Software error messages are generated from template strings, filled with context information. We aim to improve error classification by learning the templates and identify the context information from error messages generated by different applications. We want to be able to group messages like

Cannot read property 'vdata1234' of null
Cannot read property 'isDestroyed' of undefined
eCommerceData is not defined
The action 'buy_now' could not be found for Api::V1::LotsController

Our approach uses PCA to transform the data to $k \geq t$ principal components then K-means to find the messages generated by the templates.

Assuming that the messages have been generated by $t$ templates, we apply PCA to transform the data to $k \geq t$ principal components then K-means to find the messages generated by the templates.

PCA was very sensitive to collisions and outliers so it was practically useless without data cleaning. We used the results of agglomerative clustering for it. We kept the errors that appeared in multiple projects and identified generated at least 100 distinct messages. This covered 12 patterns. PCA found 3 identical clusters and 9 mixed ones.

Unsupervised learning is a great to explore large datasets and understand its structure. The results of agglomerative clustering saves hundreds of hours by identifying the recurring templates in messages. It improves the specificity of grouping that 80% of users found helpful.

Using PCA for grouping without recovering the templates looks an extremely promising alternative. It lets us easily handle other features like the stack frames. However, it seems that we need much larger dataset and more sophisticated features to overcome the noise.

As an alternative, the results of a agglomerative clustering can be used to label and augment datasets that can be used to train neural networks.

References