PICCIL: Interactive Learning to Support Log File Categorization

David Loewenstern, Sheng Ma, Abdi Salahshour
IBM T.J. Watson Research & AC
Hawthorne, NY 10532, USA
davidloe, shengma, abdis@us.ibm.com

Abstract

Motivated by the real-world application of categorizing system log messages into defined situation categories, this paper describes an interactive text categorization method, PICCIL[1], that leverages supervised machine learning to reduce the burden of assigning categories to documents in large finite data sets but, by coupling human expertise to the machine learning, does so without sacrificing accuracy.

PICCIL uses keywords and keyword rules both to preclassify documents and to assist in the manual process of grouping and reviewing documents. The reviewed documents, in turn, are used to refine the keyword rules iteratively to improve subsequent grouping and document review.

We apply PICCIL to the problem of assigning semantic situation labels to the entries of a catalog of log events to support on-line labeling of log events.

1. PICCIL*

Most text categorization problems involve generalizing knowledge derived from a labeled sample of a large and ill-defined or even infinite space (e.g., the space of possible email texts) to classify new examples drawn from the same space, a process of supervised learning applied to text classification. There is a class of problems, by contrast, for which the sample space is well-defined and finite but no labeled samples exist. Typically, if perfect accuracy is not required, this class of problem is handled through unsupervised clustering methods; otherwise, text categorization is not used and the problem is handled using more labor-intensive methods.

The central insight driving the interactive learning process is that the task of labeling a data set is made much easier if the data set is grouped into clusters of semantically similar data. Each entry in such a cluster should receive the same label or, if not, there will often be some readily identifiable feature of the entry that makes clear why it is an exceptional case. This means that it is reasonable to expect that an expert would find it easier to identify the exceptions and correct them in a cluster of semantically similar data than to label an equal amount of data without semantic clustering.

The components and overall flow of the interactive learning process can be seen in Figure 1. The process takes the form of two nested iterative loops. First, the data (either a catalog or sample log files) are preprocessed into a standard format and stored. This data enters the outer iterative loop, where a subset is selected by an expert and labeled or reviewed in an inner iterative loop and the labels are added to the stored copy of the parsed data, along with information identifying the expert. The outer loop iterates until all experts have completed their tasks; typically, this means that the entire data set has been labeled or reviewed by at least one expert. Finally, in the case that a catalog was the source of the data, a table of (catalog entry identifier, situation) pairs is generated. By the end of the process, the stored parsed data is now labeled and reviewed, and may be used as a training set for supervised learning. As a by-product of the iterative process, a keyword rule classifier is also generated from the labeled data set.

The outer loop starts with the parsed but unfeaturized data. Initially, this data is also unlabeled, but on any iteration it may have already received a label during a previous iteration, permitting an expert to review or correct the decisions of previous experts. The data is preprocessed and preclassified by a keyword rule classifier. Initially, this classifier may be generated on another data set or a simple classifier containing only a few hand-generated rules. The data then enters an inner iterative loop in which the expert selects a subset of the data and uses a combination of keyword rules, features, and classifications to find a cluster of data which the expert judges to have similar semantics.

Alternatively, the expert may use keyword rules to find a cluster of data which the classifier judges to have similar semantics, or the expert may combine the two approaches. In any case, the expert reviews the cluster and corrects any potential misclassifications.

The inner loop iterates until the expert determines that the session is over. The data, with the expert's

[1] PICCIL: Process for Iterative Corpus Categorization by Interactive Learning.

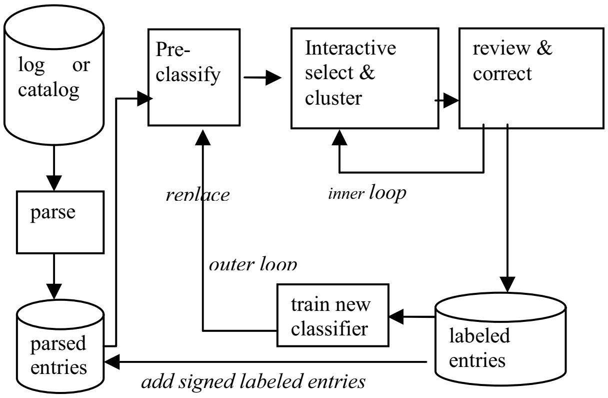

Figure 1. The interactive learning process (data flow).

labels, are then used to create a new keyword list classifier using a modified association rule generation algorithm. The new classifier replaces the earlier classifier in the outer loop. The outer loop iterates with a new session with the same or a different expert. The process completes when all experts have completed their tasks, typically by generating a completely labeled set of data.

2. Real-world Application

We applied PICCIL to a small real-world problem, the labeling of situations in logs generated by the IBM End-to-End Probe Platform (EPP). The EPP is an IBM product for monitoring performance of distributed applications. EPP logs are derived from a catalog of 406 entries. Figure shows some sample EPP catalog entries.

```
Error opening ({0}) file: {1}
Could not read a line: {0}.
Invalid input argument: {0}.
```

Figure 2. Sample EPP catalog entries (msg fields).

We gave a three-person review panel of EPP developers a version of PICCIL. Our interaction was limited to supporting the software, monitoring the process, and eliciting recommendations; the authors were not part of the EPP review process. The review process was held over the course of one week for a total of approximately 40 person-hours of review time.

We found that the PICCIL significantly reduced the amount of time involved in the labeling and review process. The 40 person-hours of time worked out to 6 minutes per catalog entry. A comparison using a hand labeled sample required 10-20 minutes per catalog entry depending upon the familiarity of the expert with the CBE situation specification. As was expected, there were several reasons for the reduction in labeling time. First, the classifier improved as the process iterated, improving the quality of preclassification, converging on an accuracy of roughly 60%. As preclassification quality improved, fewer labels required correction. Second, as the classifier improved, the labels supported grouping semantically similar entries together better, and so grouping became faster and more accurate. This meant that more entries could be reviewed together. Third, PICCIL's integrated support for the review process helped manage change tracking over the week-long process, although the time savings involved was difficult to measure objectively.

Acknowledgements

We thank Herb Lee, Jeff Perry, and Mariusz Sabath for their valuable feedback and the use of the EPP catalog, and Kevin Minerley for useful discussions, the initial classifier used in the EPP application, and the use of the Z-OS APAR data set.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. of Very Large Data Bases*, 1994.
[2] M. Hearst, Untangling text data mining, *ACL-99*, 1999.
[3] T. Li *et al.*, Document clustering via adaptive subspace iteration, *SIGIR-04*, 2004.
[4] T. Li and S. Ma, IFD: Iterative feature and data clustering, *SDM-04*, 2004.
[5] B. Liu *et al.*, Integrating classification and association rule mining, *KDD-98*, pp. 80 – 86, 1998.
[6] D. Meretakis and B. Wüthrich, Extending naïve Bayes classifiers using long itemsets, *KDD-99*, pp. 165 – 174, 1999.
[7] F. Sebastiani, Machine learning in automated text categorization, *ACM Computing Surveys* 34, pp. 1 – 47, 2002.
[8] M. Steinbach *et al.*, A comparison of document clustering techniques, *KDD Text Mining Workshop*, 2000.
[9] Y. Yang, A scalability analysis of classifiers in text categorization, *SIGIR-03*, 2003.
[10] Y. Yang and X. Liu, A re-examination of text categorization methods, *SIGIR-99*, pp. 42 – 49, 1999.