Automated Classification of NASA Anomalies Using Natural Language Processing Techniques

Davide Falessi
Fraunhofer CESE, College Park, USA
dfalessi@fc-md.umd.edu

Lucas Layman
Fraunhofer CESE, College Park, USA
llayman@fc-md.umd.edu

***Abstract*—NASA anomaly databases are rich resources of software failure data in the field. These data are often captured in natural language that is not appropriate for trending or statistical analyses. This fast abstract describes a feasibility study of applying 60 natural language processing techniques for automatically classifying anomaly data to enable trend analyses.**

Keywords—NLP; natural language processing; software failure

I. INTRODUCTION

NASA programs track *anomalies*: operational behaviors that deviate from the project's specification. Software failures are one cause of anomalies. *Anomaly reports* capture, in detail, a *Description* of the anomaly, severity assessments, a description of the *Root Cause* (after investigation), and the *Corrective Action* taken. These fields are plain text, while metadata fields (e.g., failure type, failing component) are often incomplete or inaccurate, making trend analysis inaccurate and unreliable. Accurately answering simple questions like, "how many of these anomalies were caused by software" is currently impossible without reading the text of each anomaly. The number of anomalies from Goddard Space Flight Center and Jet Propulsion Laboratory missions is over 30,000, making reading the whole set of anomalies intractable and even random sampling costly. Software assurance engineers at NASA need to make use of pattern and trend data on how software fails in operation, but the data sets do not support such queries in their current form.

To enable trending and analysis, we are applying Natural Language Processing (NLP) techniques to automatically classify anomaly reports based on their textual descriptions. This paper reports a feasibility study to assess the ability of 60 different NLP techniques to automatically classify the root cause and corrective action of NASA anomalies.

II. NLP TECHNIQUES

Given a pair of text fragments (i.e., the anomaly description in our study), we use NLP techniques to provide a similarity measurement with a value ranging between 0 and 1. NLP has been applied in different areas of software engineering for many years, e.g., Runeson et al. [1] reported promising results in using NLP techniques to detect equivalent defect reports. Menzies et al. [2] reported success in using NLP and data mining techniques to automatically triage NASA defect reports. We expand on this work by systematically evaluating many NLP technques (60) to improve predictive accuracy. In general, NLP techniques measure the similarity between two text fragments by following four main steps:

Step 1: Term extraction— Terms are extracted from the text fragments, removing special characters, and possibly splitting composite identifiers before any comparison takes place. In this work, we consider: 1) simple extraction (tokenization and stop-word removal) and 2) Part Of Speech tagging using the Stanford natural language parser, and 3) stemming, e.g., using the Porter stemmer.

Step 2: Term weighting— Terms are weighted, based on their occurrences in the analyzed text fragments. In this work we consider: 1) Term Frequency (TF); 2) Inverse Document Frequency (IDF); 3) their combination (TF-IDF); 4) Simple, and 5) Binary.

Step 3: Building algebraic models— A model of documents is created. We consider: 1) a Vector Space Model (VSM), and 2) WordNet, where synonyms are pre-defined in a thesaurus.

Step 4: Computing similarity— Finally, the similarity between the two text fragments is computed via similarity metrics. In this work we consider: 1) Cosine, 2) Jaccard, 3) Dice, 4) Resnik, 5) Lin, 6) Jiang, and 7) Pirrò and Seco.

A single NLP technique is a combination of four steps. The possible combinations of the four approaches result in 60 different NLP techniques, i.e. 60 different measures of similarity. For further details about NLP techniques, including the ones adopted in this study, see [3].

III. EVALUATION PROCEDURE

We created an oracle of 480 randomly sampled (without replacement) anomalies from a dataset of 9921 anomalies from NASA spaceflight missions since 1990. Each anomaly in the oracle was inspected and categorized in two variables:

1. Root Cause class: [software, hardware, operations, environmental, or unknown error]
2. Corrective Action class: [software fix, hardware fix, operational workaround, none/unknown].

Our underlying assumption is that equivalent anomalies should have the same root cause and corrective action class.

We randomly selected with replacement 33 anomalies from the oracle set. We then performed the following:

For each anomaly, *a*, in the sample set:

- For each of the 60 NLP techniques *p*:
  - Compute the similarity score of all other anomalies to *a* using the anomaly Title or Description as input.
  - Rank all other anomalies by their similarity score to *a*.
  - Label *a* with the Root Cause and Corrective Action of the top-ranked (most similar) anomaly.

The average correctness (computed class(*a*) = oracle class(*a*)) for the sample set was computed for each NLP technique. The entire process was executed 10 times and the results averaged for each NLP technique. The execution of the NLP techniques was facilitated by the framework in [3].

IV. Limitations

An oracle was needed to measure the correctness of the automated classifications to test the feasibility of the approach before it can be applied to unlabeled data in future work. The evaluation results are limited to the 480 anomaly dataset only. Also, we are limited to 10 random samples of 33 anomalies from the 480 set. Using the entire 480 dataset would require computing the $27*10^6$ similarity scores (114,960 pairs of anomalies, 60 NLP techniques, two class fields, and two input fields), which was intractable given our current experimental framework. Our approach consists of measuring the performances on 10 different samples of 33 anomalies each (528 anomaly pairs per sample). Moreover, due to random effect, we expect that at least one NLP technique will perform well on the 480 dataset. Thus, the use of 10 different smaller datasets, rather than one larger dataset, improved both results validity (reduced influence of random effect) and feasibility (reduced number of similarity measurements) of the study.

V. Results and Discussion

Figure 1 reports the distributions of average *correctness* for the 60 NLP techniques over the 10 random samplings, using the Description and Title to predict the Corrective Action and Root Cause. *Correctness* is the percentage of anomalies where the NLP-computed class(*a*) = oracle class(*a*)).

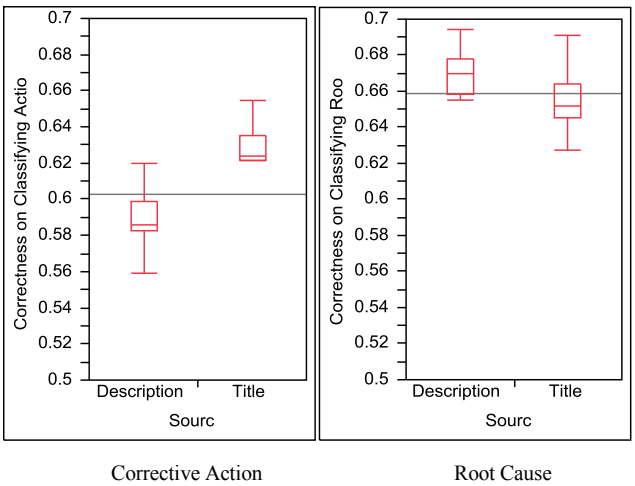

Fig. 1. Performances of 60 NLP techniques in classifying the Corrective Action and Root Cause of software anomalies using the Description or Title.

Table I. Correctness of the best NLP technique in classifying the Corrective Action (a) and Root Cause (b)

Best NLP technique	Field	Source	Average	STDV	Min	Max
VSM ,TF_IDF ,COSINE, Stanford (nouns)	Root	Description	0.712	0.109	0.515	0.879
VSM ,TF_IDF ,COSINE, Porter stemmer	Action	Title	0.655	0.091	0.515	0.848

Figure 1 shows that all NLP techniques are correct most of the time (correctness > 0.50). A random approach, i.e. using a similarity value that is randomly computed, resulted in ~0.53 correctness. Table 1 below describes the correctness across the 10 different datasets of the NLP techniques having the highest correctness in classifying a specific filed, independently from the source.

As shown in Table I, the highest correctness is achieved by using the field Description as the source of the NLP technique to classify the Root Cause variable and the Title as source to classify the Corrective Action variable. The two best NLP techniques are similar, specifically they share the same term weighting (TF-IDF), algebraic model (VSM), and similarity metrics (Cosine). They differ only in the term extraction approach. Moreover, the standard deviation of ~0.1 indicates stable performance over the 10 datasets.

VI. Future Work

The presented results support the feasibility of applying NLP processing techniques to automatically classify NASA anomalies and motivates further research. Our ultimate goal is to provide an analytical engine for NASA engineers to query, analyze, and trend the ways in which software fails in the field and the root causes of these failures. To accomplish this goal, we must approve the scalability of our approach, both through technical refinement and a methodology for selecting a "good enough" classifier without exhaustive evaluation, as the current computational cost is quite high. We stress that the observed levels of correctness are a lower bound. Correctness can be improved by combining the use of NLP techniques with Data Mining approaches, such as combining several NLP techniques together (see [3]), combining textual sources together (using both Title and Description as source of information), or incorporating meta-information, such as the period of the anomaly, the project, etc.

Acknowledgment

This work is funded by NASA OSMA Software Assurance Research Program grants NNX08A260G and NNX11AP93G.

References

[1] P. Runeson, M. Alexandersson, and O. Nyholm, "Detection of Duplicate Defect Reports Using Natural Language Processing," in 29th International Conference on Software Engineering (ICSE'07), 2007, pp. 499–510.

[2] T. Menzies and A. Marcus, "Automated severity assessment of software defect reports," in 2008 IEEE International Conference on Software Maintenance, 2008, pp. 346–355.

[3] D. Falessi, G. Cantone, and G. Canfora, "Empirical Principles and an Industrial Case Study in Retrieving Equivalent Requirements via Natural Language Processing Techniques," IEEE Transactions on Software Engineering, vol. 39, no. 1, pp. 18–44, Jan. 2013.