

Sentiment and Politeness Analysis Tools on Developer Discussions Are Unreliable, but so Are People

Nasif Imtiaz, Justin Middleton, Peter Girouard, and Emerson Murphy-Hill

背景

- ソフトウェア開発の協調的な性質で、開発者のもつ気分によって互いにパフォーマンス等に影響を及ぼすことが明らかになり、ソフトウェア工学における感情を研究することが増えている
- 感情およびポライトネス分析ツールの信頼性がいまだに大きな疑問があるにも関わらず、多くの研究で使用されている.
- Many software engineering researchers use **sentiment and politeness analysis tools** to study the emotional environment within collaborative software development. However, papers that use these tools **rarely establish their reliability.**

目的

- In this paper, we evaluate popular existing tools for **sentiment and politeness detection** over a dataset of 589 manually rated GitHub comments that represent developer discussions

したこと

- We also develop a coding scheme on **how to quantify politeness** for conversational texts found on collaborative platforms

対象データ群

- 開発者の議論を示すGitHubのコメントをラベル付け
 - 2人以上いても合意が大幅に変わることはない[26]
 - 2人で600コメントの注釈をつける
 - 慎重なステップとして40追加(??)
- 640から無作為に20選んだ時全て異なるプロジェクト発のもの
 - 多くを網羅できている
- 不完全のフラグメント, 重複, コメントがない, 英語以外のコメントを含む, 無効なURLであるなどを除外すると589が残る
- <https://github.com/DeveloperLiberationFront/AffectAnalysisToolEvaluation>
でデータセット, コード体系, その他関連ファイルを公開

Human coder

- データサイエンス学部の大学院生, 学部生
- Coder1 : すべてのコメントを評価
- Coder2,3 : 半分を評価

実験の流れ

- **First dataset**

分類→合致→”agree set”(丁寧さ374件, 感情285件)

- 意見交流(the newly negotiated set)

- **Final dataset**

= agree set + the newly negotiated set

RQ

- RQ1.

How consistent are **human coders** to rate sentiment and politeness on GitHub comments?

- RQ2.

How reliable are **sentiment analysis tools** on GitHub comments?

- RQ3.

How reliable are **politeness analysis tools** on GitHub comments?

Human coder - sentiment -

- コメントをpositive, negative, neutral, mixed or sarcasm(皮肉)の4つに分類
- Our tools label texts as "neutral" where opposite emotions counter each other.
反対の意見を両方持っている場合は中立としてみなす
→学生がmixedとしたものを最終的にはneutralに筆者らがした
- 皮肉と評価されたテキストを省いた
→これやってほしいんだが
- there is **no clear guideline** on how to match its label with the possible ratings the tools have as output.

Human coder - politeness -

- 例に従って分類
- 例になかったものや文脈が変である箇所等はコメント
- 非常に丁寧, 丁寧, 普通, 失礼, 非常に失礼の5段階評価
→結局, 3段階で判断する
- To measure the degree of politeness/impoliteness,
- the coders were suggested to consider **the frequency of strategies** used in the comment according to the scheme alongside their own judgment and the underlying context.

ツールの選択

- SentiStrength、NLTK、Alchemy、Stanford NLPはソフトウェア工学とは関係のないコーパスで訓練されている
- 開発者ディスカッションに関連するデータセットで訓練された2つのツール(Sentit4SD, SentiCR)を使用

Politeness Scheme

- テキスト文書を評価するために一般的に用いられている politeness schemeがないので開発した.
- Positive face
好かれる, 感謝される, 承認されるなどの欲求に向かっている
- Negative face
課される, 侵入される, 身につけられたりしない欲求

判定方法

- データセットが中立的な意見で偏っているので、F値のみでは**偶然の一致の結果**と考えられる
→ カッパ係数を用いる

人同士

Table 2: Inter-rater Agreement (Weighted Cohen's Kappa)

Weighted κ	Coder 1 and Coder 2	Coder 2 and Coder 3
Sentiment	.38	.27
Politeness	.48	.36

- politenessの方が若干一致度が高い
→ラベル付けを行う前の例題がより詳細であったため

各ツールの精度

Table 3: Performance of Sentiment Analysis Tools for the First Set of Agreed Data

		F-measure		
tools	Weighted κ	neutral	positive	negative
SentiStrength	0.44	74.58%	61.54%	40.68%
NLTK	0.27	52.63%	55.42%	23.73%
Alchemy	0.33	58.78%	58.21%	32.65%
Stanford NLP	0.26	55.85%	59.83%	19.44%
Senti4SD	0.53	84.92%	68.18%	41.03%
SentiCR	0.13	57.04%	29.70%	29.91%

- 否定的な意見の精度が低い
- 中立の技術文書を感情的に否定的なものとして誤って分類していることの裏付けになる. [3,6,28]

各ツールの精度

Table 4: Performance of Sentiment Analysis Tools for the Final Set of Data

		F-measure		
tools	Weighted κ	neutral	positive	negative
SentiStrength	0.24	65.68%	43.01%	30.85%
NLTK	0.24	45.38%	45.26%	33.86%
Alchemy	0.23	51.56%	41.67%	34.48%
Stanford NLP	0.16	43.05%	45.09%	26.41%
Senti4SD	0.33	79.04%	50.47%	31.25%
SentiCR	0.24	64.87%	43.48%	31.02%

- SentiCRを除くツールが全体的に悪化
- 否定的な意見の精度が相変わらず悪化

結果

- we find that not only **the popular existing tools are unreliable**, even **humans too are inconsistent** in identifying sentiment and politeness in developer discussions.
- This demonstrates **the need for standardized coding schemes for the human coders** in order to build an oracle and then perform customized training on the tools to perform reliable affect analysis in the software engineering domain.

Discussion

- コーダーは暗黙的な丁寧さ/失礼さを，根本的な文脈を知らずに単一のコメントから明確に導き出すことはできないことに合意
- 曖昧さの少ないステップを持つコーディング体系を開発
- テキスト自体の中に明示的な丁寧さの記述があるのはできる
-

Discussion

- どうしてツールが悪い結果をもたらしたのか
→ ツールと人間の評価が異なるコメントに注目した
- ツールが否定的, 人間が中立的
errorやproblemはそれらを中立としてラベルしたから
- ツールが中立的, 人間が肯定的
感謝, おめでとう, LGTM, 絵文字のような単語にはツールは十分は重みを持たない可能性がある
- 複数の意味がある文では最終的になにが言いたいのかを捉えることができていない
- 短いテキストと適切な語彙の重みに関する問題がある
- direct orders (e.g. "Do not add", "see above", "replace")
ツールによっては中立である
- 明確な失敗例があるということは改善が可能であることを示唆している

制約

- Human coderが各コメントにつき2人
- データセットの半分ずつを評価している
→人間の偏りがある

関連研究

気分はパフォーマンスに影響を与える

- プログラミングには様々な形態の認知タスクが含まれるので、プログラマーの気分は彼らの活動やパフォーマンスに影響を与える [22]

気分は伝染する

- 協調的な性質は開発者の気分が他の開発者の影響を受ける可能性があることを示す.
- 組織的な出来事→感情的な反応を引き起こす→パフォーマンスや仕事の満足度に影響を与える [33]

どこでも使えるのか

- Researchers who focus on sentiment analysis have provided **strong data that the tools are less effective when applied to new domains** they were not trained on [14, 28].
- Novielli [28]
既存のツールのドメイン依存性を指摘
→それらの新しいコーパスに適応することが困難

ツールの分析

- Jongeling and colleagues [21]
- JIRAレポジトリからデータに対して異なる感情分析ツールを使用すると、異なる結果が得られる。
→様々なツールがどのように異なる感情の結果をもたらすかを示している。
- ツールの信頼性を理解することは、研究する際にこれらのツールを使用しても良いのか否かを知るに役立つ

感情分析ツールに関して

- Guzmanら
コミットコメントを研究するためにセンチメント分析を適用[18]
- Pleteaら
セキュリティ関連の議論と変動する感情の間の相関関係を見つけることを試みた[34]
- GentooコミュニティやStack Overflowなどの他のプラットフォームも開発者の感情を理解するために研究されている[15,19,20,27]
- ツールをドメイン特化にカスタマイズする研究もある[1,6]
→ラベル付けを行う際に異なる or 全く**コード体系**を使わなかった
?

ポライトネス分析に関して

- Politeness can be described as "the practical application of good manners or etiquette" [7]
Politeness is a strong factor in social collaborations [30, 42]
- Ortu [30]
丁寧な開発者であればあるほど、問題解決にかかる時間が長い
- Tsayら [41]
提出者の以前の交流基準が開発の貢献度や、提案された代替案の性質についてどれくらい丁寧に議論しているのかに変化した。
→プロジェクトに貢献するかしらないか、どこに貢献するかは上記の要因で判断している？
- However, similar to sentiment analysis, different studies had different methods to rate politeness [4, 41].