Text Classification using Language-independent Pre-processing

Yanbo J. Wang, Frans Coenen, Paul Leng, Robert Sanderson
Department of Computer Science, The University of Liverpool
Liverpool L69 3BX, United Kingdom
{jwang, frans, phl, azaroth} @ csc.liv.ac.uk

Abstract

A number of language-independent text pre-processing techniques, to support multi-class single-label text classification, are described and compared. A simple but effective statistical keyword identification approach is proposed, coupled with a number of phrase identification mechanisms. Experimental results are presented.
Keywords: Text Mining, Multi-class Single-label Text Classification, Text Pre-processing.

1. Introduction

In this paper we present and compare a number of approaches to text pre-processing for multi-class single-label text classification that operate in a language-independent manner. Rule-based classification systems operate, in general, by deriving a set of classification rules from a training set of previously-classified data: in this case, text documents. In the work described here, we apply a Classification Association Rule Mining (CARM) algorithm to derive these rules. CARM methods require each record in the training set to be expressed in the form of a set of binary-valued attributes, from which predicates for classification rules are formed.

The aim of the work described is to examine ways in which these textual attributes can be defined. We consider both single-word attributes (keywords) and phrases, defined in several ways. We wish to identify strategies that can be applied statistically, without deep analysis of the linguistic structure of the documents, and so will be essentially language-independent.

The following section describes some related works in text pre-processing for text classification. In section 3 we introduce a number of approaches we have considered for the identification of keywords and phases. In section 4 we present experimental results obtained using the TFPC (Total From Partial Classification) CARM algorithm [1], and in section 5 discuss our conclusions from this analysis.

2. Related Work

In theory, the textual attributes of a document could include every word / phrase which might be expected to occur in a given document set. However, this is

computationally unrealistic, so we require some method of pre-processing documents to identify the *key* words and phrases that will be useful for classification. Various techniques have been proposed to identify keywords within document sets such as Hidden Markov Models [2], Naïve Bayes [8] and Support Vector Machines [4]; however these all tend to make use of specific language-dependent meta-knowledge. Other methods use statistical information, such as word frequency. A well-known technique is the TF-IDF weighting (Term Frequency - Inverse Document Frequency) where TF is the number of occurrences of a given term in a given document and IDF is a measure of the total number of documents in a document set compared to the number of documents containing a given word [9]. Related techniques which include other statistical information derived from the document set have also been proposed in recent years (i.e., information gain [10], odds ratio [7], CORI [3], etc.) which improve the effectiveness of the approach.

3. Keyword and Phrase Extraction

The approach described here commences by processing each document (d) in the document base (D) to identify the "words" in it. The resulting collection of words is stored in a binary tree, in which each word is stored together with the identifiers of the documents in which it appears and its support value (S), i.e. the number of documents that contain the given word. Four types of word are identified:

1. **Stop marks:** Not actual words, but the punctuation marks ('!', ',', '.', ':', ';' and '?'),
2. **Noise words**: Words whose support is above / below user defined Upper / Lower Noise Thresholds (*UNT* / *LNT*) and which are therefore unlikely to prove significant. Noise words are thus either very common words that appear frequently across the document base or very rare words that appear in very few documents.
3. **Ordinary words (non-significant words)**: Non-noise words that do not serve to distinguish between classes.
4. **Significant words**: Keywords that do serve to distinguish between classes.

To identify significant words, we calculate, for each word (w) and class (C), the c*ontribution* made by w to C, defined as (proportion of documents in C that include w) / (proportion of all documents that include w). *Contributions* greater than 1 indicate that w may be a significant word for classifying C. We identify significant words as those whose *contribution* exceeds a significance threshold value (G) for at least one class.

Key phrases are then sequences of words that include at least one significant word. A number of different schemes for defining phrases can be identified depending on: (i) what are used as *delimiters* and (ii) what the *contents* of the phrase should be made up of:

1. **Delimiters: stop marks and noise words, Contents: significant and ordinary words (DelSN-contGO).** Phrases are made up of sequences of one

or more significant words and ordinary words, including at least one significant word.

2. **Delimiters: stop marks and ordinary words, Contents: significant and noise words (DelSO-contGN).** The rationale here is that there are many noise words which are used to link important words into a short, significant phrase, and so should not be treated as delimiters.
3. **Delimiters: stop marks and noise words, Contents: significant and "wildcard" words (DelSN-contGW).** As 1 but replacing ordinary words in phrases by wild card symbols that can be matched to any single word.
4. **Delimiters: stop marks and ordinary words, Contents: significant and "wildcard" words (DelSO-contGW).** As 2 but replacing noise words in phrases by wild card characters.

4. Experimental Results

For our experiments three document sets were used:

- The Reuters-21578 set[1] of 21,578 news documents. Following the practice of many researchers (for example [6]), we use only the 10 most popular classes, and considered only those documents uniquely placed in one of these. As a consequence two classes were dropped as they had very few documents associated with them, leaving 6,643 documents. We refer to this data set as Reuters.D6643.C8.
- The USENET (20 NEWSGROUP) set[2] [5]. There are exactly 1,000 documents per group (class) with the exception of one class that contains only 997. Due to efficiency issue, we randomly split this data into two document sets, each of 10 classes: NG.D10000.C10 and NG.D9997.C10.

A chosen document set is divided into a training set and a test set. The training set is processed to identify words and phrases, which are then used to recast all documents in the set as bags of words / phrases. From this, a text classifier is generated using TFPC, although in principle any general classification algorithm could be used. The accuracy of the resulting classifier is determined using tenfold cross-validation.

Initial experiments examined the four methods outlined above for defining key phrases; the selection of values for the *UNT*, *LNT* and *G* thresholds; and variations in the support and confidence thresholds used in the application of the CARM algorithm. In these experiments, the Del-SN algorithms (using stop marks and noise words as delimiters) performed significantly better than the alternatives. In general, and consistent with general Association Rule Mining (ARM) experience, we found a low support threshold (0.05-0.15%) worked best, and also a relatively low confidence threshold of around 35%. It was also found that a low *LNT* (of

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[2] http://www.cs.cmu.edu/afs/cs/project/theo-11/www/native-bayes/20_newsgroups.tar.gz

about 0.2%) was also beneficial to ensure that potentially significant words were not omitted.

Varying the other parameters was more problematic, partly because, for implementational reasons, it was necessary to limit the number of attributes (phrases) to 2^{15}. Because the three threshold parameters all influence the number of phrases generated, this limit was sometimes reached without identifying words and phrases that are significant in some classes. For this reason, a final set of experiments was conducted that refined the approach for identifying significant words. These experiments, focussed on the **DelSN-contGO** algorithm, began by identifying significant words for each class, placing these in order of their *contribution* to that class. The final selection of significant words was then made so that each class has an equal number *n*, i.e. the *n* words with the highest *contribution* to the class. Some results using the NG.D10000.C10 document set are given in Figure 1. Best accuracy is obtained with an *UNT* of 7% and a support of 0.05%.

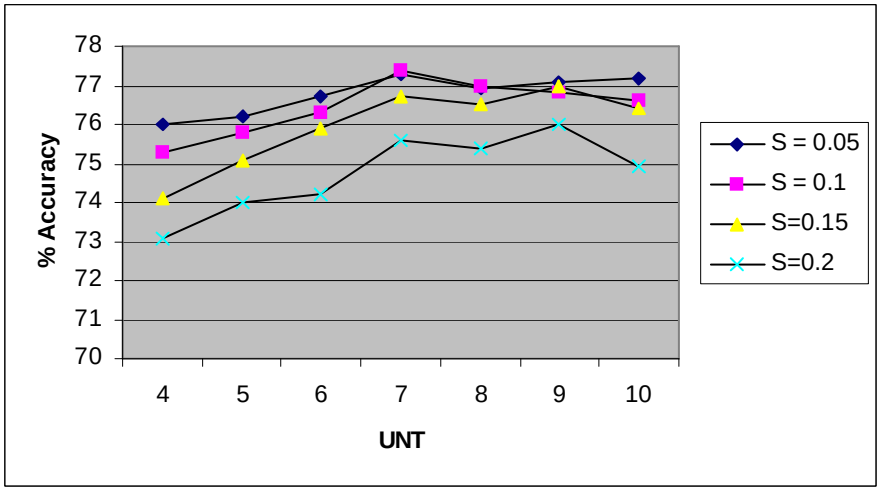

Figure 1 Accuracy obtained for a range of support and *UNT* values (confidence= 35%, *LNT*=0.2%, *G*=3, max # significant words=1500) for NG.D10000.C10

5. Conclusions

We have described here an approach to text classification that is based on a pre-processing of documents to identify significant words and phrases to be used as attributes in the classification algorithm. The methods we describe use simple numerical measures to identify these attributes, without the need for any deep linguistic analysis. Preliminary experiments have indicated values required for the threshold parameters to give best results. In future work, we intend to use the framework described to investigate other ways of defining phrases, and to determine optimal parameter values.

References

1. Coenen, F., Leng, P. & Zhang, L. Threshold tuning for improved classification association rule mining. In: Ho, T.B., Cheung, D. & Liu, H. (ed) Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2005), Hanoi, Vietnam, 2005 (LNAI 3518, Springer, pp. 216-225)
2. Collier, N., Nobata, C. & Tsujii, J. Extracting the names of genes and gene products with a hidden markov model. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), Saarbrücken, Germany, 2000 (pp. 201-207)
3. French, J.C., Powell, A.L., Callan, J., Viles, C.L., Emmitt, T., Prey, K.J. & Mou, Y. Comparing the performance of database selection algorithms. Technical report CS-99-03, Department of Computer Science, University of Virginia, January 1999
4. Kazama, J., Makino, T., Ohta, Y. & Tsujii, J. Tuning support vector machines for biomedical named entity recognition. In: Proceedings of the ACL (Association for Computational Linguistics) Workshop on Natural Language Processing in the Biomedical Domain (ACL 2002), Philadelphia, PA, USA, 2002 (pp. 1-8)
5. Lang, K. Newsweeder: learning to filter netnews. In: Proceedings of the 12th International Conference on Machine Learning (ICML 1995), Tahoe City, California, USA, 1995 (pp. 331-339)
6. Li, X. & Liu, B. Learning to classify texts using positive and unlabeled data. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, 2003 (pp. 587-594)
7. Mladenic, D. & Grobelnik, M. Word sequences as features in text-learning. In: Proceedings of the 7th Electrotechnical and Computer Security Conference (ERK 1998), IEEE Region 8, Slovenia Section IEEE, Ljubligana, Slovenia, 1998 (pp. 145-148)
8. Nobata, C., Collier, N. & Tsujii, J. Automatic term identification and classification in biological texts. In: Proceedings of the 5th Natural Language Pacific Rim Symposium (NLPRS 1999), Beijing, China, 1999 (pp. 369-375)
9. Spärck Jones, K. Exhaustivity and specificity. Journal of Documentation 1972; 28:11-21 (reprinted in 2004; 60:493-502)
10. Yang, Y. & Pedersen, J.O. A comparative study on feature set selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning (ICML 1997), Nashville, TN, USA, 1997 (pp. 412-420)