

提案データのクラスタリングによる類似提案の提示システム

菅野 久樹[†] 新美 礼彦[†]

公立はこだて未来大学システム情報科学部[†]

1. 背景と目的

ソースコード共有サイト GitHub[1]では多くのオープンソースソフトウェアが開発されており、特にユーザから注目されている多くのオープンソースソフトウェアプロジェクトには日々多くの提案が投稿されている。投稿される提案の中には無益なものも多数含まれており、プロジェクトオーナーはその対応に多大なコストがかかるという問題がある。

檀上ら[2]は、GitHub の 3 つのオープンソースソフトウェアプロジェクトに投稿された提案が有益であるか無益であるかを調査した。檀上らは無益な提案を、以下のいずれかに該当するものであると定義した。

- (1) 過去の提案と内容が重複している
- (2) 提案者のミスが原因の不具合の報告
- (3) プロジェクトで扱うべきでない問題や機能の報告

人気のプロジェクト 3 件を調査した結果、すべての提案に占める無益な提案の割合がいずれのプロジェクトでも 50%を超えていることが示された。

提案者が無益な提案を行ってしまう原因の 1 つとして、過去の提案を読んでいないということがある。提案の検索機能を利用して適切な提案を探すことができるが、提案の取り扱い方法がプロジェクトごとに異なるため、過去の提案を探すことが困難である。

本研究では、提案者が過去の提案を探すことが困難であることに着目し、投稿しようとしている提案と類似した過去の提案を提案者に提示するシステムを構築することで、無益な提案の投稿を予防し、プロジェクトオーナーの負担を軽減することを目的とする。

2. GitHub

GitHub は GitHub 社が運営するソースコード共有サイトである。Git のホスティングサービスだけでなく、気になるアカウントの更新通知を受け取れるなどの SNS 的な要素も備えている。

プロジェクトオーナーはリポジトリのソースコードへの変更権限を持つユーザのことである。提案をプロジェクトへ取り込むかどうかの意思決定はプロジェクトオーナーが行う。複数のユーザがプロジェクトオーナーである場合もある。

Issue はバグトラッキングシステムにおけるチケットのような機能である。提案者は追加してほしい機能やバグ報告ごとに Issue を発行する。Issue の投稿は GitHub ユーザであれば誰でも行うことができる。Issue にはコメント機能があり、掲示板のスレッドのような形でコメントのやりとりを行うことができる。コメント機能を利用して、提案についての議論が行われる。ラベルと呼ばれる Issue を分類するための文字列を付与することができる。ラベルを Issue の検索に使用することができる。テキストとしての Issue の特徴として、Markdown 記法に用いられる記号が多く含まれていること、特定のソフトウェア、プログラミング言語などの技術用語が多く含まれていることが挙げられる。本研究では、この Issue を提案と呼ぶ。

3. 提案手法

大蔵ら[3]によって、ソフトウェア開発に使用された E メールのクラスタリングが分析の糸口として有用であることが示されている。本研究では提案者に似た話題の提案を提示するために、過去の提案を話題ごとにクラスタリングし、新たに投稿しようとする提案と内容が最も近いクラスタの提案を提示するシステムを提案する。ラベルを元にした分類を提示する手法も考えられるが、Issue の話題を表すラベルの使用方法がプロジェクトごとに異なること、同じプロジェクト内であってもラベルの使用方法が徹底されていないことから、事前に Issue の話題を分類することが困難であると考え、クラスタリングを用いることとした。

提案するシステムは一定期間ごとに Issue の収集、及び収集した Issue のクラスタリングを実行し、クラスタリングモデルを最新の状態に更新する。提案者が新しい Issue をシステムへ投稿すると、システムはクラスタリングモデル

Similar Proposals Summarizing System by Proposal Clustering
[†]Kanno Hisaki, Niimi Ayahiko, Future University Hakodate

表 1 各クラスタリングアルゴリズムのエントロピー平均値

単純連結法	完全連結法	平均法	重心法	メディアン法	ウォード法	Mcquitty 法
0.081	0.581	0.113	0.069	0.074	0.671	0.216

に従って投稿された新しい提案と最も近いクラスタを判定し、提案者にそのクラスタを提示する。

4. 実験と評価

提案手法の実験として、Issue のテキストをクラスタリング可能であることを検証した。Issue のテキストを単語出現頻度ベクトル化し、TFIDF による重み付けを行った。そのベクトルに対してクラスタリングを行った場合に、近しい話題を同じクラスタとして判定できるかを評価した。クラスタリング手法として階層的クラスタリングの複数のアルゴリズムを用い、エントロピーを評価基準とした。エントロピー算出のための正解データとなるクラスラベルとして Issue に付与されているラベルを用いることとした。

Issue 数の違いとクラスタリング結果の違いを比較するため、実験には 21 件のオープンソースソフトウェアプロジェクトのデータを使用した。そのうち、Issue 数が最大のプロジェクトは 3862、最少のプロジェクトは 20 であった。

実験から、平均値で表 1 に示すようなエントロピーが得られた。

5. 考察

Issue のような特徴を持つテキストを単語頻度ベクトル化してクラスタリングすることについて、及びエントロピーが低くなる条件について考察する。

Markdown 記法で用いられる記号がクラスタリング結果に影響を及ぼす可能性があったが、記号を予め削除した場合と削除しなかった場合で結果を比較したところ、エントロピーに差は生じなかった。Markdown 記法で用いられる記号は各 Issue で横断的に使用されており、TFIDF による重み付けによって、Issue の意味を表現しうる重要な単語ではないとされたため、記号を削除した場合と削除しなかった場合でのクラスタリング結果に差が生じなかったものと考えられる。単語頻度ベクトルと TFIDF による重み付けは Issue のテキストに対して有効であった。

図 1 は重心法を用いた場合の Issue 数とエントロピーの比較である。エントロピーは Issue 数が多いほど減少する傾向にあり、総 Issue 数が多いプロジェクトへクラスタリングを適用するのがより効果的であると考えられる。

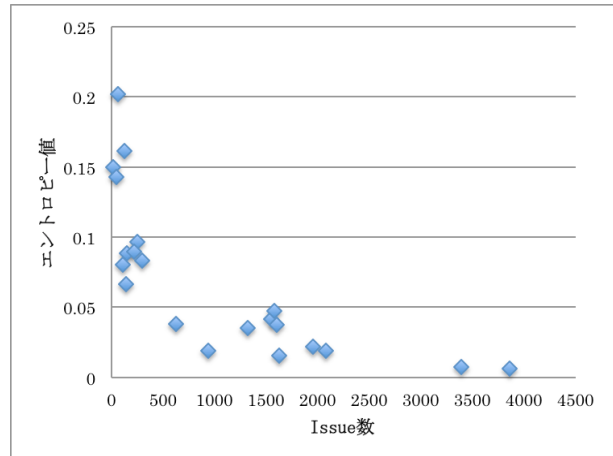


図 1 Issue 数とエントロピーの比較

一般的には正解クラスタの数が多いほどエントロピーは低くなるが、実験ではラベル数とエントロピーには相関が見られなかった。

6. まとめ

本研究では無益な提案の投稿を予防するためにクラスタリングを用いて過去の類似した Issue を提案者に提示するシステムを提案した。提案システムの有効性の検証のためエントロピーを評価基準として Issue のテキストのクラスタリング実験を行った。実験の結果から、Issue のテキストは TFIDF による重み付けを行った単語頻度ベクトルを用いたクラスタリングが可能であることが確認できた。また、Issue の数が多いほどエントロピーが低かったことから、提案システムは Issue 数が多いプロジェクトへ適用するのがより効果的であることを確認した。

参考文献

- [1]GitHub, (online), <<https://github.com/>> (2014.12.29)
- [2]檀上 未来, 乃村 能成, 谷口 秀夫: ソーシャルコーディングにおける有益な提案の抽出について, 情報処理学会研究報告, Vol. 2013, No4, pp. 1-8 (2013).
- [3]大蔵 君治, 川口 真司, 飯田 元: Eメールアーカイブのクラスタリングによる開発コンテキストの可視化, SEC journal 6(3), pp. 134-143 (2010).