Automatic Text Classification: A Technical Review

Mita K. Dalal
Sarvajanik College of Engineering & Technology,
Surat, India

Mukesh A. Zaveri
Sardar Vallabhbhai National Institute of Technology,
Surat, India

ABSTRACT
Automatic Text Classification is a semi-supervised machine learning task that automatically assigns a given document to a set of pre-defined categories based on its textual content and extracted features. Automatic Text Classification has important applications in content management, contextual search, opinion mining, product review analysis, spam filtering and text sentiment mining. This paper explains the generic strategy for automatic text classification and surveys existing solutions to major issues such as dealing with unstructured text, handling large number of attributes and selecting a machine learning technique appropriate to the text-classification application.

General Terms
Artificial Intelligence, Machine Learning, Mining

Keywords
automatic text classification, feature-extraction, pre-processing, text mining

1. INTRODUCTION
Automatic Text Classification involves assigning a text document to a set of pre-defined classes automatically, using a machine learning technique. The classification is usually done on the basis of significant words or *features* extracted from the text document. Since the classes are pre-defined it is a supervised machine learning task. Most of the official communication and documentation maintained in commercial and governmental organizations is in the form of textual electronic documents and e-mails. Much of the personal and other communication done by private individuals is in the form of e-mails, blogs etc. Due to this information overload, efficient classification and retrieval of relevant content has gained significant importance.

This paper explains the generic strategy for automatic text classification which includes steps such as pre-processing (eliminating stop-words [1] [2] [3], stemming [2] [4] etc.), feature selection using various statistical or semantic approaches, and modeling using appropriate machine learning techniques (Naïve Bayes, Decision Tree, Neural Network, Support Vector Machines, Hybrid techniques).

This paper also discusses some of the major issues involved in automatic text classification such as dealing with unstructured text, handling large number of attributes, examining success of purely statistical pre-processing techniques for text classification v/s semantic and natural language processing based techniques, dealing with missing metadata and choice of a suitable machine learning technique for training a text classifier.

Automatic text classification has several useful applications such as classifying text documents in electronic format [1] [5]; spam filtering; improving search results of search engines; opinion detection [6] and opinion mining from online reviews of products [7], movies [8] or political situations [9]; and text sentiment mining [9] [10] [11]. Blogging has become a popular means of communication over the Internet. New abbreviations, slang terms etc. are added on a daily basis on blogs, which are in turn quickly accepted by the blog users. In order to implement text classification applications like opinion mining or sentiment classification, it is required to keep track of such newly emerging terms (not found in standard language dictionaries). The nature of blog entries is such that additional content is added on a daily basis. Moreover, text posts on a blog do not strictly adhere to the blog topic. This introduces the need to develop incremental and multi-topic text classification techniques. There is also the need to develop automated, sophisticated text classification and summarization tools for many regional languages as several blogs and newspaper sites in these languages have become popular.

The remainder of the paper is organized as follows. Section 2 of the paper explains the generic strategy for text classification. Section 3 discusses the major issues in text classification and surveys existing solutions. Finally, Section 4 concludes the paper and provides pointer to future work in this field.

2. GENERIC STRATEGY FOR CLASSIFYING A TEXT DOCUMENT
The generic strategy for text classification is depicted in Fig 1. The main steps involved are i) document pre-processing, ii) feature extraction / selection, iii) model selection, iv) training and testing the classifier.

Data pre-processing reduces the size of the input text documents significantly. It involves activities like sentence boundary determination [2], natural language specific stop-word elimination [1] [2] [3] and stemming [2] [4]. Stop-words are functional words which occur frequently in the language of the text (for example, ‘a’, ’the’, ’an’, ’of’ etc. in English language), so that they are not useful for classification. Stemming is the action of reducing words to their root or base form. For English language, the Porter’s stemmer is a popular algorithm [4] [12], which is a suffix stripping sequence of systematic steps for stemming an English word, reducing the vocabulary of the training text by approximately one-third of its original size [4]. For example, using the Porter’s stemmer, the English word “generalizations” would subsequently be stemmed as “generalizations → generalization → generalize → general → gener”. In cases where the source documents are web pages,

additional pre-processing is required to remove / modify HTML and other script tags [13].

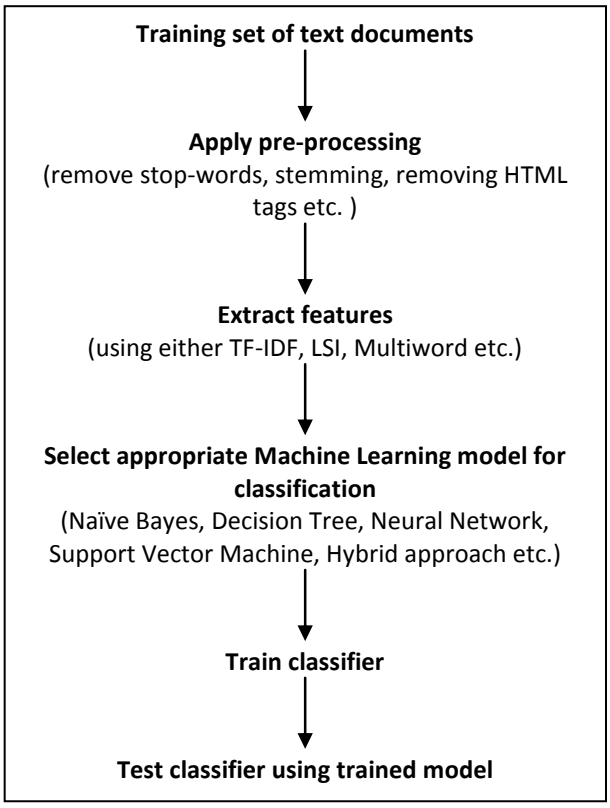

Fig 1: Generic strategy for text classification

Feature extraction / selection helps identify important words in a text document. This is done using methods like TF-IDF (term frequency-inverse document frequency) [14], LSI (latent semantic indexing) [15], multi-word [2][16] etc. In the context of text classification, features or attributes usually mean significant words, multi-words or frequently occurring phrases indicative of the text category.

After feature selection, the text document is represented as a document vector, and an appropriate machine learning algorithm is used to train the text classifier. The trained classifier is tested using a test set of text documents. If the classification accuracy of the trained classifier is found to be acceptable for the test set, then this model is used to classify new instances of text documents.

3. AUTOMATIC TEXT CLASSIFICATION

Automatic text classification is a widely researched topic due to its practical applicability to several areas of text mining. The various issues in text classification and currently available solutions are discussed next.

3.1 Automatic Text Classification: Issues and Solutions

This section discusses some important issues related to automatic text classification such dealing with unstructured text, feature selection for handling large number of attributes, retrieving metadata for classification and choice of machine learning technique for text classification.

3.1.1. Classifying unstructured text

Some types of text documents like scientific research papers are usually written strictly in a pre-specified format, which makes it easier to classify them, because of positional information of attributes. However, most text documents are written in an unstructured manner, so classification has to be done on the basis of attributes such as presence or absence of keywords and their frequency of occurrence. Text documents can be represented as document vectors using models such as the i)Multivariate Bernoulli Model [1] [17] in which the document vector is a binary vector simply indicating the absence or presence of feature terms; or the ii)Multinomial Model [1] [17] in which document vectors additionally retain the information regarding frequency of occurrence of feature terms.

3.1.2. Handling large number of attributes: Feature selection using statistical and semantic preprocessing techniques

Features useful in text classification are simple words from the language vocabulary, user-specified or extracted keywords, multi-words or metadata. In text classification literature, the steps involved in feature reduction are mainly applying pre-processing such as stop-word removal [1] [2] [3], stemming [4] etc. Text documents generally use words from a large vocabulary, but all words occurring in a document are not useful for classification. So, researchers have proposed feature reduction techniques like TF-IDF [14] [18] [19], LSI [5] [15] [18], multi-word [2] [16] etc. or a combination of such techniques. The TF-IDF [14] is a purely statistical technique to evaluate the importance of a word based on its frequency of occurrence in the document and in its relevant corpus. The LSI and multi-word techniques are semantics-oriented techniques which also attempt to overcome the two basic problems in classification 'polysemy' (one word having many distinct meanings) and 'synonymy' (different words having same meaning). The LSI technique basically tries to use the semantics in a document structure using SVD (Singular Value Decomposition) matrix manipulations. A multi-word is a sequence of consecutive words having a semantic meaning (for example, "Information Technology", "Delhi Public School", "Computer Engineering Department", "State Bank of India"). Multi-words are useful in classification as well as disambiguation. Several methods can be used to extract multi-words from text such as the frequency approach [2], mutual information approach [16] etc.

3.1.3. Retrieving metadata useful for classification

Information about metadata is useful in classification. Metadata useful in classification are keywords, proper nouns such as names of persons / places, document title, name of document author [13] etc. Web documents optionally maintain metadata using the "META" tags which is very useful in classification. Metadata such as keywords are often given by users during search. A method for retrieving features (spatial and contextual) and extracting metadata using decision tree model has been proposed in [13].

3.1.4. Modeling: Selection of appropriate machine learning technique for classification of text documents

Various supervised machine learning techniques have been proposed in literature for the automatic classification of text documents such as Naïve Bayes [1] [17], Neural Networks [20], SVM (Support Vector Machine) [22] [23] [24], Decision Tree and also by combining approaches [12] [21] [25].

No single method is found to be superior to all others for all types of classification. The Naïve Bayesian classifier is based on the assumption of conditional independence among attributes. It gives a probabilistic classification of a text document provided there are a sufficient number of training instances of each category. Since the Naïve Bayesian approach is purely statistical its implementation is straightforward and learning time is less, however, its performance is not good for categories defined with very few attributes/ features. SVM is found to be very effective for 2-class classification problems (for example, text document belongs/ not belongs to a particular category; opinion is classified as positive/negative) but it is difficult to extend to multi-class classification. A class-incremental SVM classification approach has been proposed in [26]. A Decision Tree can be generated using algorithms like ID3 [27] or C4.5 [13] [28]. Unlike Naïve Bayesian classification, Decision Tree classification does not assume independence among its features. In a Decision Tree representation the relationship between attributes is stored as links. Decision tree can be used as a text classifier when there are relatively fewer number of attributes to consider, however it becomes difficult to manage for large number of attributes.

Researchers have reported improved classification accuracy by combining machine learning methods. In [12], the performance of Neural Network based text classification was improved by assigning the probabilities derived from Naïve Bayesian method as initial weights. In [21], Naïve Bayesian method was used as a pre-processor for dimensionality reduction followed by the SVM method for text classification. There is a need to experiment with more such hybrid techniques in order to derive the maximum benefits from machine learning algorithms and to achieve better classification results.

4. CONCLUSION

Due to an upsurge in the number of blogs, websites and electronic storage of textual data, the commercial importance of automatic text classification applications has increased and much research is currently focused in this area. Text classification can be automated successfully using machine learning techniques, however pre-processing and feature selection steps play a crucial role in the size and quality of training input given to the classifier, which in turn affects the classifier accuracy.

Sophisticated text classifiers are not yet available for several regional languages, which if developed would be useful for several governmental and commercial projects. Incremental text classification, multi-topic text classification, discovering the presence and contextual use of newly evolving terms on blogs etc. are some of the areas where future research in automatic text classification can be directed.

5. REFERENCES

[1] Kim S., Han K., Rim H., and Myaeng S. H. 2006. Some effective techniques for naïve bayes text classification. IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 11, pp. 1457-1466.

[2] Zhang W., Yoshida T., and Tang X. 2007. Text classification using multi-word features. In proceedings of the IEEE international conference on Systems, Man and Cybernetics, pp. 3519 – 3524.

[3] Hao Lili., and Hao Lizhu. 2008. Automatic identification of stopwords in Chinese text classification. In proceedings of the IEEE international conference on Computer Science and Software Engineering, pp. 718 – 722.

[4] Porter M. F. 1980. An algorithm for suffix stripping. Program, 14 (3), pp. 130-137.

[5] Liu T., Chen Z., Zhang B., Ma W., and Wu G. 2004. Improving text classification using local latent semantic indexing. In proceedings of the 4th IEEE international conference on Data Mining, pp. 162-169.

[6] M. M. Saad Missen, and M. Boughanem. 2009. Using WordNet's semantic relations for opinion detection in blogs. ECIR 2009, LNCS 5478, pp. 729-733, Springer-Verlag Berlin Heidelberg.

[7] Balahur A., and Montoyo A.. 2008. A feature dependent method for opinion mining and classification. In proceedings of the IEEE international conference on Natural Language Processing and Knowledge Engineering, pp. 1-7.

[8] Zhao L., and Li C.. 2009. Ontology based opinion mining for movie reviews. KSEM 2009, LNAI 5914, pp. 204-214, Springer-Verlag Berlin Heidelberg.

[9] Durant K. T., Smith M. D. 2006. Predicting the political sentiment of web log posts using supervised machine learning techniques coupled with feature selection,. WebKDD 2006, LNAI 4811, pp. 187-206, Springer-Verlag Berlin Heidelberg.

[10] Polpinij J., and Ghose A. K. 2008. An ontology-based sentiment classification methodology for online consumer reviews. In proceedings of the IEEE international conference on Web Intelligence and Intelligent Agent Technology, pp. 518-524.

[11] Ng V., Dasgupta S., and S. M. Niaz Arifin. 2006. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In proceedings of the 21st international conference on Computational Linguistics and 44th annual meeting of the Association for Computational Linguistics, pp. 611-618.

[12] Goyal R. D. 2007. Knowledge based neural network for text classification. In proceedings of the IEEE international conference on Granular Computing, pp. 542 – 547.

[13] Changuel S., Labroche N., and Bouchon-Meunier B. 2009. Automatic web pages author extraction. LNAI 5822, pp. 300-311, Springer-Verlag Berlin Heidelberg.

[14] Jones K. S. 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, Vol. 28, No. 1, pp. 11-21.

[15] Deerwester S., Dumais S. T., Landauer T. K., Furnas G. W., and Harshman R.. 1990. Indexing by Latent Semantic Analysis. Journal of American Society of Information Science, 41(6), pp. 391-407.

[16] Church K. W., and Hanks P. 1990. Word association norms, mutual information and lexicography. Computational Linguistics, Vol. 16, No. 1, pp. 22-29.

[17] Meena M. J., and Chandran K. R. 2009. Naïve bayes text classification with positive features selected by statistical method. In proceedings of the IEEE international conference on Advanced Computing, pp. 28 – 33.

[18] Zhang W., Yoshida T., and Tang X.. 2008. TF-IDF, LSI and Multi-word in information retrieval and text categorization. In proceedings of the IEEE international conference on Systems, Man and Cybernetics, pp. 108 – 113.

[19] Jones K. S. 2004. IDF term weighting and IR research lessons. Journal of Documentation, Vol. 60, No. 5, pp. 521-523.

[20] Wang Z., He Y., and Jiang M.. 2006. A comparison among three neural networks for text classification. In proceedings of the IEEE 8th international conference on Signal Processing.

[21] Isa D., Lee L. H., Kallimani V. P., and RajKumar R. 2008. Text document pre-processing with the Bayes formula for classification using the support vector machine. IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 9, pp. 1264 – 1272.

[22] Rujiang B., and Junhua L.. 2009. A novel conception based text classification method. In proceedings of the IEEE international e-conference on Advanced Science and Technology, pp. 30 – 34.

[23] Wang Z., Sun X., Zhang D., Li X. 2006. An optimal SVM-based text classification algorithm. In proceedings of the 5th IEEE international conference on Machine Learning and Cybernetics, pp. 1378 – 1381.

[24] Zhang M., and Zhang D.. 2008. Trained SVMs based rules extraction method for text classification. In proceedings of the IEEE international symposium on IT in medicine and Education, pp. 16 – 19.

[25] Yuan P., Chen Y., Jin H., and Huang L. 2008. MSVM-kNN : Combining SVM and k-NN for multi-class text classification. IEEE international workshop on Semantic Computing and Systems, pp. 133 – 140.

[26] Zhang B., Su J., and Xu X. 2006. A class-incremental learning method for multi-class support vector machines in text classification. In proceedings of the 5th IEEE international conference on Machine Learning and Cybernetics, pp. 2581 – 2585.

[27] Quinlan J. R. 1986. Induction of Decision Trees. Machine Learning, pp. 81-106.

[28] Quinlan J. R. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.