

AFFECT ANALYSIS

Jonathan Dreyer

Horia Mut

Valentin Py

SOMMAIRE

- Problématique
- Solution proposée
- Application développée
- Implémentation et tests
- Démonstration de l'application

PROBLÉMATIQUE

- Projets gérés avec un système de version (git,...):
 - Modifications: Pull requests, composées de plusieurs commits
 - Modifications difficiles à vérifier
 - Nombreux changements
 - Programme complexe
- Comment faciliter le travail du “maintainer” ?
 - Donner des informations sur la qualité de la modification (PR)

SOLUTION PROPOSÉE

- Classification des Pull Requests:
 - Utilisation des réactions relatives aux commits / PR
 - Développeur ayant proposé la modification
 - Autres développeurs
 - Reviewers
 - Réactions sous la forme d'émoticônes
 - Ajout de labels à la pull request

APPLICATION DÉVELOPPÉE

- Application Python
 - Connexion à un compte Github (Username & Token)
 - Choix d'un repo
 - Polling sur les Pull Requests
 - Analyse des Pull Requests:
 - Emotions (émoticônes) => positif ou négatif ?
 - Attribution d'un label

IMPLÉMENTATION ET TESTS

- Application Python
- Pipeline CI / CD
 - Basé sur Docker
 - Tests unitaires
 - Travis CI
 - Jenkins

PROBLÈMES / AMÉLIORATIONS

- Problèmes et défauts:
 - Peu de métriques
- Améliorations possibles:
 - Contexte des émotions: machine learning
 - Analyse du code source
 - Utilisation de webhook (API)

CONCLUSION

- Implémentation fonctionnelle du concept
 - Vérification de chaque pull request
- Résultats bons si les émoticônes sont utilisées
- Améliorations
- Bonne démonstration de l'utilisation de pipelines

DÉMONSTRATION