



Differentiable Dynamic Programming for Stack Trace Deduplication

Adem Aber Aouni

adem.aber-aouni@polymtl.ca

École Polytechnique de Montréal
Laboratoire DORSAL

Motivation

- Popular applications can receive high volume of bug reports.
- Mozilla Core:
 - Average of 3 361.15 bug reports per month [4].
 - 24.70% of reported bugs are duplicates [4].
- This slows down the fixing process by a considerable margin [5].



Sequence Matching Based Research

- Basis of several methods : Brodie et al. (2005) [2], Modani et al. (2007) [6], Bartz et al. (2008) [1], Dang et al. (2012) [3], Vasiliev et al. (2020) [7]

https://en.wikipedia.org/wiki/Levenshtein_distance#/media/File:Levenshtein_distance_animation.gif



Proposition

- All methods proposed earlier use hyper-parameters difficult to optimize.
- Most functions used in the similarity methods are differentiable, ex:
 - TraceSim Vasiliev et al. (2020) [7] :

$$\mathbf{w}(f_i) = \mathbf{lw}_\alpha(f_i) \cdot \mathbf{gw}_{\beta\gamma}(f_i)$$

$$\mathbf{lw}_\alpha(f_i) = \frac{1}{i^\alpha}$$

$$\mathbf{gw}_{\beta\gamma}(f_i) = \sigma(\beta(\text{IDF}(f_i) - \gamma))$$



Proposition

- Dynamic programming's min/max functions can be replaced with real soft(min/max) (LogSumExp).
- Hyper parameters can be found optimally for each bug dataset.
- Similarity functions can include neural networks.



DDP - General Idea

- Differentiable Dynamic programming



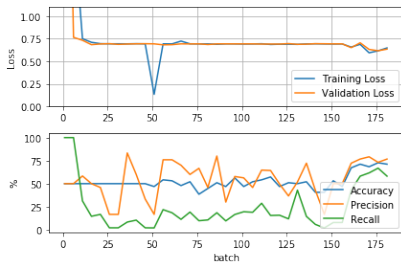
How it works

- Batch together query and candidate pairs.
- Implement a method using only differentiable functions.
- Compute Binary cross entropy loss with ground truth (are the given two stacktraces duplicates).
- Backpropagate error.



TraceSim

- State of the Art (TraceSim) Vasiliev et al. (2020) [7] preliminary results:



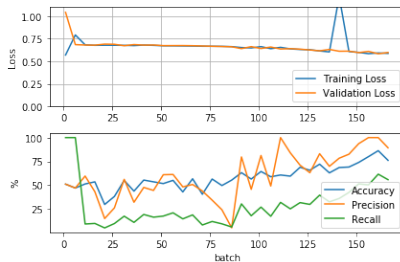
Training of TraceSim using DDP

- Issue: slow training due to high vocabulary (241535 for campbell) .



Word2Vec Embedding

- Use NLP embeddings for stack traces.



Training of TraceSim using DDP and Word2Vec Embedding



Next steps

- Create neural networks for similarity functions.
- Optimize hyper-parameters for state-of-the-art.



Questions

?



- [1] K Bartz, Jw Stokes, Jc Platt, Ryan Kivett, Ryan Kivett Jack W. Stokes Kevin Bartz, Gretchen Loihle Silviu Calinoiu David Grant John C. Platt, K Bartz, Jw Stokes, Jc Platt, and Ryan Kivett. 2008. Finding Similar Failures Using Callstack Similarity. *SysML*.
- [2] Mark Brodie, Sheng Ma, Guy Lohman, Tanveer Syeda-Mahmood, Laurent Mignet, Natwar Modani, Mark Wilding, Jon Champlin, and Peter Sohn. 2005. Quickly finding known software problems via automated symptom matching. *Proceedings - Second International Conference on Autonomic Computing, ICAC 2005*, 2005:101—110.
- [3] Yingnong Dang, Rongxin Wu, Hongyu Zhang, Dongmei Zhang, and Peter Nobel. 2012. ReBucket: A Method for Clustering Duplicate Crash Reports Based on Call Stack Similarity. *2012 34th International Conference on Software Engineering (ICSE)*, 1:1084–1093.
- [4] Jennifer L. Davidson, Nitin Mohan, and Carlos Jensen. 2011.

Coping with duplicate bug reports in free/open source software projects. In *2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 101–108.

- [5] Tejinder Dhaliwal, Foutse Khomh, and Ying Zou. 2011. Classifying Field Crash Reports for Fixing Bugs: A Case Study of Mozilla Firefox. *2011 27th IEEE International Conference on Software Maintenance (ICSM)*, pages 333–342.
- [6] Natwar Modani, Rajeev Gupta, Guy Lohman, Tanveer Syeda-Mahmood, and Laurent Mignet. 2007. Automatically Identifying Known Software Problems. *2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 433–441.
- [7] Roman Vasiliev, Dmitrij Koznov, George Chernishev, Aleksandr Khvorov, Dmitry Luciv, and Nikita Povarov. 2020. TraceSim: A Method for Calculating Stack Trace Similarity. *arXiv*.

