author assigned key-phrases are known. During the extracting stage, it chooses key-phrases from a new document, using the model generated training stage. Both stages have two steps: choose a set of candidate phrases from their input documents, and then calculate the values of certain features (TFIDF and fist-occurrence) for each candidate phrase. More explicitly, KEA, by default, generates key-phrases using Naïve Bayes as the classifier. KEA's Naïve Bayes learning model uses a set of training documents with known key-phrases (author assigned), and then uses the model to determine which of an input document's phrases are most likely to be candidate key-phrases. The desired number of key-phrases may be defined in the KEA algorithm. KEA chooses candidate phrases in three steps: input cleaning, phrase identification, and case-folding (Witten et al. 1999). An implementation is available from the New Zealand Digital Library project (www.nzdl.org).

As we highlighted above, key-phrase classification can be affected by various machine learning approaches of classifiers, such as Naïve Bayes, SVM, and linear regression. A large number of classification algorithms have been used for a majority of automatic key-phrase extraction systems to address key-phrase extraction problems for text documents. Each algorithm has its own strengths and drawbacks. However, Naïve Bayes, regression, and SVM are the most frequently used supervised learning techniques. Indeed, some studies show that the performances of the algorithms are compatible, and they achieve very good performance in text classification tasks (Colas et al. 2006). The following sections describe basic classification algorithms (analyze biological datasets) that we tested in this study: Naive Bayes (Ferrari et al. 2006), regression (Arshadi et al. 2005), and Support Vector Machine (Brown et al. 2000).

The Naïve Bayes classifier is popular due to its simplicity, and computational efficiency, and has been widely used for text classification (D'Avanzo et al. 2006). The algorithm uses the joint probabilities of words and categories to estimate the probabilities of categories given in a text document. It computes the posterior probability that the text document belongs to different classes and assigns it to the class with the highest posterior probability. The posterior probability of class is computed using the Bayes rule, and the testing sample is assigned to the class with the highest posterior probability. The advantage of Naïve Bayes can be explained in a way which is fast to train and fast to evaluate; the classifier can also handle missing values by ignoring the examples during model building and classification.

Linear Regression is another important classifier algorithm used to analyze biological datasets. It is from regression analysis in which the relationship between one or more independent variables and dependent variables (Arshadi et al. 2005). The goal of linear regression is to find the line that best predicts the dependent variable from independent variables. To achieve this goal, linear regression finds the line that minimizes the sum of the squares of the vertical distances of the points from the line.

The other type of learning system is Support Vector Machines (SVMs), which is relatively new machine learning process, influenced by advances in statistical learning theory. The algorithm is a linear learning system that builds two spate classes (suitable for binary classification tasks), which are generated from training examples. The overall aim is to generalize test-data well. Utilized as binary categorical classifiers, the SVM method performs classification more accurately than most other supervised learning algorithms in biological data analysis applications, especially those applications involving high dimensional datasets (Brown et al. 2000; Liu 2007). The SVM method uses kernel function (the similarity function, which is computed in the original attribute space). The Support Vector Machine can be considered complex, slow and takes a lot of memory (limitation of the kernel), but is a very effective classifier in a wide range of bioinformatic problems, and in particular performs well in analyzing biological datasets (Brown et al. 2000).

As we highlighted earlier, estimating classifier accuracy is one of the most important criteria for evaluating the systems. Classifier accuracy can be measured by using several evaluation methods: holdout, k-fold cross validation, leave-one-out-cross-validation, bootstrapping, and counting the cost (Jones et al. 2003; Kohavi 1995). Although a debate continues in machine learning and text mining circles about what is the best method for evaluation, holdout validation has been widely used schema in practice (Witten et al. 1999; Zhang et al. 2005). The holdout method is the simplest kind of cross validation. In this method, the data set is separated into two sets: training and testing. A certain amount of data is held over testing (also called hold-out set), and the remaining data set is used for training. In this study, we also tested classifiers accuracy using holdout validation method.

In this paper, we also approach the key phrase extraction problem as a classification task using supervised learning. We aim to conduct a benchmark study in order to investigate whether there is a significant difference between the various machine learning approaches to automated key phrase extraction from full text documents and abstract of the documents. First, we compare the classier approaches using full text document. Second, we compare the classifiers' approaches using the abstracts of documents. Finally, we compare the classifier performance based on abstract vs. full text.

EXPERIMENTAL SETTING AND DATA COLLECTION

This section reviews the setting and data collection process of our experimental evaluation. The purpose of this experiment was to assess the differences between classifier techniques (Naïve Bayes, linear regression, and SVMreg-1, and SVMreg-2) on biomedical full-text documents and the abstracts of these documents. SVMreg implements the support vector machine for regression; more information see (Shevade et al. 2000; Smola et al. 2003). We used two options of SVMreg: option 1 standardizes the data source while option 2 does not. We compare the performance of each classifier using the KEA system. We chose to use KEA for numerous reasons: this system evaluation can be carried out automatically; it allows the modifying of parameters such as CutOff points and the number of key-phrases extracted, e.g., 5, 10, and 15 key-phrases per document; it performs at the current state of the art (Frank et al. 1999; Witten et al. 1999).

We measure the performance of the classifiers by counting the number of matches between the output of the four systems and the key-phrases that were originally assigned by the author. We used this measure instead of traditional information retrieval metrics (recall and precision) for three reasons (Witten et al. 1999). First, a single overall value is more easily interpreted than two values. Second, the common information retrieval metrics of precision and recall can be misleading, for it is easy to increase precision at the expense of system's recall or increase recall at the expense of systems' precision. Third, this measure fits reasonably well into the expected behavior of end-users who are likely to ask for certain numbers of key-phrases for a text document.

Our experiment is divided into three parts: (1) we perform Naïve Bayes, linear regression classifiers, and SVMs on biomedical full-text documents; (2) we perform the four classifiers using the abstracts rather than full-text, when extracting 5, 10, and 15 key-phrases; (3) we consider whether the performance of the classifiers suffers when it only uses abstracts to extract key-phrases.

We measure the performance of the classifiers by comparing key-phrases with author-assigned key-phrases (Chen 1999; El-Beltagy 2006). To evaluate the performance of the four techniques, we use the holdout procedure in which the document collection is split into two sets, where the first one serves for training and the second for testing purposes. The training set is analyzed to adjust the model to the characteristics of the data. The test set serves solely for testing purposes. Our data set consists of 1002 full text documents, and is collected from PubMed www.pubmedcentral.nih.gov. From those 1002 full text documents, we picked the first 50 and the last 50 documents to build up our training data set, which has 100 documents. The others are used as a test set. For each document, there is a set of key-phrases, assigned by the authors of the articles (or professional indexers). We extract key-phrases and the abstract of each document and check them one by one. We have made a comparative study of all techniques. A sample system output is given in Table 1 which contains 5 key-phrases extracted by the four techniques.

<table>
<tr><td colspan="5">Document Title: A Cdt1–geminin complex licenses chromatin for DNA replication and prevents rereplication during S phase in Xenopus
PubMed ID: PMC7601436</td></tr>
<tr><td>Author assigned key-phrases</td><td>Naïve Bayes</td><td>Linear Regression</td><td>SVMreg-1</td><td>SVMreg-2</td></tr>
<tr><td>Cdt1
DNA replication
licensing
geminin
prereplication complex
rereplication</td><td>geminin
Cdt1
chromatin
licenses
al</td><td>Genetics
synthesis
prevents
Xenopus
Institute</td><td>Lutzmann
sufficient
prevents
Xenopus
activity</td><td>geminin
Cdt1
chromatin
licenses
complex</td></tr>
</table>

Table 1: Five extracted key-phrases by all four systems

EXPERIMENT AND RESULTS

Each of the performances of the four algorithms is tested. Table 2, Table 3, and Table 4 show the results of the performance of the Naïve-Bayes, linear regression, SVMreg-1, and SVMreg-2 classifier on full text documents and abstracts based on the holdout evaluation method. AVG denotes the mean of the number of subjects, and STDEV denotes the standard deviation.

Table 2, first experiment, represents the average number of correct matches with full-text documents by the four techniques. The comparative study results show that (1) SVMreg-1 classifier outperforms Naïve Bayes, Linear Regression and SVMreg-2 when we use the full text documents to extract 5, 10, and 15 key-phrases; (2) Linear regression, the second best performer technique, tends to better perform than Naïve Bayes and SVMreg-2 when we use the abstracts to extract 5, 10, 15 key-phrases; (3) It is acknowledged that the SVMreg-2 performs worse than others in most of the cases. SVMreg-1 performs better than Linear regression in 5, 10, 15 key-phrases by 2.04%,2.13%, and 6.02% respectively. SVMreg-2 was the worst performer compared to the best performer by 6.38%, 23.08%, and 40.80% respectively.

. Algorithm	5 Key phrases	10 Key-phrases	15 Key-phrases
	AVG/STDEV	AVG/STDEV	AVG/STDEV
Naïve Bayes	0.95 +/- 0.91	1.31 +/- 1.09	1.57 +/- 1.19
Linear Regression	0.98 +/- 0.97	1.41 +/- 1.19	1.66 +/- 1.30
SVMreg-1	1.00 +/- 0.94	1.44 +/- 1.16	1.76 +/- 1.28
SVMreg-2	0.94 +/- 0.99	1.17 +/- 1.14	1.25 +/- 1.21
The best vs the second	**2.04%**	**2.13%**	**6.02%**
The best vs the worst	**6.38%**	**23.08%**	**40.80%**

Table 2: Performance of the classifiers using full text

In the second experiment, our data set consists of 1002 abstracts of the same documents, which were extracted from the full text documents used in the first experiment. In Table-3, the comparative study results show that (1) Naïve Bayes outperforms SVMreg-1, SVMreg-2 and linear regression classifier when we use the abstracts only to extract 5, 10, 15 key-phrases. For example, Naïve Bayes can match on average between one and two of key-phrases of the 5, 10, 15 key-phrases assigned by the author. This difference is statistically significant. (2) The second best performer technique, SVMreg-2, tends to perform better than linear and SVMreg-1 when we use the abstracts to extract 5, 10, 15 key-phrases. (3) Linear regression performs worse than others in most of the cases. Naïve Bayes performs better than SVMreg-2 in 5, 10, 15 key-phrases respectively by 47.27, 35.80%, and 34.07%. Linear regression was the worst performer compared to the best performer in 5, 10 key-phrases by 161.29%, 59.42% respectively. Also when it comes to 15 key-phrases, SVMreg-1 is the worst performer with 50.62%.

Algorithm	5 Key phrases	10 Key-phrases	15 Key-phrases
	AVG/STDEV	AVG/STDEV	AVG/STDEV
Naïve Bayes	0.81 +/- 0.84	1.10 +/- 1.02	1.22 +/- 1.07
Linear Regression	0.31 +/- 0.56	0.69 +/- 0.84	0.88 +/- 0.92
SVMreg-1	0.38 +/- 0.61	0.69 +/- 0.81	0.81 +/- 0.90
SVMreg-2	0.55 +/- 0.73	0.81 +/- 0.88	0.91 +/- 0.93
The best vs the second	**47.27%**	**35.80%**	**34.07%**
The best vs the worst	**161.29%**	**59.42%**	**50.62%**

Table 3: Performance of the classifiers using abstracts

Our final analysis checks whether the classifiers perform better when it only uses the full-text documents. We find significant performance differences between full-text and abstracts of the documents. Table 4 represents the number of correct matches with full-text documents and abstracts of these documents. The results show that all of the four classifier techniques extract more matching key-phrases from full-text compared to the abstracts of the documents. One can argue that when using abstracts, the reason for reduced performance is more likely to retrieve a smaller number of key-phrases found in the abstract than in the biomedical full-text document (Witten et al. 1999).

	Number of key-phrases	Naïve Bayes	Linear Regression	SVMreg-1	SVMreg-2
Full Text	5	0.95 +/- 0.91	0.98 +/- 0.97	1.00 +/- 0.94	0.94 +/- 0.99
	10	1.31 +/- 1.09	1.41 +/- 1.19	1.44 +/- 1.16	1.17 +/- 1.14
	15	1.57 +/- 1.19	1.66 +/- 1.30	1.76 +/- 1.28	1.25 +/- 1.21
Abstract	5	0.81 +/- 0.84	0.31 +/- 0.56	0.38 +/- 0.61	0.55 +/- 0.73
	10	1.10 +/- 1.02	0.69 +/- 0.84	0.69 +/- 0.81	0.81 +/- 0.88
	15	1.22 +/- 1.07	0.88 +/- 0.92	0.81 +/- 0.90	0.91 +/- 0.93

Table 4: Effect of Full text and Abstract

Our result shows that the classifiers can match on average between one and two of the key-phrases assigned by the author in the documents. However, classifiers' incorrect key-phrase choices are not necessarily poor key-phrases for several reasons. Authors do not necessarily choose key-phrases that best describe the content of their paper. Another reason is that authors might choose key-phrases to slant their work in a certain way, or to maximize its chance of being noticed by particular searchers (Witten et al. 1999). The limitations of this study come from two parts. One is the size of data set, in the future, we will add more documents (may expand to 10k level) into our data set. The other one is from evaluation method. We will use 10-fold cross validation method instead of holdout method in next step.

CONCLUSION

Automatic key-phrase extraction is important because it makes it feasible to generate key-phrases for a large number of biomedical documents that do not have manually assigned key-phrases. It reduces the cost and time spent in manually assigning key phrases to documents. Naturally, key-phrase extraction from a text document is considered a classification activity using supervised learning method. In this paper, (1) we compare the performance among different classifier methods (Naïve Bayes, Linear Regression, and SVMreg-1, and SVMreg-2) using the KEA system; and (2) we compare performance differences between biomedical full-text documents and abstracts of these documents only. Based on the experimental study, we find that the SVMreg-1 classifier improves the performance of key-phrase extraction from biomedical full-text documents when extracting 5, 10, 15 key-phrases, while Naïve Bayes classifier improves the performance of key-phrase extraction from the abstracts of documents when extracting 5, 10, and 15 key-phrases. As a future work, we plan to recruit the subject to evaluate the extracted results by the system and measure the perceived accuracy of results by the system without using original author's (or professional indexers) choices; this could be an interesting contribution to this study.

ACKNOWLEDGMENTS

Partial support for this research was provided by the National Science Foundation under grants DUE-0434581 and DUE-0434998, by the Institute for Museum and Library Services under grant LG-02-04-0002-04, and by the New Jersey Institute of Technology.

REFERENCES

1. Abu-Nimeh, S., Nappa, D., Wang, X., and Nair, S. "An empirical comparison of supervised machine learning techniques in bioinformatics," in: *Research and Practice in Information Technology Series*, 2003, pp. 219-222.
2. Arshadi, N., and Jurisica, I. "Feature Selection for Improving Case Based Classifiers on High Dimentional Data Sets," AAAI, 2005.
3. Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares, M., and Haussler, D. "Knowledge-based analysis of microarray gene expression data by using support vector machines," *The National Academy of Sciences* (97:1) 2000, pp 262-267.
4. Chen, K. "Automatic Identification of Subjects for Textual Documents in Digital Libraries," 1999.
5. Colas, F., and Brazdil, P. "Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks," International Federation for Information Processing, Santiago, Chile, 2006, pp. 169-178.
6. D'Avanzo, E., Frixione, M., and Kuflik, T. "LAKE system at DUC-2006," 2006.
7. D'Avanzo, E., Magnini, B., and Vallin, A. "Keyphrase Extraction for Summarization Purposes:The LAKE System at DUC-2004," in: *Document Understanding Workshop*,, HLT/NAACL, Boston, USA, 2004.
8. Dumais, S.T., Platt, J., Heckerman., D., and Sahami, M. "Inductive Learning Algorithms and Representations for Text Categorization," Information and Knowledge Management, 7th International CNF, ACM Press, 1998, pp. 148-155.
9. El-Beltagy, S. "KP-Miner: A Simple System for Effective Keyphrase Extraction," Innovation in Information technology, IEEE Xplore, 2006, pp. 1-5.
10. Ferrari, L.D., and Aitken, S. "Mining housekeeping genes with a Naive Bayes classifier," *BMC Genomics*) 2006.
11. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., and Nevill-Manning, C.G. "Domain-specific keyphrase extraction," 6th International Joint Conference on Artificial Intelligence (IJCAI-99), 1999, pp. 668-673.
12. Frantzi, K.T., Ananiadou, S., and Tsujii, J.-i. "The C-value/NC-value Method of Automatic Recognition for Multi-Word Terms," The Second European Conference on Research and Advanced Technology for Digital Libraries T, Springer-Verlag, London UK, 1998, pp. 585 - 604.
13. Huang, C., Tian, Y., Zhou, Z., Ling, C.X., and Huang, T. "Keyphrase Extraction using Semantic Networks Structure Analysis ", 2006.
14. Jones, S., and Paynter, G.W. "An Evaluation of Document Keyphrase Sets," *Journal of Digital Information* (4:1) 2003.
15. Kohavi, R. "A study of cross-validation and bootstrap for accuracy estimation and model selection ", 1995.
16. Kumar, N., and Srinathan, K. "Automatic keyphrase extraction from scientific documents using N-gram filtration technique," the 8th ACM symposium on Document engineering ACM SIGDOG, Sao Paulo, Brazil, 2008, pp. 199-208.
17. Liu, B. *Web Data Mining* Springer, 2007, pp. 56-116.
18. Medelyan, O., and Witten, I.H. "Thesaurus Based Automatic Keyphrase Indexing," in: *JCDL*, ACM, 2006.
19. Pala, N., and Cicekli, I. "Turkish keyphrase extraction using KEA," Computer and Information Science, IEEE Xplore, 2008, pp. 1-5.
20. Piramuthu, S., and Sikora, R.T. "Genetic Algorithm Based Learning Using Feature Construction," in: *INFORMS Annual Meeting*, INFORMS, Pittsburgh, PA, 2006.
21. Piramuthua, S., and Sikora, R.T. "Iterative feature construction for improving inductive learning algorithms," *Expert Systems with Applications* (36:2) 2009, pp 3401-3406.
22. Shevade, S.K., Keerthi, S.S., Bhattacharyya, C., and Murthy, K.R.K. "Improvements to the SMO Algorithm for SVM Regression," *IEEE TRANSACTIONS ON NEURAL NETWORKS* (11:5) 2000, pp 1188-1193.
23. Smola, A.J., and Scholkopf, B. "A Tutorial on Support Vector Regression," 2003.
24. Song, M., Song, I.-Y., and Hu, X. "KPSpotter: a flexible information gain-based keyphrase extraction system," International workshop on Web information and data management ACM, 2003, pp. 50-53.

25. Turney, P. "Extraction of key Phrases from Text: Evaluation of Four Algorithms," National Research Council Canada.
26. Turney, P. "Learning algorithms for keyphrase extraction," *Journal of Digital Information*) 2000a.
27. Turney, P.D. "Learning to Extract Keyphrases from Text," in: *Technical Report ERB-1057 (NRC #41622),* Canadian National Research Council, Institute for Information Technology, 1999.
28. Turney, P.D. "Learning Algorithms for Keyphrase Extraction," *Information Retrieval* (2:4) 2000b, pp 303-336.
29. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., and Nevill-Manning, C.G. "KEA: Practical Automatic Keyphrase Extraction," The Fourth on Digital Libraries '99, ACM CNF, 1999, pp. 254-255.
30. Zhang, Y., Zincir-Heywood, N., and Milios, E. "Narrative text classification for automatic key phrase extraction in web document corpora," 7th annual ACM international workshop on Web information and data management ACM SIGIR, Bremen, Germany, 2005, pp. 51-58.