TRANSFORMER-BASED MODEL FOR THE SEMANTIC PARSING OF ERROR MESSAGES IN DISTRIBUTED COMPUTING SYSTEMS IN HIGH ENERGY PHYSICS

D.V. Grin[1,a], M.A. Grigorieva[2,b]

[1] *National Research Center "Kurchatov Institute," Akademika Kurchatova sq., 1, Moscow, 123182, Russian Federation*

[2] *Scientific Research Computing Center, Lomonosov Moscow State University, Leninskie Gory, 1, p.4, Moscow, 119991, Russian Federation*

E-mail: [a] dymong@yandex.ru, [b] maria.grigorieva@cern.ch

Large-scale computing centers supporting modern scientific experiments store and analyze vast amounts of data. A noticeable number of computing jobs executed within the complex distributed computing environments ends with errors of some kind, and the amount of error log data generated every day complicates manual analysis by human experts. Moreover, traditional methods such as specifying regular expression patterns to automatically group error messages become impractical in a heterogeneous computing environment without a well-defined structure of error messages. ClusterLogs framework for error message clustering was developed to address this challenge. The framework can discover common patterns in error messages from various sources and group them together. One of the essential results of this process is the clear automated description of the resulting clusters, which will be used for the analysis. In this research, we propose that interpreting error messages as a natural language allows us to use transformer-based deep learning models such as BERT for this task. A model for extracting the relevant part of messages was trained and integrated into ClusterLogs to represent each cluster as a few actionable items, ensuring better interpretation and validation of the results of clustering.

Keywords: log analysis, clustering, natural language processing, deep learning, transformers

Dmitry Grin, Maria Grigorieva

1. Introduction

One of the most important goals of the monitoring system of large-scale distributed computing systems is to detect and analyze faults and errors in the infrastructure. It is often useful to investigate the distribution of different types of errors: detect the most frequent error patterns for some period of time, carry out a retrospective analysis of these patterns and discover common characteristics of computing jobs that finished with particular failures.

But due to the scale of the problem significant resources are needed to solve it, and today the error message analysis is only partially automated, mostly in cases such as the error message format being known in advance. Another important aspect of the problem is being able to not only analyze textual patterns, but also link them to the metadata such as job identificators in workload management systems, or computing site and launch parameters for the particular computing job.

To solve the aforementioned problems, the ClusterLogs framework [1, 2] was developed. It is devoted to the automation of error message analysis tasks, allowing the human experts to investigate different types of errors and discover common characteristics of the jobs that resulted in those errors.

This paper describes the improvement of the cluster description as well as optional improvement of the clustering results using the BERT model to extract the significant parts of the messages.

2. Error message clustering overview

The analysis of the data for large-scale physics experiments requires an ability to execute millions of computing jobs simultaneously, which explains the need for robust infrastructure. Maintaining this infrastructure includes the detection, analysis and prevention of different kinds of errors. For example, for the ATLAS experiment in CERN a big part of the error detection is done by the PanDA workload management system. It is responsible for the execution of physics analysis and computing jobs, and has proved itself to be a scalable and reliable system capable of handling millions of jobs daily. When, during the execution of a job, some component or service fails, the error code and error message are usually registered and saved in the database in the record of the corresponding job.

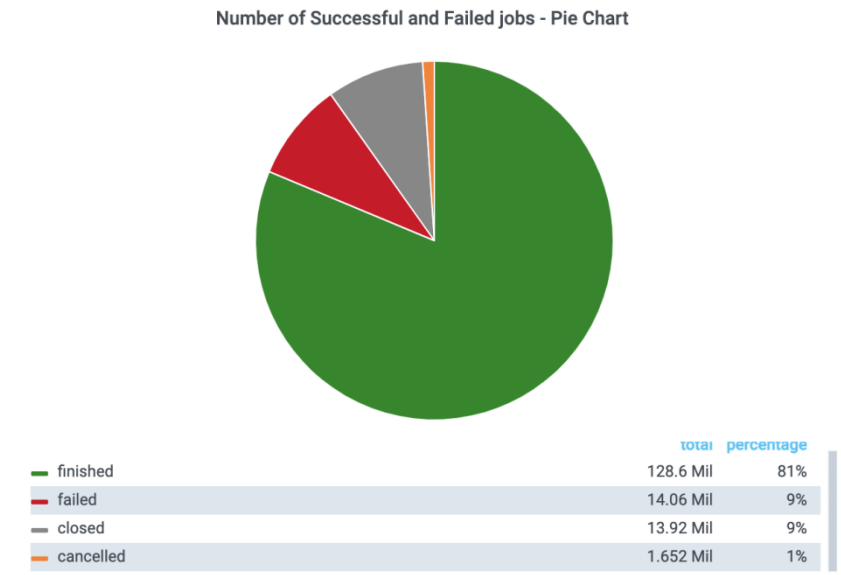

Figure 5. Number of successful and failed jobs for a one-year period

Figure 1 depicts the chart that details the number of jobs that failed or ended successfully in a year. Around 9% of the jobs ended with failures of some kind.

Each category of messages has dozens of error codes, and each code is associated with different error conditions that can have non-standardized (often unpredictable) textual patterns.

And so, the problem arises when a new error message appears that is not yet associated with the error code. This can happen because the standards of logging can change with time. Moreover, new errors may appear due to different factors such as the rapid development of experiments using new algorithms and program modules, especially considering the messages that are written by the users, where there's no standardized form and structure of those messages.

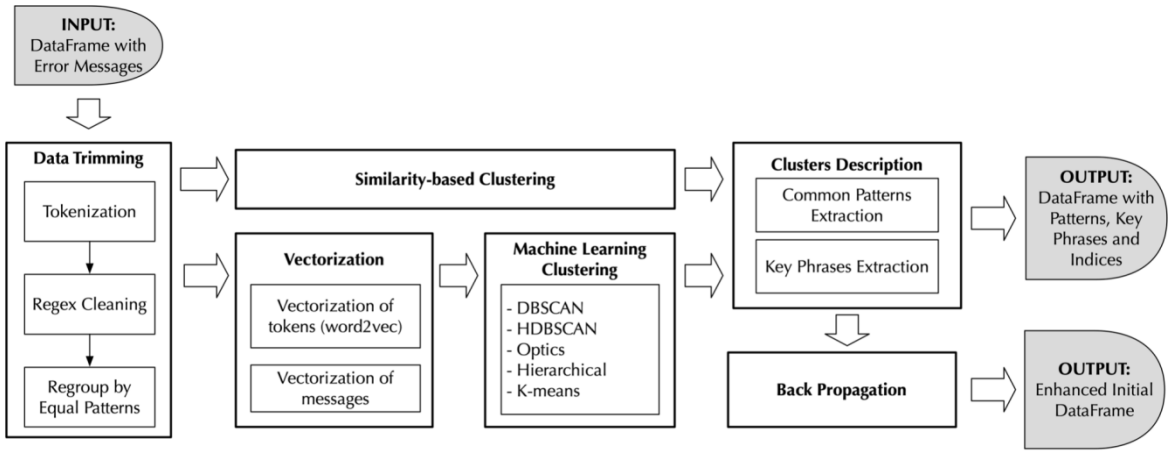

Figure 6. The overview of ClusterLogs structure

ClusterLogs is a framework that allows to automatically cluster error messages with associated metadata. The overall framework structure can be seen on Figure 2. The preprocessing of error messages is the first step. Ideally a domain knowledge can be used to remove all the unnecessary parts of messages, but even just filtering out stuff like the exact numbers, punctuation, file paths and URLs allows to greatly reduce the size of the dataset. Then the word embedding model such as word2vec [3] is used to transform textual messages into the numeric vectors. And, when each message is represented as a numerical vector, traditional clustering methods like DBSCAN or k-means can be used. After the clustering each cluster is described using common patterns of the messages, and key word and phrase extraction.

3. Relevant part extraction model

The current results are represented in two major ways. The first one is a table, with rows representing particular clusters. For each cluster one or more textual templates describing all the messages in the cluster is shown, along with key phrases extracted from it. The second way of showing the results is the interactive visual representation of the clusters. The t-SNE algorithm is used to perform a dimensionality reduction of vectorized data to two dimensions. Every point on the plot represents a textual pattern common to one or more error messages. The size of each point is proportional to the number of messages (on a logarithmic scale) while the colour of the point corresponds to the cluster this pattern belongs to.

There are some weak points in the table-based approach. First of all, common phrases extracted from a small text such as the error message may not be representative of the whole cluster. A related problem is that not every part of an error message will be relevant for message analysis. For example, a message starting with a phrase "Non-zero return code from ..." can represent any error. This phrase does not give us useful information itself, and it can appear regardless of the actual error, reducing the accuracy of the clustering.

To solve this problem, we propose the usage of the BERT model published by Google in 2018 [4]. One possible usage is an alternative for word2vec vectorisation. BERT, as a state-of-the-art approach to natural language processing, can give us better results on this stage. The problem of this approach is that for achieving better results compared to the current model, a relatively big annotated dataset is necessary. The model without fine-tuning is included into the ClusterLogs as a point of comparison.

But the main point of this article is the usage of the BERT model for the extraction of a significant part of error messages to better represent the error.

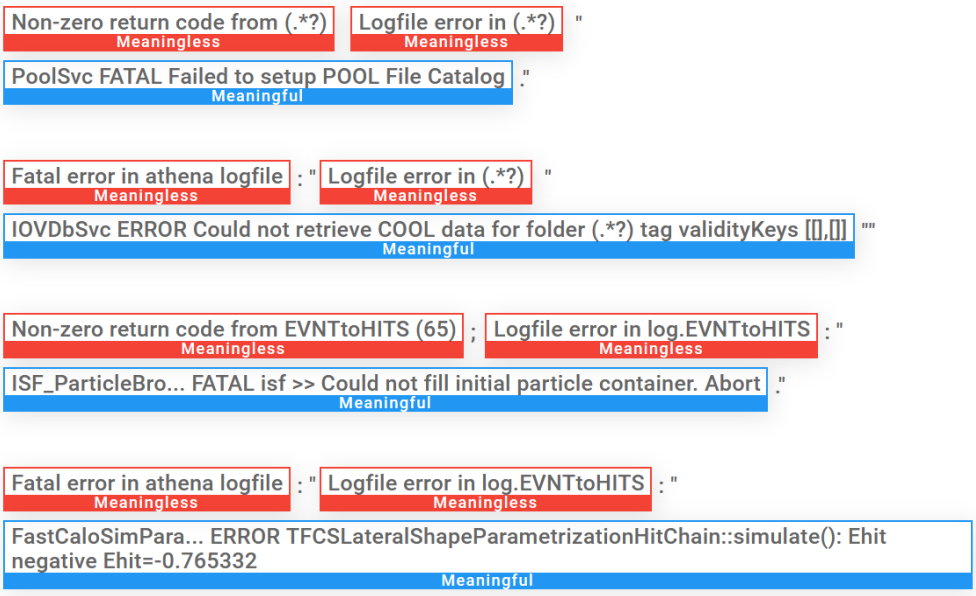

Figure 7. The fragment of an annotated dataset

Figure 3 gives the example of dataset annotations. As can be seen, most messages have some irrelevant parts that don't give the information about the error (annotated in red), and the error description part (annotated in blue).

It has to be noted that the ClusterLogs framework itself is dataset-independent, and its usage does not require any additional preparations. For a new step, a small dataset of error messages had to be annotated, which was later used to fine-tune a BERT model. This work will have to be done for every new type of message, but the dataset of less than a thousand messages was sufficient for a near-perfect accuracy.

Figure 4 shows an example of a new system in action. The number of messages represented by those patterns are in the first column while the patterns themselves with relevant parts highlighted in bold are in the second column. The model correctly identified a significant part of every message, except for one unnecessary word in one of the clusters. This is a dataset of execution errors of computing jobs, consisting of around 70 thousand messages, and which translated into 89 clusters, with around 200 different message patterns. Compared to a human expert, the model's significant part extraction was different only in several patterns, in which cases the model included some unnecessary information.

588	• Fatal error in athena logfile: "**Long ERROR message at line (.*?) (see jobReport for further details)**"
588	• Non-zero return code from (.*?) (65); Logfile error in (.*?) **FATAL Check number of writes failed. See messages above to identify which continer is not always written**"
507	• Non-zero return code from EVNTtoHITS (8); Logfile error in log.EVNTtoHITS: "**SyntaxError: unexpected character after line continuation character**"
448	• Non-zero return code from AODtoDAOD (33); Logfile error in log.AODtoDAOD: **(.*?) FATAL #BTAG# Failed to retrieve tool (.*?) = (.*?)** • Non-zero return code from POOLMergeAthenaMPAOD0 (64); Logfile error in log**.POOLMergeAthenaMPAOD0: "ToolSvc.CaloTrackingGeometryBuilder FATAL Failed to retrieve tool LArVolumeBuilder = PublicToolHandle('LAr::LArVolumeBuilder/LArV (truncated)**
428	• Non-zero return code from RAWtoALL (8); Logfile error in log.RAWtoALL: "**UnboundLocalError: local variable 'fnt' referenced before assignment**"
343	• Non-zero return code from generate (8); Logfile error in log.generate: "**IOError: [Errno 2] No such file or directory: (.*?)** • Non-zero return code from generate (8); Logfile error in log.generate: "**RuntimeError: No output LHEF file produced by Powheg. Terminating job**."

Figure 8. The fragment of the results table (relevant parts highlighted in bold)

4. Conclusion

The inclusion of BERT model into ClusterLogs pipeline shows that a small dataset is sufficient to solve the problem of significant part extraction from error messages, which gives an improvement in the description of clustering results. The same method was used to remove the common irrelevant parts at the start and only cluster the meaningful parts of error messages, which improves the accuracy of the message categorization. These enhancements were incorporated into both the framework itself and the graphical frontend, so that the experts could more easily use it for error message analysis.

References

[1] M. Grigorieva, D. Grin. Clustering error messages produced by distributed computing infrastructure during the processing of high energy physics data // International Journal of Modern Physics A, 36(10), 2150070-130 (2021)

[2] M. Grigorieva, D. Grin. “ClusterLogs” // Github. Available at: https://github.com/maria-grigorieva/ClusterLog (accessed 16.09.2021)

[3] T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient estimation of word representations in vector space // arXiv preprint: arXiv:1301.3781 (2013)

[4] J. Devlin, M. W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding // arXiv preprint: arXiv:1810.04805. (2018)