Automatic Scientific Text Classification Using Local Patterns: KDD CUP 2002 (Task 1)

Moustafa M. Ghanem, Yike Guo, Huma Lodhi, Yong Zhang
Dep. Of Computing, Imperial College of Science Technology & Medicine
180 Queens Gate, London SW7 2BZ, UK
{mmg, yg, yzhan, hml}@doc.ic.ac.uk

ABSTRACT
In this paper, we describe our approach for addressing Task 1 in the KDD CUP 2002 competition. The approach is based on developing and using an improved automatic feature selection method in conjunction with traditional classifiers. The feature selection method used is based on capturing frequently occurring keyword combinations (or motifs) within short segments of the text of a document and has proved to produce more accurate classification results than approaches relying solely on using keyword-based features.

Keywords
Document Categorization, Feature Selection, SVM.

1. INTRODUCTION
The task addressed in this paper is that of developing a system to automatically curate a database of scientific papers by analyzing a training data set of past human curation decisions. Sub-tasks 1 and 2 of this task (providing a ranking of the relevance of the papers and deciding whether a paper should be curated or not) can be handled directly using a document categorization framework. With a little preprocessing, sub-task 3 (identifying whether a particular item mentioned in a paper is related to a given concept) can be easily converted into a categorization question. In this section we briefly describe the background to document categorization and how it fits to all three sub-tasks.

Given a set of *N* training documents, a generic approach to document categorization first constructs a feature vector table such as that shown in Figure 1 where each document is represented by a score in relation to each of the *K* features. The table is then used as input to any traditional classification algorithm to generate a classification model. To classify an unseen paper, a feature vector is constructed using the same set of *K* features and then passed as input to the classification model. Clearly, the success of any document categorization method is closely tied to the selection of the features to represent the documents in question, we address this issue in Section 2.

ID	F_1	F_2	…	…	…	…	F_k	Class
ID1	0.8	0.12	0.3	0	0	0.12	0.97	Y
ID2								N
…								…
IDN	…	…						…

Figure 1 A Traditional Feature Vector Table

This generic document categorization approach can be directly applied to sub-task 2 that simply requires classifying a document as belonging to either class "Y" or class "N". Furthermore, by choosing a classifier that attaches a confidence value on the prediction (e.g. SVM light [1]), this generic approach can be directly used to provide a ranking of the relevance of all documents, and hence provide an answer to sub-task 1.

The same document categorization approach can also be easily modified to address sub-task 3. This sub-task requires deciding whether for a particular gene mentioned in a paper there is evidence of any of two given types of gene products (transcripts, and/or polypeptides) also being mentioned in the same paper. If a document has *n* gene names, we can create *n* virtual documents (by duplicating the document *n* times in the feature vector table), and thus each virtual document relates to a single gene/document pair. We then treat the sub-task as two categorization problems; the first is to predict transcript association and the second is to predict polypeptide association.

The virtual document approach will only work if the feature vectors for two virtual documents generated from the same physical document are sufficiently distinct. Our final approach to choosing the feature vectors neatly handles this issue.

2. KEYWORD-BASED CLASSIFIERS
We started our investigations into all sub-tasks by generating feature vectors based on keywords and an SVM classifier. Two questions arose in this case how should the keywords be selected and how should they be weighed in feature vector table.

Our first attempt was to experiment with a simple traditional automatic information retrieval approach. All words appearing in the document set were passed through a stop-word filter to eliminate common words, and then through a stemming algorithm to reduce variants of the same word to a canonical form. The unique terms remaining in the final output list were then used as the feature list.

Our second and third attempts were based on using domain knowledge to choose only relevant keywords as a basis for constructing the feature vectors. We experimented with lists of keywords supplied by local domain experts (biology postgraduate students). We also we used keywords extracted from evidence files supplied with the training data. These files contain what the human curators who supplied the training set perceived as evidence of the gene expression criteria for each paper.

Overall, for sub-task 2, we experimented with feature vectors ranging between 200 select keywords to about 60,000 words. In all experiments we used a traditional approach to weighing the significance of each term using TFIDF (Term Frequency/Inverse Document Frequency) to score each word. Unfortunately all experiments proved quickly to be disappointing generating poor classifiers with accuracy in the range of 60% on the training data.

3. PATTERN-BASED CLASSIFIERS

An alternative approach to addressing the curation problem is to develop techniques based on natural language processing (NLP) technologies [2]. The use of NLP techniques may offer a solution to the problems inherent with the simple keyword-based approach. These problems relate to the fact that the generated features and consequently the generated classification models do not capture the semantic relationship or the association between the words appearing in a document.

Rather than attempting an NLP approach, we decided to use an approach that captures the association between words appearing in each document. This is based in automatically identifying frequently co-occurring localized word patterns or motifs. By restricting the search for these patterns to localized parts of each document (e.g. a sentence or neighbouring sentences) our approach models the associations between these words and generates classifiers based on these associations.

Our patterns are defined using regular expressions on words automatically extracted from the documents. An example of one of our patterns is:

```
interact\s([a-z]*(\s)+)*genexx[a-z]+\s([a-z]*(\s)+)*bind\s
```

This patterns describes all sentences (or groups of sentences) having variants of the word “interact” followed by any number of words followed by a “gene name” followed by any number of words followed by variants of the word “bind”.

The first step in our approach was thus to automatically build a database of patterns by scanning the training data set using a variant of an association rule induction algorithm. To reduce the size of our pattern base we first filtered out from each document the sentences that do not contain either a gene name or a keyword extracted from the evidence files. Note that when creating the virtual documents used in sub-task 3, we only keep the sentences related to the gene name in question. This approach naturally leads to generating different feature vectors for virtual documents created from the same physical document. Our implemented pattern extractor only considered patterns that contain up to three words. We could have extended this, but felt that this was unnecessary given the execution time to generate the patterns.

The second step in the approach was to decide which patterns are to be kept within the pattern base and used as features. This can be decided either by an expert or automatically. The advantage of using a regular expression notation is it allows the end user to review and update the pattern base. In our final system we used a simple frequency threshold to remove infrequent patterns.

The third step in our approach was then to use the patterns as features to construct the feature vector tables, score each document against the patterns and pass the table as input to the classification algorithm.

Our final classifier was based on 335 automatically extracted patterns and provided accuracy in the range of 80% for the training data and providing the following accuracy results on the evaluation data: Ranked-list: 84%, Yes/No curate paper: 58%, Yes/No gene products: 59%.

4. CONCLUSIONS

Throughout our study, we used a mixture of custom-built text processing tools, data mining tools from the Kensington system [3], and a publically available tools. We experimented with various text pre-processing approaches, had to handle a large number of files, and managed a large number of parameters. By the end of the study we had designed a visual text mining system that is compatible with the Kensington visual programming paradigm, where data processing and analysis routines are represented as acyclic task graphs. We are currently evaluating the functionality of our system.

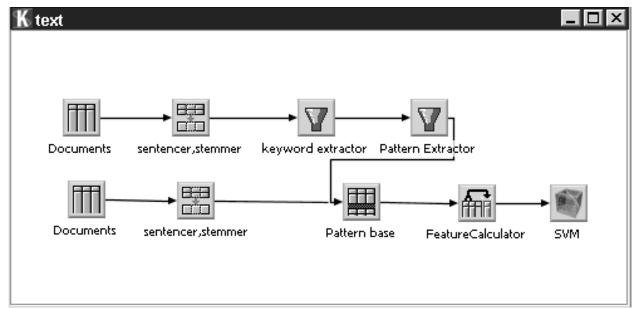

Figure 2 Kensington Visual Data Mining Interface

5. ACKNOWLEDGMENTS

We would like to thank our colleagues at both Imperial College and InforSense for their help and encouragement. In particular we would like to thank Bo Han for early versions of pattern extractor, Oliver Sturm and Hui Lim for their domain knowledge expertise in manually reading the documents and providing us with an early initial list of relevant keywords, and Emmanouil Mayiakis and Ramkumar Rasaratnam for their help in implementing various text pre-processing routines used in our system.

6. REFERENCES

[1] SVM light, http://svmlight.joachims.org/

[2] Foundations of statistical natural language preprocessing. Christopher D. manning and Hinrich Schutze, 2000, The MIT Press.

[3] Kensington Discovery Edition, http://www.inforsense.com

About the authors:

Moustafa Ghanem is a Research Fellow at the Department of Computing, Imperial College and has a PhD in High Performance Computing.

Yike Guo is Professor of Computer Science in the Department of Computing, Imperial College and Founder of InforSense Ltd.

Huma Lodhi is a Research Associate at the Department of Computing, Imperial College and has a PhD in machine learning.

Yong Zhang is a Research Associate at the Department of Computing, Imperial College and has a PhD in statistics.