Date: 2017/01/31

Master Thesis Project Proposal

Automating Text Categorization with Machine Learning: Error responsibility routing in a multi-layer hierarchy

***Alexander Persson**, Chalmers* *alepers@student.chalmers.se*	*Relevant courses:* *TIN172 – Artificial Intelligence* *TDA206 – Discrete Optimization*
***Ludvig Helén**, Linköping University* *Masterprogram inom datavetenskap* *ludhe894@student.liu.se*	*Examiner: Ola Leifler (ola.leifler@liu.se)* *Supervisor: Jonas Wallgren (jonas.wallgren@liu.se)*

Academic supervisor	*Selpi at Dept. of Applied Mechanics*
Company supervisor	*Michael West, Ericsson AB*

Introduction

Machine learning is a popular topic in the field of computer science and widely used to make accurate predictions based on existing training data. The company Ericsson is now taking steps towards embracing these techniques and applying them to their product development cycle. More specifically, they would like to use machine learning to automate the evaluation of trouble reports (TRs). A trouble report is a description of some issue or bug in the company's intellectual property with relevant log data attached to it. The current method of evaluating TRs is an arduous, bureaucratic process where experts manually analyze the logs to identify the product team responsible for correcting the issue, resulting in long lead times between meetings at great cost for the company.

The challenge posed is to accurately predict which company product team a TR belongs to, based on pre-defined fields and error descriptions. In addition to merely making this classification, the implementation should derive the route through the company hierarchy the TR would normally take. This is important information that serves the purpose of enabling a department to preprocess a TR before routing it to the next relevant department, and so on.

Through some abstraction, the challenge can be formulated as a text categorization problem with the added property of being able to present a decision trace. The nature of the data suggests that a dimensionality reduction technique can be applied, which in this case means selecting a subset of relevant terms encountered in the problem descriptions to use while classifying the documents. In order to accomplish this, a few research questions are posed:

- How can the corporate process of assigning responsibility for correcting product errors be automated with the use of supervised text categorization?
- Can a dimensionality reduction technique improve the performance of the applied classifier? Which technique works best for this task?

There exists a prototype implementation built by a research graduate at the company. This prototype predicts the correct category roughly 60 % of the time, which is deemed not to be enough and should be improved on.

Context

The recent work by Basu & Murthy [1] in the area of text categorization has yielded some improvements for dimensionality reduction which reduces the (very large) term space one must handle in documents of great size. This yields a more efficient text categorizer. Other, somewhat older algorithms, address this same issue. This could come in use due to the nature of the size of the error descriptions in the training set (>100 lines of text). The idea is to incorporate these term selection algorithms and complement them so that an artificial intelligence can derive the rationale behind the classifiers prediction. Text categorization and dimensionality reduction are actively researched topics, so it is not clear what the best choice of technique for this task is.

The topic of assigning bug reports to software development teams has been actively researched for many years. The work usually boils down to a one-to-one mapping between a bug report and an individual developer/team, and do not seem to encapsulate the vast department hierarchy in a big corporation. A bug report may be in need of processing in several layers of the hierarchy and to automatically predict such a route should be of interest in many cases. A recent journal article by Jonsson et al. [2] proposes that a model called Stacked Generalization can be used to classify reports, where a range of different classifiers are combined to achieve greater precision, outperforming many individual classifiers. The authors generalize the bug handling process here as well, making no mention of this proposed multi-layer approach.

Goals and Challenges

The goals of this thesis are:

- Identify suitable machine learning technique for achieving text classification precision > 60 %
- Identify suitable dimensionality reduction technique for boosting performance of the classifier
- Build a classifier system that can accurately assign TRs to appropriate product organizations with ability to trace the decision

The main challenges involved in this thesis work will be to improve on the precision of the prototype classifier, as well as producing a feasible route through the corporate hierarchy. Somewhat older literature suggests around 60 % precision is common when it comes to text categorization, but since it is a highly researched topic new techniques and algorithms have emerged.

Approach

As this is a classification problem with supervised learning, it seems likely that it could be solved through logistic regression or the Support Vector Machine method. Combined with the fact that there is a large number of training examples (> 25,000) available, this choice of method appears sound since it has been proven to be effective under similar circumstances [3]. If it proves to be inadequate, a different technique will be evaluated.

The work will be implemented in TensorFlow, an open source Python library created by the Google Brain team suited for optimization problems in the area of machine learning [4]. The library provides implementations of many useful standard algorithms and is comparable with other machine learning libraries in terms of performance [5].

To evaluate the quality of the implementation, it will be compared to the existing prototype in its current state. The performance aspect is also important, and the prototype provides a baseline for comparison with the new implementation. There is currently no prototype for the decision trace.

References

[1] T. Basu and C. A. Murthy, "A supervised term selection technique for effective text categorization," *International Journal of Machine Learning and Cybernetics,* vol. 7, no. 5, pp. 877-892, 2016.

[2] L. Jonsson, M. Borg, D. Broman, K. Sandahl, P. Runeson and S. Eldh, "Automated bug assignment: Ensemble-based machine learning in large scale industrial contexts," *Empirical Software Engineering,* vol. 21, no. 4, pp. 1533-1578, 2016.

[3] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys (CSUR),* vol. 34, no. 1, pp. 1-47, 2002.

[4] "TensorFlow," 2016. [Online]. Available: https://www.tensorflow.org/. [Accessed 20 January 2017].

[5] M. Schrimpf, "Should I use TensorFlow," Cornell University Library, 2016.