

# Sentence embedding とファインチューニングを通じた重複バグレポート検出

磯谷 春奈<sup>1</sup> 鷲崎 弘宜<sup>1</sup> 深澤 良彰<sup>1</sup>  
野元 励<sup>2</sup> 應治 沙織<sup>2</sup> 斎藤 忍<sup>2</sup>

**概要**：ソフトウェア保守において、重複バグレポートを探すのに時間と手間を多くかけている場合がある。本論文では、ある特定の手法の保守を効率化するための、重複バグレポート検出システムを提案する。提案システムは、レポートの項目ごとの記載内容を Sentence embedding でベクトル化し、その類似度からレポート全体の類似度を計算し、重複レポートを検出する。Sentence embedding には、レポートの文章でファインチューニングした Sentence BERT を使用する。

## 1. はじめに

ソフトウェア保守では、バグレポートでユーザから報告されたバグを修正する。同じバグが複数回報告されることもあり、それらのレポートは重複バグレポートと呼ばれる。新しいバグレポートを受け取ったとき、過去のレポートに重複レポートがないか調べるが、その調査は時間がかかるという問題がある。そこで、その問題を抱える組織の、ある特定の手法の保守に適した重複バグレポート自動検出システムを提案する。

提案システムは、レポートの項目ごとの記載内容を Sentence embedding (文の分散表現) でベクトル化し、その類似度からレポート全体の類似度を計算し、重複レポートを検出する。Sentence embedding には、レポートの文章でファインチューニングした Sentence BERT [1]を使用する。Sentence BERT (SBERT) [1]は、State-of-the-art の Sentence embedding であり、事前学習済みの BERT [2]の出力にプーリング操作を加えたモデルである。BERT は、Attention メカニズムを用いた言語表現モデルである。事前学習により、文脈理解・文の関係性理解を獲得できる。

## 2. 重複バグレポート検出システムの提案

提案システムの目的は、保守担当者が新規問合せ票の同一票 (同じ問題を扱う問合せ票・故障票) を探す手助けをすることである。システムの入力は新規問合せ票であり、出力は、既存の問合せ票・故障票を、入力との類似度が高い順に順位付けしたものである。

まず新規票の表題の項目の記述と内容の項目の記述をそれぞれ SBERT モデルでベクトル化する。そして、新規票の表題ベクトルと、表題用モデルでベクトル化済みの各既存票の表題ベクトルの類似度 (表題類似度) を計算する。内容ベクトルについても同様に、新規票と各既存票の内容ベクトルの類似度 (内容類似度) を計算する。各既存票について、新規票との項目類似度 (表題類似度と内容類似度)

から、票全体としての新規票との類似度 (票類似度) を計算する。最後に、票類似度が高い順に既存票を順位付けして出力する。

モデルは、大規模な文章で学習させた後、問合せ票と故障票の記述で学習させる。既存票の表題の項目の記述を表題用モデルでベクトル化し表題ベクトルとする。既存問合せ票は内容の項目の記述、既存故障票は試験環境と詳細内容の項目の記述を内容用モデルでベクトル化し内容ベクトルとする。項目類似度はコサイン類似度で計算する。

## 3. 評価

提案システムと、提案システムの表題用モデルと内容用モデルを票の記述でファインチューニングしないモデル (大規模文章でのみ学習させたモデル) に置き換えて構築したシステムと、ベクトル化に TF-IDF および LDA を用いるベースラインシステムの性能を評価した結果、提案システムの方が、提案システムの SBERT モデルを票の記述でファインチューニングしないものに置き換えたシステムより性能評価値が高かったことから、票の記述でのファインチューニングが提案システムの性能に寄与することがわかった。項目ごとの処理と、レポートでファインチューニングした SBERT の組み合わせは、本研究で対象とするレポートにおける重複バグレポート検出には有効であった。

票は項目ごとに記載内容が異なるので、使われる表現も異なる。同一票は同じ内容が記載される項目同士で表現が類似するという考えに基づき、項目ごとに類似度を計算したのが有効であったと考えられる。さらに、表題用モデルと内容用モデルでファインチューニングに使う票の項目を変えたことで、それぞれに使われる表現を適切にベクトル化できたと考えられる。また、票によって記入者が異なるため、同じ問題でも、表現や説明が多少異なる。同じ原因の問題でも、ユーザに認識される異常の表れ方がいくらか異なる場合がある。ファインチューニングした SBERT は、それらの差異をうまく吸収することができたと考えられる。

<sup>1</sup> 早稲田大学 Waseda University

<sup>2</sup> 日本電信電話(株) NTT CORPORATION

a システムの構成や詳細、他手法との評価を [11] に詳説している。

## 4. 関連研究

### 4.1 重複バグレポート検出の研究

提案システムと同じように、バグレポートの複数の項目の文章について項目を考慮してベクトル化し、その類似度から重複を判断する重複バグレポート検出手法を紹介する。Rodrigues らは、バグレポートの断定的な要素（製品情報や優先度など）と文章部分それぞれをベクトル化して統合し、重複の確率を出力する手法を提案した[3]。比較対象のレポートによってベクトルが変わる点が、レポートの文章のベクトル化における提案手法との大きな違いである。

Deshmukh らは、バグレポートの断定的な要素と要旨と説明をそれぞれ別のモデルで符号化し、各符号を連結してレポートのベクトルとし、類似度から重複を判断する手法を提案した[4]。Deshmukh らの手法では要旨を bi-LSTM、説明を CNN でベクトル化するのに対し、提案システムでは長距離依存性の学習が可能な Attentive neural network でベクトル化するので、提案システムでのベクトル化の方が、より文脈を反映したベクトルを作成できると考えられる。

提案システムと同じように、バグレポートをベクトル化してその類似度から重複を判断するが、バグレポートの 1 つの項目の文章のみをベクトル化する、あるいは複数の項目の文章について項目を考慮せずベクトル化する重複バグレポート検出手法を紹介する。Kukkar らは、バグレポートの文章を CNN でベクトル化し、その類似度から重複を判断する手法を提案した[5]。Xiao らは、Heterogeneous information network でバグレポートを表し、レポートとその断定的な要素のノードをベクトル化したものと、レポートの要旨の記述を RNN でベクトル化したものを統合し、その類似度から重複を判断する手法を提案した[6]。Akilan らは、バグレポートをトピックモデリングによってクラスタリングし、比較するレポートを絞ってから、レポートの文章のベクトルの類似度で重複を判断することで、高速な検出を実現する手法を提案した[7]。Sehra らは、バグレポートの文章を 4 つの Embedding を用いて 4 種類にベクトル化し、4 種類のベクトルそれぞれの類似度から 1 つの類似度を計算して重複を判断する手法を提案した[8]。

### 4.2 BERT のバグレポートへの適用事例

ドメイン知識でファインチューニングすることで、BERT はドメイン固有の表現も適切に処理できるので、ソフトウェア開発・保守に関係する文書を、BERT を用いて分析する研究が盛んにおこなわれている。その中で、BERT でバグレポートを分析する研究を紹介する。Ardimento らは、バグレポートの文章を、出力層として分類器を付け加えた BERT に入力し、そのバグの修正にかかる時間が長いかわいかな予測する手法を提案した[9]。Hirakawa らは、OSS を組み合わせて構築されたシステムのバグレポートについて、レポートの文章を、出力層として分類器を付け加えた

BERT に入力し、どの OSS のバグかでレポートを分類する手法を提案した[10]。筆者が知る限りでは、本研究は BERT を用いて重複バグレポート検出を行う初の研究である。

## 5. おわりに

本論文では、ある特定の保守のための重複バグレポート検出システムを提案した。今後、項目ごとに処理し他のベクトル化手法を用いるシステムと、項目を考慮せずレポートの文章でファインチューニングした SBERT を用いるシステムとも性能比較を行うことが課題である。また、提案システムでの重複バグレポート検出の手法を一般化することで、本研究で扱う保守とは異なる方法の保守が行われている製品のバグレポートへの手法の適用が期待できる。そして一般化した手法を、重複バグレポート検出の State-of-the-art の手法と性能比較したい。さらに、製品の構造を考慮することで、性能向上が期待できる。

**謝辞:** 本研究の取り組みにあたりご支援いただいた小泉雄太氏、徐振宇氏、金丸朗氏、村本達也氏に御礼申し上げます。

## 参考文献

- [1] Reimers, N. and Gurevych, I., Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.
- [2] Devlin, J. et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), 2019.
- [3] Rodrigues, I.M. et al., A Soft Alignment Model for Bug Deduplication. 17th International Conference on Mining Software Repositories (MSR), 2020.
- [4] Deshmukh, J. et al., Towards Accurate Duplicate Bug Retrieval Using Deep Learning Techniques. IEEE International Conference on Software Maintenance and Evolution (ICSME), 2017.
- [5] Kukkar, A. et al., Duplicate Bug Report Detection and Classification System Based on Deep Learning Technique. IEEE Access, 2020, vol. 8.
- [6] Xiao, G. et al., HINDBR: Heterogeneous Information Network Based Duplicate Bug Report Prediction. 31st IEEE International Symposium on Software Reliability Engineering (ISSRE), 2020.
- [7] Akilan, T. et al., Fast Detection of Duplicate Bug Reports using LDA-based Topic Modeling and Classification. IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020.
- [8] Sehra, S.S. et al., Amalgamated Models for Detecting Duplicate Bug Reports. 33rd Canadian Conference on Artificial Intelligence (Canadian AI), 2020.
- [9] Ardimento, P. and Mele, C., Using BERT to Predict Bug-Fixing Time. 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), 2020.
- [10] Hirakawa, R. et al., Study on Automatic Defect Report Classification System with Self Attention Visualization. 2020 IEEE International Conference on Consumer Electronics (ICCE), 2020.
- [11] Isotani, H. et al., Duplicate Bug Report Detection by Using Sentence Embedding and Fine-tuning, 37th IEEE International Conference on Software Maintenance and Evolution (ICSME), Industry Track, pp.1-10, 2021.