IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 11, November 2015.
www.ijiset.com
ISSN 2348 – 7968

State Machine Based Approach to Text Classification

Narayanan Srinivasan
CSE Department, Velammal Engineering College,
Chennai, Tamil Nadu, India

Abstract
Text Classification problem was tackled in many ways. Solution to the text classification problem is usually viewed as an objective function whose value can be maximized. Thus, there is no absolute solution for the problem. Most of the solutions are semi-automated, requires little expert intervention. The solution space for the text classification problem includes set of words which best describe the category under training. Arriving at these set of words with minimal overlapping with other categories is a challenging task. In this paper, a solution to text classifier design is proposed with minimal math concepts. The solution uses the classical Theory of Computation concept, a Turing machine. The classifier is evolving as the training documents are processed using feedback from the previous learning.
Keywords: *Objective function, minimal maths, Turing Machine, Classifier*

1. Introduction

Text Classification problem is around with us almost over two decades [1][12]. As defined, given an input text document and predefined set of categories, the software system should be able to decide the correct category it belongs to, is what a text classification. Well proven methods and algorithms are also available to solve this problem as per the literature. Machine learning based solutions are predominant among the different approaches. Freely downloadable tools are also available. However, they require training of the human themselves to understand and use the classification system. Most of the tools expect the input in certain format which is having some characteristics described by mathematical functions or intermediary notations which again requires the trained human expertise.

2. Challenges and limitations of the existing solutions

Text classification will be as simple as we think, if there are very small numbers of categories. However, the situation which actually uses the text classification system usually involves sufficiently larger categories. Hence, the classification accuracy decreases. Also, it poses a complexity in generalizing the algorithm.

If we closely observe the solutions available in the literature and its practical usage in real life scenario, we would arrived a conclusion that the text classification system requires human intervention ranging from trivial to non-trivial in correct decision making of the category for a text under consideration. [15]. Usually the trivial intervention is achieved by means of hierarchical classification[14] , combining multiple classifiers [3], voting, bagging etc,. In a common human intervention scenario, to run a classifier first and to accept all its high confidence decisions, but to put low confidence decisions in a queue for later manual review.

3. Trivial solution for the Text Classification Problem

Is there any simplest solution possible for text classification? Let me analyse the possibility. In most of the practical situations text classification is used in conjunction with mining and other applications. So, complex processes are still need to be done to achieve the goal under context. Meanwhile, the result of the text classification influences the result of the mining like applications which subsequently uses the outcome of classification process. If and all there is an agreement (i) category of the document is , even it may be a multimedia file, prefixed explicitly by the originator of the document in the file name itself mandatorily which is to be feed to the classification system. (ii) category of the document is made as one file attribute by the editor application and whose value will be obtained from the user by listing category labels after analyzing the document terms frequency except stop words.

Option (ii) might seem feasible as the originator of the text document is usually at least non-naive user. This eliminates training and testing time, classification pitfalls and huge cost in maintaining the software. But, downsides are there. We cannot rely on the user's decision as we cannot fully rely on the text classifier when the number of categories keeps going to infinity. Moreover, at present web page content mining also requires classification, for example, classification of a tweet in twitter [14], which could not come with user affixed category label, can be considered as non-suitable solution. In other cases, one could argue that it is the responsibility of the user to decide the category he is intending to convey as one decides the correct suggestion wrongly spelled word in a document.

ISSN 2348 – 7968

4. Automation versus Expert need

Having analyzed the trivial solutions, any responsible engineer will take the problem as one's own and trying to find the solution. These kinds of solutions are available in the literature, like SVM [10], Neural Networks [13], Naive Baysian Classifier [8][9], Genetic Algorithms based Classifier [17], kNN classifier [11] etc,. Tools based on these solutions are also available for example SVMlight, Spider etc,. They are all more or less difficult to use, requires understanding of internals, file format and content length dependent. Hence, newer solution which effectively hides the internals from the end user and provide different file format is to be developed. Having this in mind, a new approach for text classification is proposed at the architectural level.

5. Architectural diagram for the proposed solution

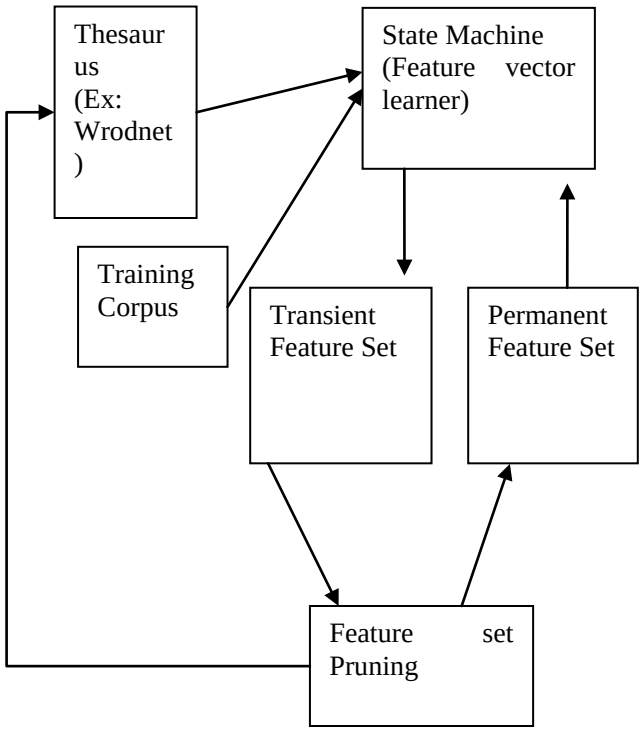

Fig.1 Architectural elements of State Machine based Classifier development

6. Functionalities of the Architectural elements

State Machine: State Machine has the well defined states while reading the input text document. Though it is given the name state machine, conceptually it is equivalent to a Turing Machine. The state machine read one symbol at a time and remains in the same state until it encounters a blank space, a delimiter symbol. When a delimiter appears, it tokenize the characters read so far using Thesaurus. If it is a stop word it is removed from the buffer, if it can be stemmed the current word is replaced by its root word. Increment the count associated with the word. It continues reading next token. Once it encounter full stop it restarts. Before restarting, it writes the attribute set (Feature set) in to the transient feature set buffer. This process repeats for all the lines in the text. The state machine halts when end of file marker encountered. Before it is halting, it retrieve the content from the permanent feature set available, if any, for the same label under training and append it at the end. Thus it serves as a feedback mechanism.

Feature set pruning: At the end of each file processing, transient feature set is pruned using Synsets creation support available in the Thesaurus [16]. Synsets are nothing but grouped nouns, verbs, adjectives and adverbs into sets of cognitive synonyms, each expressing a distinct concept. Synsets can be inter linked through conceptual-semantic and lexical relations. Two types of count is maintained, one is for the token as it appears and the other is for the Synset participation instances. All the tokens are moved to permanent feature set along with their count value if it is not already present. It the token is already present only the value of the count updated.

Training corpus: It can be standard training set as Reuters-21578 or any text document like notepad text file. The text document is free from any formatting issues.

7. Merits of proposed method

This approach is easy to implement. There is no need of separate pre-processing module as it is tightly embedded in the State machine itself. Synsets creation promises good accuracy of the final outcome, of course with the demerit of having large dimensionality.

References

[1]. Durga Bhavani Dasari & Dr . Venu Gopala Rao. K, "Text Categorization and Machine Learning Methods: Current State of the Art" , Global Journal of Computer Science and Technology Software & Data Engineering, Volume 12, Issue 11, Version 1.0, Year 2012. pp 36-40.

[2]. Bao Y. and Ishii N., "Combining Multiple kNN Classifiers for Text Categorization by Reducts", LNCS 2534, 2002, pp. 340-347

ISSN 2348 – 7968

[3]. Bi Y., Bell D., Wang H. , Guo G. , Greer K. , "Combining Multiple Classifiers Using Dempster's Rule of Combination for Text Categorization", MDAI, 2004, 127-138.
[4]. Brank J., Grobelnik M., Milic-Frayling N., Mladenic D., "Interaction of Feature Selection Methods and Linear Classification Models", Proc. of the 19th International Conference on Machine Learning, Australia, 2002.
[5]. Chawla, N. V. , Bowyer, K. W. , Hall, L. O. , Kegelmeyer, W. P. , "SMOTE: Synthetic Minority Over-sampling Technique, " Journal of AI Research, 16 , 2002, pp. 321-357.
[6]. D. E. Johnson, F. J. Oles, T. Zhang, T. Goetz, "A decision-tree-based symbolic rule induction system for text categorization", IBM Systems Journal, September 2002.
[7]. Han X. , Zu G. , Ohyama W. , Wakabayashi T. , Kimura F. , “Accuracy Improvement of Automatic Text Classification Based on Feature Transformation and Multi-classifier Combination”, LNCS, Volume 3309, Jan 2004, pp. 463-468
[8]. Kim S. B. , Rim H. C. , Yook D. S. and Lim H. S. , "Effective Methods for Improving Naive Bayes Text Classifiers", LNAI 2417, 2002, pp. 414-423
[9]. Klopotek M. and Woch M., "Very Large Bayesian Networks in Text Classification", ICCS 2003, LNCS 2657, 2003, pp. 397-406
[10]. Leopold, Edda & Kindermann, Jörg, "Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?", Machine Learning 46, 2002, pp. 423 - 444.
[11]. Heui Lim, “Improving kNN Based Text Classification with Well Estimated Parameters”, LNCS, Vol. 3316, Oct 2004, pp. 516 - 523.
[12]. Sebastiani F., "Machine Learning in Automated Text Categorization", ACM Computing Surveys, vol. 34 (1), 2002, pp. 1-47.
[13]. Sung-Bae Cho, Jee-Haeng Lee, “Learning Neural Network Ensemble for Practical Text Classification”, Lecture Notes in Computer Science, Volume 2690, Aug 2003, pp. 1032 – 1036.
[14]. S. T. Dumais and H. Chen, "Hierarchical Classification of Web Content, " Proc. ACM SIGIR '00, July 2000, pp. 256-263.
[15]. http://nlp.stanford.edu/IR-book/html/htmledition/large-and-difficult-category-taxonomies-1.html
[16]. Princeton University "About WordNet." WordNet. Princeton University. 2010. http://wordnet.princeton.edu
[17]. Narayanan.S, Dr.Saswati Mukherjee, “Text Classification using Genetic Algorithm”, PG project report, School of Computer Science and Engineering, College of Engineering, Guindy, Anna University, India, 2007.

Narayanan.S Completed Masters in Software Engineering from Anna University. Currently working as an Assistant Professor in the department of Computer Science and Engineering, in Velammal Engineering College. Successfully guided many projects in career.