AutoARTS: Taxonomy, Insights and Tools for Root Cause Labelling of Incidents in Microsoft Azure

Pradeep Dogga

UCLA

Chetan Bansal  Richie Costleigh  Gopinath Jayagopal  Suman Nath  Xuchao Zhang

Microsoft
Incident Postmortems in Clouds

On-Call Engineers → Fix → Cloud (X) → Document → Postmortem
Retrospective Analysis using Postmortems

On-Call Engineers ➔ Document ➔ Postmortems ➔ Analyze ➔ Insights ➔ Top Root Causes

Owners ➔ Analyze ➔ Trends of Specific Root Causes
Retrospective Analysis Today

On-Call Engineers → Manual Document → Postmortems → Manual Label → Insights → Trends, Aggregation, ...

Owners → Manual Analyze → Postmortems → Manual Label → Postmortems
Root Cause Labelling Today – Taxonomies

Postmortems

Label

Labelled Postmortems

Team 1
- Network
- ...

Team 2
- DC Networking
- ...

Team n
- Code bug
- ...

Ambiguous
Incomplete
Flat
Root Cause Labelling Today

Postmortems → Manual Label → Error-prone Incomplete → Labelled Postmortems
Retrospective Analysis Today

On-Call Engineers

Manual Document

Postmortems

Label

AutoARTS

Manual Analyze

Owners

Insights

Trends, Aggregation, ...

What AutoARTS is about

**Problem:** Lengthy postmortems, poor root cause taxonomies, error-prone and incomplete root cause labelling.

**Solution:** Develop comprehensive taxonomy, bootstrap labelling postmortems, generate succinct contexts and labels with ML.

**Ideas:** Leverage hierarchy in taxonomy, train text encoders w.r.to tags, finetuning gap sentence summarization.

**Opensource Taxonomy:** Share wide variety of contributing factors with others and develop continuously.
Postmortems – Treasure Troves of Rich Debugging Insights

- Title, symptoms, root causes, mitigation steps, 5-Whys, etc.

- Written in natural language with little to no structure.

- Valuable insights lost due to lengthy reports.

Widespread **** failures impacting multiple *** services due to overload of Azure ***** system

Azure ***** utilizes two layers of ........ (omit)........ It must be noted that the edge caches do not cache negative responses like **** since the range of these values is infinite. A non-authoritative server like the ***** not reasonably figure out the range of values to cache. ........ (omit)........

Post-Incident Report (PIR)
Retrospective Analysis - Challenges

• Lengthy – avg. 4500 words long.
• Complex – on average, 9 engineers involved in an incident
• Written by many – 34K engineers.
  • Varying degrees of expertise and linguistic styles.
Retrospective Analysis - Challenges

• Error-prone – 20% labelled as ‘Other’.
• Incorrect – 29% labelled incorrectly.
• Incomplete – 58% incomplete labels (e.g., Networking – Other).
Manual Analysis at Microsoft Azure

- Extensive multiple person-year effort.
  - 2051 incidents.
  - 468 services from Microsoft Azure.

- Goals:
  - Identify all the contributing factors behind the incident.
  - Extract key context from the postmortem for each factor.

- Weekly peer review to refine analysis and develop taxonomy of contributing factors.
Manual Analysis At Microsoft Azure - Principles

• Intellectually honest
  • Involve teams and domain experts.

• Focus on depth and breadth
  • Extract all the contributing factors to an incident.

• Actionable findings
  • Lead to creating/updating standards to mitigate future incidents.

• Continuous evolution
  • Learn new factors and evolve the taxonomy.
• 4 contributing factors on average – Contrary to existing work
• Addressing easiest one can reduce incidents!
Manual Analysis At Microsoft Azure - Example

• A service became unavailable after a customer pushed a load that was 60x greater than what the service can handle.

• Contributing factors:
  • Inrush of load from a single customer
  • Lack of throttling on both customer and service ends
  • High CPU, heap usage and thread count led to request failures with exceptions
  • Exception handling of failed request led to resource leaks
  • No automated watchdogs to detect early outage symptoms (or resource leaks)
  • Team cannot access metrics (collocated with service) during the outage.

• Originally chosen label: ‘Service – Load Threshold’
Manuel Analysis At Microsoft Azure – Contributing Factors

- Wide Variety – 346 distinct factors!

<table>
<thead>
<tr>
<th>Category</th>
<th>Frequency</th>
<th>TTM (Hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>61%</td>
<td>50</td>
</tr>
<tr>
<td>Authoring</td>
<td>50%</td>
<td>58</td>
</tr>
<tr>
<td>Dependency</td>
<td>37%</td>
<td>16</td>
</tr>
<tr>
<td>Architecture</td>
<td>20%</td>
<td>33</td>
</tr>
<tr>
<td>Deployment</td>
<td>20%</td>
<td>27</td>
</tr>
<tr>
<td>Process</td>
<td>18%</td>
<td>123</td>
</tr>
<tr>
<td>Load</td>
<td>14%</td>
<td>13</td>
</tr>
<tr>
<td>Auth</td>
<td>7%</td>
<td>21</td>
</tr>
<tr>
<td>Performance</td>
<td>6%</td>
<td>16</td>
</tr>
<tr>
<td>Datacenter</td>
<td>4%</td>
<td>70</td>
</tr>
</tbody>
</table>

https://autoarts-rca-taxonomy.github.io/taxonomy.html
ARTS Taxonomy

• Azure Reliability Tagging System (ARTS) taxonomy to label incidents with contributing factors.

• Visualization: https://autoarts-rca-taxonomy.github.io/taxonomy.html

• Qualities:
  • Hierarchical (4 levels deep)
  • Comprehensive (built from analysis)
  • Unambiguous (clear separation of categories)
But manual labelling is still error-prone!

Our analysis is expensive and cannot scale to all postmortems.
AutoARTS – Automated Root Cause Labelling

A recent code change in service X caused ...

ARTS Taxonomy
- Authoring
  - Code
  - Config
- Change
- Bug
- Latent

AutoARTS
- Context Extraction
- Root Cause Classification

Context
- A recent code change in service X caused ...

Root-Cause Tags
- Authoring, Code, Bug, Change
- Architecture, SPOF, Config
AutoARTS – Root Cause Classification

• Multi-label text classification
  • Noise: Irrelevant details in postmortems
  • Data sparsity: 68% of tags have < 10 postmortems

• Leverage hierarchy in ARTS taxonomy using GCN\(^1\)

• LLMs need large amounts of data to encode text
  • Train custom text encoder w.r.t. taxonomy

Can language models encode postmortems?

- 110K postmortems (20% Test split)
- Poor performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Test Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-uncased</td>
<td>7.57</td>
</tr>
<tr>
<td>BERT-cased</td>
<td>6.69</td>
</tr>
<tr>
<td>XLNet-uncased</td>
<td>23.67</td>
</tr>
</tbody>
</table>
## AutoARTS – Context Extraction Examples

<table>
<thead>
<tr>
<th>Root-Cause Tag</th>
<th>Context from PIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authoring.Code.Bug.Change</td>
<td>SQL team made some recent changes to a gateway component that introduced this regression</td>
</tr>
<tr>
<td>Detection.Validation.MissingTest</td>
<td>NRP test infrastructure doesn't support component tests for standard public IPs.</td>
</tr>
</tbody>
</table>
AutoARTS – Context Extraction

• Extract key context from PIR to justify root cause tags.

• LLMs are good at summarization (abstractive/extractive)
  • But context is not a summary of PIR

• Pegasus\textsuperscript{[1]} is trained for summarization by masking sentences
  • Context sentences should be extracted from PIR
  • Use labelled contexts to finetune Pegasus to extract context from PIRs

\textsuperscript{[1]} PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. ICML'20
AutoARTS – Evaluation

• 1120 labeled PIRs from Microsoft Azure.

• Dataset splits: Train (72%), Validation (8%), Test (20%).
### Which parts of PIR to use?

<table>
<thead>
<tr>
<th>Section</th>
<th>Micro-F1</th>
<th>Weighted-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole PIR</td>
<td>0.55</td>
<td>0.40</td>
</tr>
<tr>
<td>Title</td>
<td>0.53</td>
<td>0.45</td>
</tr>
<tr>
<td>Summary</td>
<td>0.47</td>
<td>0.46</td>
</tr>
<tr>
<td>RC-Details</td>
<td>0.52</td>
<td>0.45</td>
</tr>
<tr>
<td>5-Whys</td>
<td>0.54</td>
<td>0.40</td>
</tr>
<tr>
<td>Discussion</td>
<td>0.53</td>
<td>0.40</td>
</tr>
<tr>
<td>Mitigation</td>
<td>0.47</td>
<td>0.40</td>
</tr>
<tr>
<td>RC-Details + 5-Whys</td>
<td><strong>0.56</strong></td>
<td><strong>0.42</strong></td>
</tr>
</tbody>
</table>

Language models have limits on text sequence length!
Hierarchical structure of ARTS is beneficial for classification!
AutoARTS – Context Extraction

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE</th>
<th></th>
<th></th>
<th>BLEU</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rouge-1</td>
<td>Rouge-2</td>
<td>Rouge-L</td>
<td>BLEU</td>
<td>BLEU-1</td>
<td>BLEU-2</td>
<td>BLEU-3</td>
</tr>
<tr>
<td>Pegasus - Pretrained</td>
<td>32.55</td>
<td>18.72</td>
<td>24.30</td>
<td>9.61</td>
<td>18.03</td>
<td>10.31</td>
<td>8.93</td>
</tr>
<tr>
<td>Pegasus - Finetuned</td>
<td>45.46</td>
<td>35.65</td>
<td>38.43</td>
<td>24.60</td>
<td>32.19</td>
<td>24.98</td>
<td>23.41</td>
</tr>
<tr>
<td>T5 - Pretrained</td>
<td>34.38</td>
<td>23.31</td>
<td>28.03</td>
<td>10.06</td>
<td>15.68</td>
<td>10.83</td>
<td>9.43</td>
</tr>
<tr>
<td>T5 - Finetuned</td>
<td>41.63</td>
<td>33.86</td>
<td>35.76</td>
<td>23.81</td>
<td>29.81</td>
<td>24.10</td>
<td>22.70</td>
</tr>
<tr>
<td>BERT-cased - Pretrained</td>
<td>40.05</td>
<td>27.03</td>
<td>31.01</td>
<td>18.62</td>
<td>28.43</td>
<td>18.95</td>
<td>16.83</td>
</tr>
<tr>
<td>BERT-cased - Finetuned</td>
<td>40.08</td>
<td>27.35</td>
<td>31.20</td>
<td>18.80</td>
<td>28.32</td>
<td>19.03</td>
<td>16.95</td>
</tr>
<tr>
<td>BERT-uncased - Pretrained</td>
<td>39.52</td>
<td>26.58</td>
<td>30.74</td>
<td>17.63</td>
<td>27.47</td>
<td>17.98</td>
<td>15.89</td>
</tr>
<tr>
<td>BERT-uncased - Finetuned</td>
<td>39.92</td>
<td>27.44</td>
<td>31.57</td>
<td>18.64</td>
<td>28.08</td>
<td>18.91</td>
<td>16.90</td>
</tr>
</tbody>
</table>
AutoARTS – User Feedback

• 10 PIRs not previously in evaluation dataset.

• **Metric:** How useful were the AutoARTS generated contexts in identifying all contributing factors?
  • 1 – Not useful at all
  • 5 – Very useful.

• **Response:** 4.6.

• **Metric:** How many contexts were generated with unnecessary details?
• **Response:** 0.
AutoARTS – User Feedback

• **Metric:** How many new root cause labels were you able to identify using the generated contexts?
  • **Response:** 2.

• **Metric:** How many crucial root cause tags were missing from the outputs?
  • **Response:** 7/10.
What AutoARTS is about

**Problem:** Lengthy postmortems, poor root cause taxonomies, error-prone and incomplete root cause labelling.

**Solution:** Develop comprehensive taxonomy, bootstrap labelling postmortems, generate succinct contexts and labels with ML.

**Ideas:** Leverage hierarchy in taxonomy, train text encoders w.r.to tags, finetuning gap sentence summarization.

**Opensource Taxonomy:** Share wide variety of contributing factors with others and develop continuously.
Thank you!

Join Us: https://autoarts-rca-taxonomy.github.io/

Contact: dogga@cs.ucla.edu
http://web.cs.ucla.edu/~dogga