unesp

Clusterização de commits e análise de sentimento para a exploração de repositórios de software

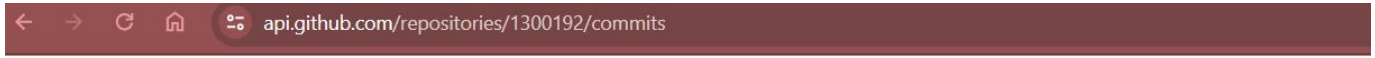
Disciplina: Mineração de Dados
Aluna: Bianca Puerta Rocha Vieira
Professora: Verônica Oliveira de Carvalho

Roteiro

- Introdução
- Trabalhos Relacionados
- Referencial Teórico
  - Mineração de Repositório de Software (MRS)
  - Coleta de dados
  - Pré-processamento
  - Mineração de texto
  - Pós-processamento
  - Análise de sentimento
- Metodologia
  - Aquisição de dados
  - Pré-processamento realizado
  - Construção do modelo de clustering
  - Pós-processamento
  - Análise de sentimento
- Análise dos Resultados
- Conclusão

Introdução

- Cada vez mais têm sido estudados métodos para analisar os dados presentes nos repositórios de software [1];
- A mineração de dados é uma alternativa para realizar estudos e criar modelos baseados em dados extraídos dos repositórios;
- GitHub é a melhor plataforma para conseguir dados sobre os repositórios [2];
- Commits armazenam metadados sobre modificações no código junto a um comentário;
- Mensagens de commit possuem dados importantes sobre o desenvolvimento e sentimento [3].




```
[
  {
    "sha": "d0dd1f61b33d64e29d8bc1372a94ef6a2fee76a9",
    "node_id": "MDY6Q29tbWl0MTMwMDE5MjpkMGRkMWY2MWIzM2Q2NGUyOWQ4YmMxMzcyYTk0ZWY2YTJmZWU3NmE5",
    "commit": {
      "author": {
        "name": "The Octocat",
        "email": "octocat@nowhere.com",
        "date": "2014-02-12T23:20:44Z"
      },
      "committer": {
        "name": "The Octocat",
        "email": "octocat@nowhere.com",
        "date": "2014-02-12T23:20:44Z"
      },
      "message": "Pointing to the guide for forking",
      "tree": {
        "sha": "d7cee29eaada459ba458a63ad983a89915c6a10a",
        "url": "https://api.github.com/repos/octocat/Spoon-Knife/git/trees/d7cee29eaada459ba458a63ad983a89915c6a10a"
      },
      "url": "https://api.github.com/repos/octocat/Spoon-Knife/git/commits/d0dd1f61b33d64e29d8bc1372a94ef6a2fee76a9",
      "comment_count": 204,
      "verification": {
        "verified": false,
        "reason": "unsigned",
        "signature": null,
        "payload": null
      }
    },
    "url": "https://api.github.com/repos/octocat/Spoon-Knife/commits/d0dd1f61b33d64e29d8bc1372a94ef6a2fee76a9",
    "html_url": "https://github.com/octocat/Spoon-Knife/commit/d0dd1f61b33d64e29d8bc1372a94ef6a2fee76a9",
    "comments_url": "https://api.github.com/repos/octocat/Spoon-Knife/commits/d0dd1f61b33d64e29d8bc1372a94ef6a2fee76a9/comments",
    "author": { … }, // 18 items
    "committer": { … }, // 18 items
    "parents": [ … ] // 1 item
  },
```

Introdução

- Este trabalho tem como objetivo geral realizar o agrupamento dos commits e a comparar sentimentos obtidos a partir de metadados dos commits dentro dos grupos obtidos (que são tidos como as atividades) e também com outros dados sobre commits e repositórios;
- RQ1: as linguagens têm uma influência na distribuição das proporções dos grupos?
- RQ2: o tipo de atividade de desenvolvimento influencia os sentimentos expressos nos comentários?
- RQ3: Os sentimentos expressos em comentários de commits refletem na popularidade do projeto entre desenvolvedores?
- RQ4: As linguagens de programação mais utilizadas têm influência nos sentimentos expressos nos comentários?
- RQ5: as atividades mais recorrentes dentro do projeto tem alguma relação com a popularidade dos repositórios?

Trabalhos Relacionados

- O trabalho de Zafar et. al (2019) tem como objetivo um estudo para aumentar a acurácia da rotulação de commits baseados nas mensagens/comentários dos commits. O foco da classificação é dizer se o commit é de bug-fix ou não [1];
- Ji et. al (2018) usam as mensagens de *commits* para detectar os *commits* de correção (*fix*) por meio de palavras chaves que geralmente estão presentes nesse tipo de mensagem de modificação;
- Meng et. al (2021) usam as *convolutional neural networks* (CNN) para classificar commits considerando as modificações registradas nos *commits* e suas relações. O método é aplicado no estudo em 5 repositórios Java de código aberto. Basicamente a classificação é feita entre as classes: *commits* de ajuste (*bug-fix*), inserção de funcionalidade e outros (que não se encaixam nas duas classes anteriores).

Referencial teórico

Referencial Teórico

Mineração de Repositório de Software

- A Mineração de Repositório de Software (MRS) é uma área pesquisa que dá suporte a melhoria do processo de desenvolvimento de software [5];
- A MRS é a aplicação dos conceitos de mineração, descobrimento de padrões e/ou regras associativas em um repositório de software;
- Importante realizar análise de dados e de atividade dos projetos, além as informações coletadas sobre projetos relacionados (mesmo *framework*, público alvo ou nicho de mercado) [6].

Referencial Teórico

Coleta de dados

- O BigQuery é o data warehouse gerenciado pela Google e sem servidores. É um PaaS (Platform as a Service) que permite análises escalonáveis de imensas quantidades de dados por meio de consultas SQL;
- GitHub disponibiliza base de dados pública no BigQuery;
- A coleta também pode ser feita pela própria API disponibilizada pelo GitHub.

Referencial Teórico

Pré-processamento

Pode ser a fase mais demorada e meticulosa da mineração de dados - **pré-processamento de dados textuais.** Alguns passos de acordo com [7][8]:

- **Remoção de Ruído**
  - Remoção de Caracteres Especiais: Remova caracteres especiais, como pontuação, símbolos e números, que podem não ser relevantes para a análise de texto;
  - Remoção de Espaços em Branco Extra: Limpe espaços em branco extras, como espaços, guias e quebras de linha.
- **Tokenização**: Divida o texto em palavras ou tokens individuais. Para contagem de palavras ou modelagem de tópicos;
- **Conversão para Minúsculas**: Padronize todas as palavras para letras minúsculas para garantir que as palavras em maiúsculas e minúsculas sejam tratadas da mesma forma;
- **Remoção de Stop Words;**
- **Stemming e Lematização**: Reduza as palavras às suas formas básicas, removendo sufixos e prefixos. Isso ajuda a reduzir as palavras a um formato padrão (por exemplo, "correndo" -> "corre") para evitar a duplicação de palavras semelhantes;
- **Normalização**: palavras erradas ou que não são na língua original;
- **Remoção de Duplicatas;**
- **Feature Engineering**: criar características adicionais, como contagem de palavras, Utilizar TF-IDF (Term Frequency-Inverse Document Frequency)

Referencial Teórico

Pré-processamento

- No TF-IDF, cada palavra recebe um peso de acordo com sua importância na frase/documento;
- TF e a frequência de uma palavra *p* em um documento *d*;
- *log* para reduzir a frequência bruta;
- O IDF é a inversa da frequência da palavra no documento, assim balanceando pesos de palavras que aparecem muitas vezes.

a) $TF\_IDF = TF * IDF$

b) $TF(p, d) = log_{10}(count(p, d) + 1)$

c) $IDF(t) = log_{10}(\frac{count(D)}{df(t)})$

Fonte: Moser et al. (2022)[9]

Referencial Teórico

Mineração de texto

Abordagens utilizadas

- **Aprendizado não supervisionado** - *Unsupervised Learning*: métodos para tentar encontrar estruturas ocultas nos dados. Não precisa de treinamento anterior. Um exemplo e a clusterização, que divide os registros em grupos;
- **Mineração de opinião e análise de sentimento** - *Opinion Mining and Sentiment Analysis*: encontrar opinião dentro dos dados e sentimentos negativos ou positivos os quais o texto expressa.

Referencial Teórico

Mineração de texto

De acordo com [7]:

- Clusterização é um meio de agrupar os documentos que não possuem labels;
- Quando se usa **documentos**, a intenção é agrupar documento que tem mesmo assunto;
- Ao utilizar **parágrafos ou sentenças**, se analisa as sentenças de diferentes documentos da mesma fonte;
- Ao analisar **palavras ou termos**, são agrupadas palavras associadas ao mesmo tema

Referencial Teórico

Mineração de texto

Tipos de Algoritmos:

- **Hierárquico**: Esse tipo de algoritmo desempenha melhor quando existe uma hierarquia dentro do grupo de dados que é analisado [10];
- **Particionamento**: Definição de hiperparâmetro K (número de clusters). Calculando a proximidade dos pontos ao redor do centroid (definido aleatoriamente). Essa abordagem diminui o custo de processamento pois possibilita menos iterações [11];
- **Baseado em densidade**: Nesta abordagem são consideradas as regiões com maior densidade para a criação de clusters. Definição de hiperparâmetros de mínimo de pontos para formar um cluster e distância máxima dos pontos [11];

Referencial Teórico

Pós-processamento

- **Validação do modelo** -
  - **Índice de silhueta** - mede a qualidade do agrupamento com base na distância média entre os objetos intra-cluster (varia de -1 a 1) [12];
  - **Índice de Davies-Bouldin** - tem como objetivo medir o quão bem os clusters estão separados uns dos outros - um valor mais baixo indica uma melhor qualidade de agrupamento [13];
  - **Índice Calinski-Harabasz** - avalia a qualidade de agrupamentos de dados com base na relação entre a variância entre clusters e a variância dentro de clusters - quanto maior o valor do índice, melhor a qualidade do agrupamento [14].
- **Visualização da estrutura** - a visualizar e interpretar os padrões encontrados após a mineração de dados;
  - visualização dos clusters por meio do PCA;
  - análise das palavras-chave dos clusters

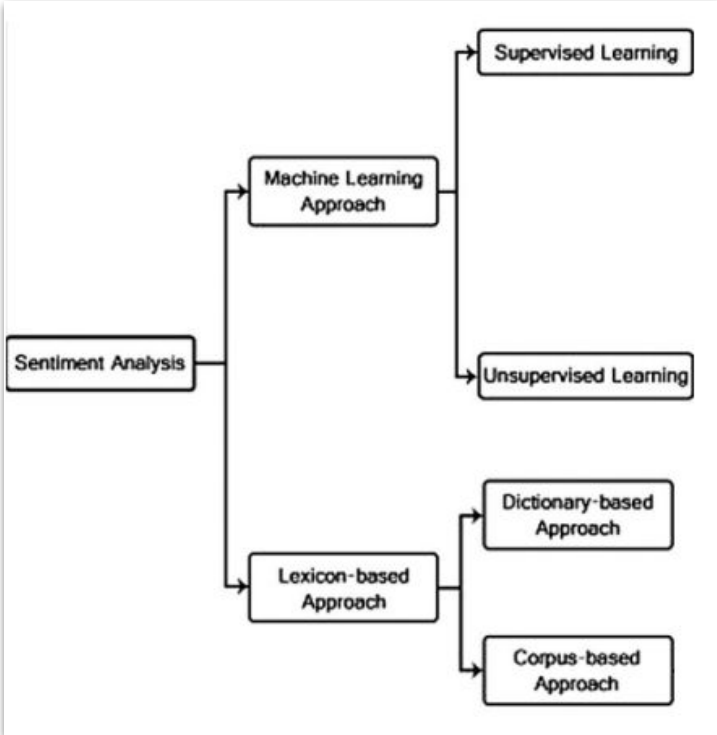
Referencial Teórico

Análise de sentimento

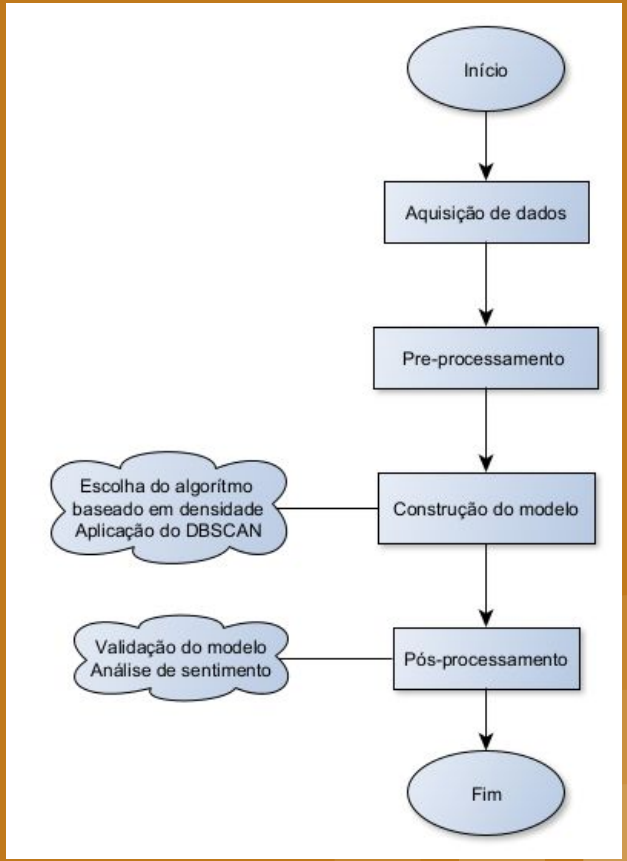
De acordo com [15]:

- **Classificação de sentimento**: negativo, positivo e neutro;
- **Classificação de subjetividade**: detectar se a sentença e subjetiva ou não;
- **Resumo de opinião**: extração das features principais que uma entidade compartilha com as outras;
- **Recuperação de opinião**: retornar um grupo de registros que expressam opinião a respeito de um tópico específico;
- **Sarcasmo e ironia**: foco em captação de registros que expressam ironia ou sarcasmo;

Fonte: Adaptado de Serrano et al. 2015 [16]



Metodologia



Metodologia

Aquisição de dados

```sql
SELECT
    lang, COUNT(*) repos,
    ARRAY_AGG(STRUCT(name, stars) ORDER BY stars DESC LIMIT 20) repo
FROM (SELECT
        repo.name,
        MAX(CAST(JSON_EXTRACT_SCALAR(payload, '$.pull_request.base.repo.stargazers_count')AS INT64)) stars,
        JSON_EXTRACT_SCALAR(payload, '$.pull_request.base.repo.language') lang
      FROM `githubarchive.month.201912`
      GROUP by repo.name, lang)
where lang is not null and lang not in ('HTML', 'CSS', 'TypeScript')
GROUP BY lang
ORDER BY repos DESC
limit 5
```

Fonte: Autores

```sql
SELECT message,
 repo_name[0],
 author.name,
 committer.name,
 subject
FROM `bigquery-public-data.github_repos.commits` WHERE repo_name[0] in
('freeCodeCamp/freeCodeCamp','vuejs/vue','facebook/react','twbs/bootstrap',
'airbnb/javascript','facebook/react-native','facebook/create-react-app',
'axios/axios','nodejs/node','FortAwesome/Font-Awesome',
'trekhleb/javascript-algorithms','mrdoob/three.js','puppeteer/puppeteer',
'30-seconds/30-seconds-of-code','mui-org/material-ui','jquery/jquery',
'webpack/webpack','atom/atom','hakimel/reveal.js','socketio/socket.io',
'rails/rails',...)
```

Metodologia

Aquisição de dados

	lang	repos	repo_name	stars	message	project	name	author	subject
1	JavaScript	387469	vuejs/vue	154667	release: v2.7.14\n	vuejs/vue	Evan You	Evan You	release: v2.7.14
2	JavaScript	387469	vuejs/vue	154667	fix(provide/inject): do not mutate original pr...	vuejs/vue	Evan You	Evan You	fix(provide/inject): do not mutate original pr...
3	JavaScript	387469	vuejs/vue	154667	fix(reactivity): avoid using WeakMap for IE co...	vuejs/vue	Evan You	Evan You	fix(reactivity): avoid using WeakMap for IE co...
4	JavaScript	387469	vuejs/vue	154667	test: add test case for #12778\n	vuejs/vue	Evan You	Evan You	test: add test case for #12778
5	JavaScript	387469	vuejs/vue	154667	fix(types): fix spreading VNodeData in tsx (#1...	vuejs/vue	k-furusho	GitHub	fix(types): fix spreading VNodeData in tsx (#1...

Fonte: Autores

Metodologia

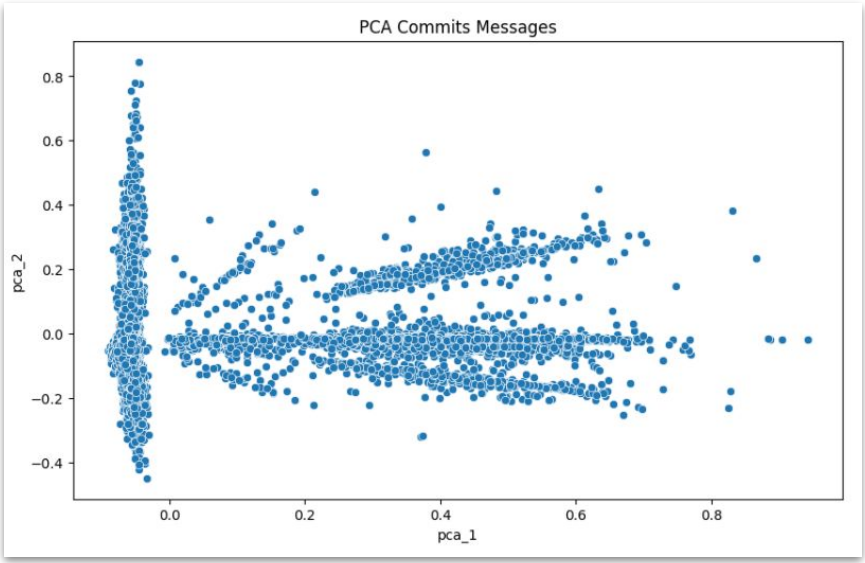
Pré-processamento realizado

- Usando NLTK (*Natural Language Toolkit*) do Python;
  Normalização de commits por linguagem (partindo da que tem menos);
- Remoção de valores vazios (não são grande quantidade);
- Remoção de caracteres não alfabéticos;
- Remoção de stopwords;
- Transformação de todas as palavras para letra minúscula (diminuir o dicionário);
- Remoção de palavras que não estão em inglês;
- Stemização (*Stemming*);

Metodologia

Pré-processamento realizado

- Frases com tamanhos considerados outliers - registros removidos;
  - utilizando quartiles acima de 75% e abaixo de 25%
- Features desnecessárias removidas;
- Aplicado o TF-IDF: atribui pesos a palavras em documentos com base em sua frequência no documento e raridade na coleção, destacando palavras importantes ao reduzir o impacto de termos comuns;
- Utilizado o PCA para redução de dimensionalidade.



Fonte: Autores

Metodologia

Pré-processamento realizado

- Exemplos de mensagens antes e após o pré-processamento

message_old	message
see 11/05 log\n	see log
Add builder to MockServerWebExchange\n\nIssue:...	add builder
fixed incorrect example and JSF reference\n	fix incorrect exampl refer
Typo\n	typo
Map resolution for multiple beans applies to p...	map resolut multipl plain map interfac declar
Support Part/MultiPartFile arrays in ArgumentR...	support prior request could argument array not...
@Nullable all the way: null-safety at field le...	field level field interact underli therefor
Add WebSocketMessage and WebSocketHandler sub-...	add type also choos handl also
Mark log4j support classes as deprecated in fa...	mark support class favor apach declar
Merge pull request #835 from ndebeiss/master\n...	merg pull request support singl doubl

Fonte: Autores

Metodologia

Pré-processamento realizado

- Dados realmente utilizados

lang	stars	message
Python	64967	add topolog sort add topolog sort fix topolog ...
Python	46118	remov invalid document
Java	34507	execut per event
Java	34507	implement interfac directli
JavaScript	57571	remov unnecessari anim loop
Java	34507	fix issu commit lazi issu
Java	34507	allow commit interfac use flag commit properti
Ruby	27765	ensur simul version snapshot choos last simul ...
Python	46118	improv new add new
Python	46118	bump nightli version

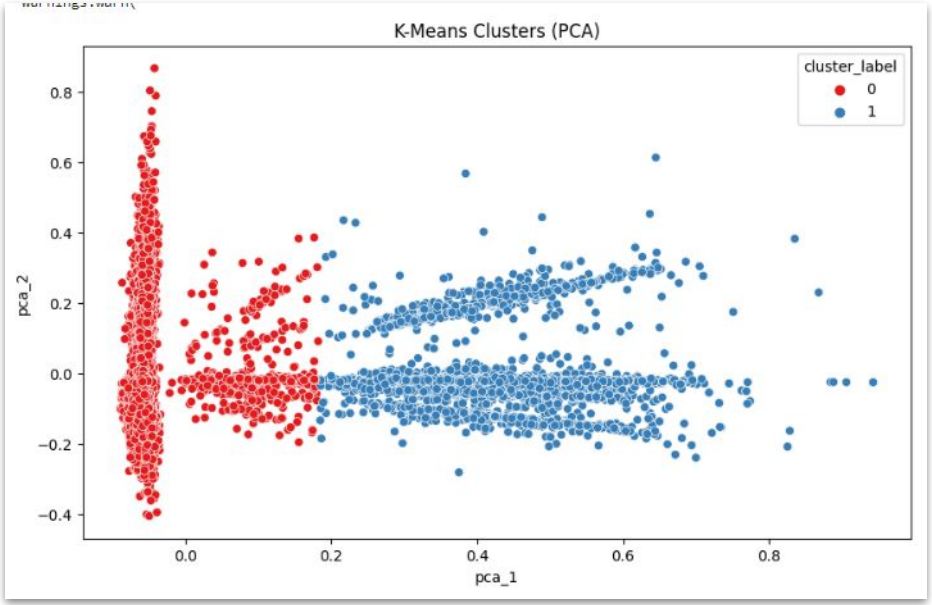
Fonte: Autores

Metodologia

Construção do modelo de clustering

Tentativa com algoritmo de **particionamento**

```
Silhouette Score: 0.7689744495397207
Davies-Bouldin Score: 0.42321097000774394
Calinski-Harabasz Score: 56214.71651701608
```



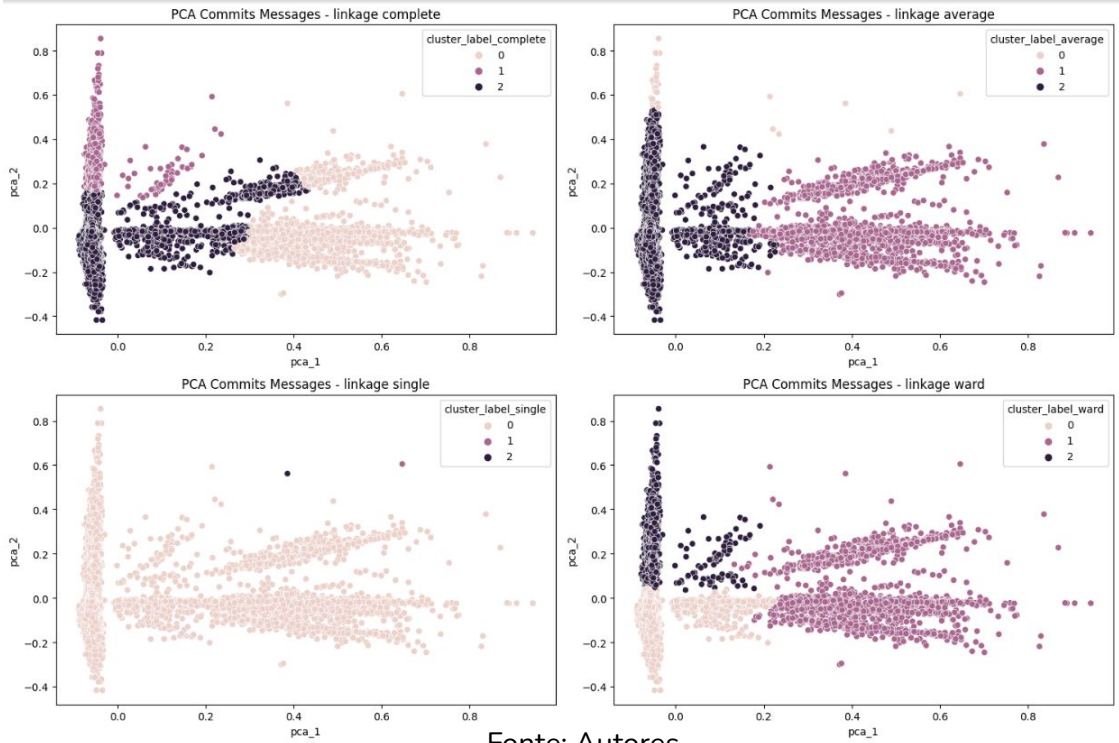
Fonte: Autores

Metodologia

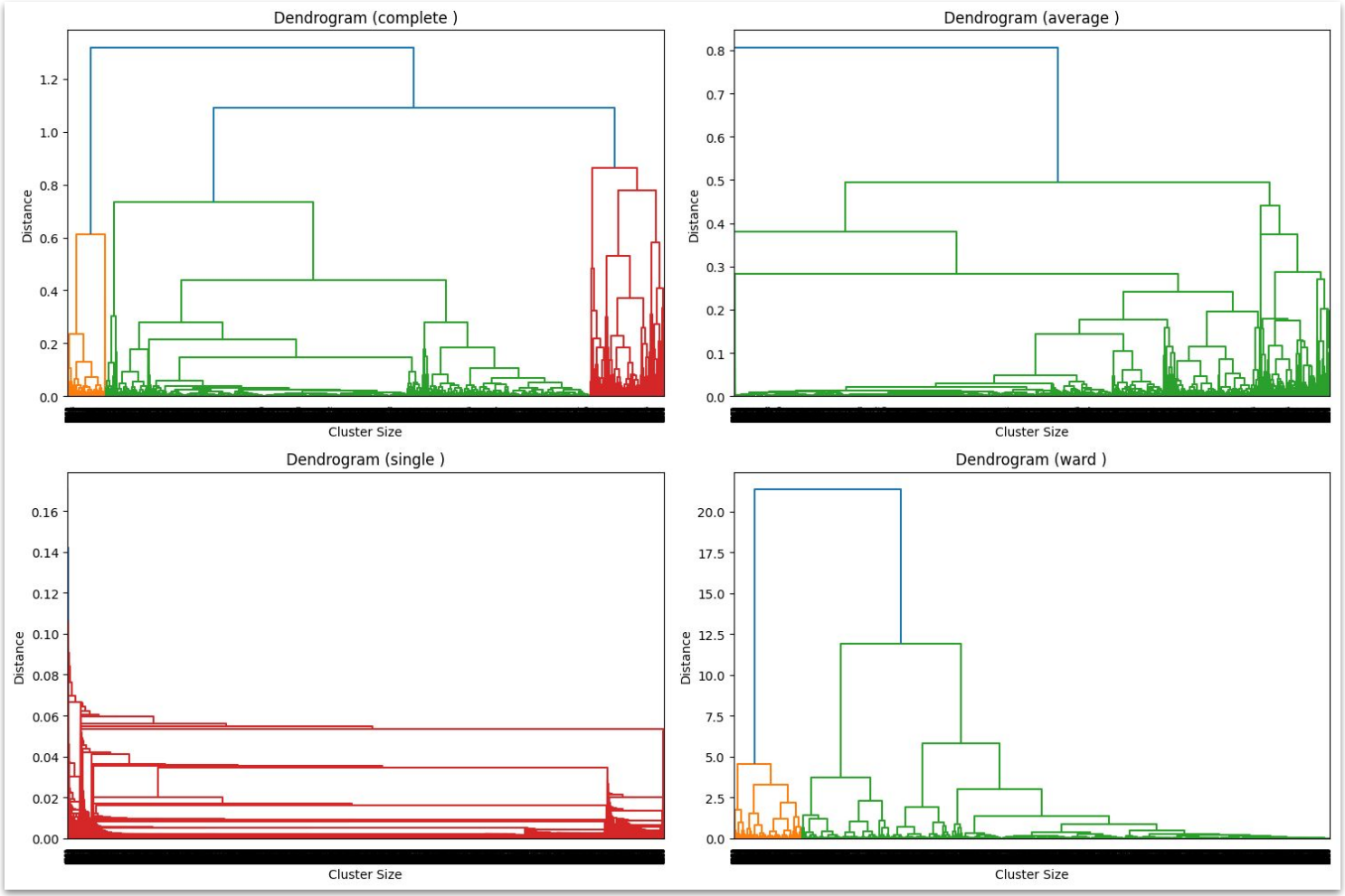
Construção do modelo de clustering

Tentativa com algoritmo **hierárquico**

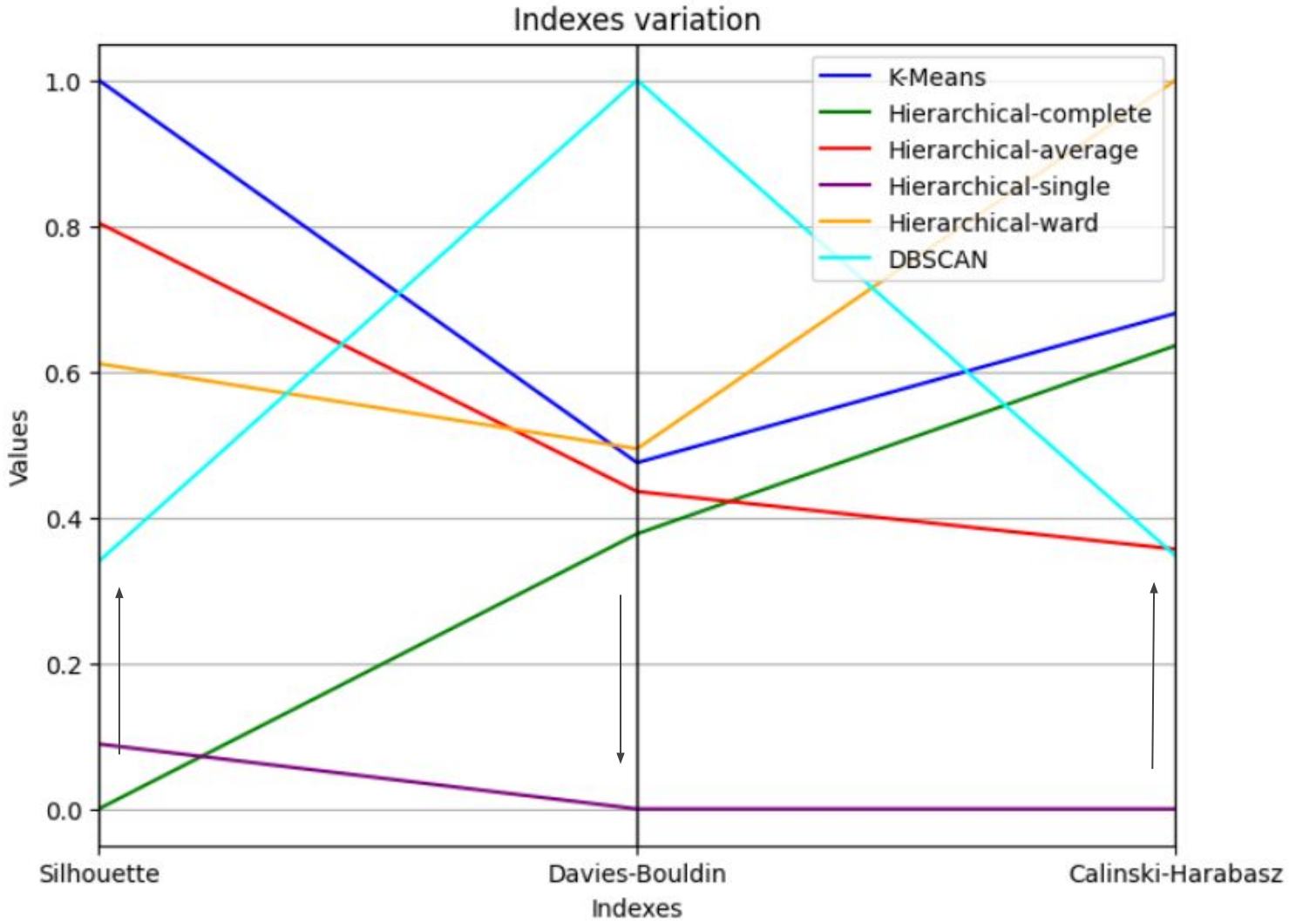
```
-----
complete
Silhouette Score: 0.5877327971353952
Davies-Bouldin Score: 0.9291933873012368
Calinski-Harabasz Score: 10939.239701288416
-----
average
Silhouette Score: 0.7287283432916225
Davies-Bouldin Score: 0.5533805396330911
Calinski-Harabasz Score: 15460.019139940609
-----
single
Silhouette Score: 0.6793851666020668
Davies-Bouldin Score: 0.1888395531441093
Calinski-Harabasz Score: 20.747345009215632
-----
ward
Silhouette Score: 0.5737347705138809
Davies-Bouldin Score: 0.6996531126939034
Calinski-Harabasz Score: 64994.38475599128
```



Fonte: Autores



Fonte: Autores


Indexes variation
K-Means
Hierarchical-complete
Hierarchical-average
Hierarchical-single
Hierarchical-ward
DBSCAN
1.0
0.8
0.6
0.4
0.2
0.0
Values
Silhouette
Davies-Bouldin
Calinski-Harabasz
Indexes

(-1 -> 1)

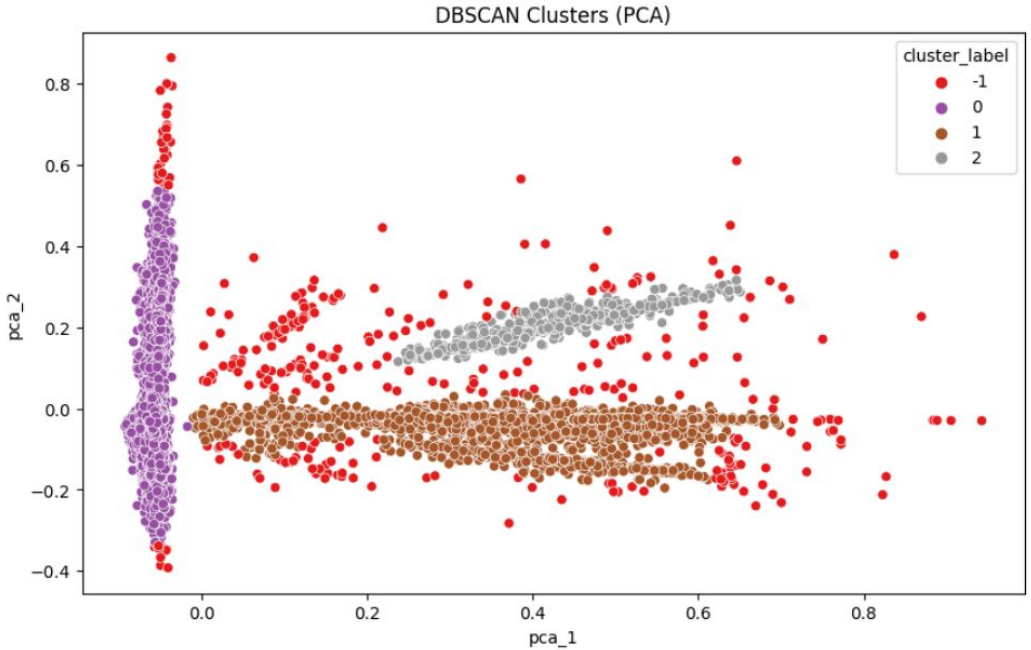
Metodologia

Construção do modelo de clustering

Algoritmo baseado em **densidade**

- Identificação automática dos clusters;
- Clusters de diferentes formas e tamanhos;
- Não é sensível a outliers;
- Independência em relação a primeira posição dos centróides;

```
Silhouette Score: 0.7291168010334109
Davies-Bouldin Score: 0.6903176087951147
Calinski-Harabasz Score: 29037.557569842287
```



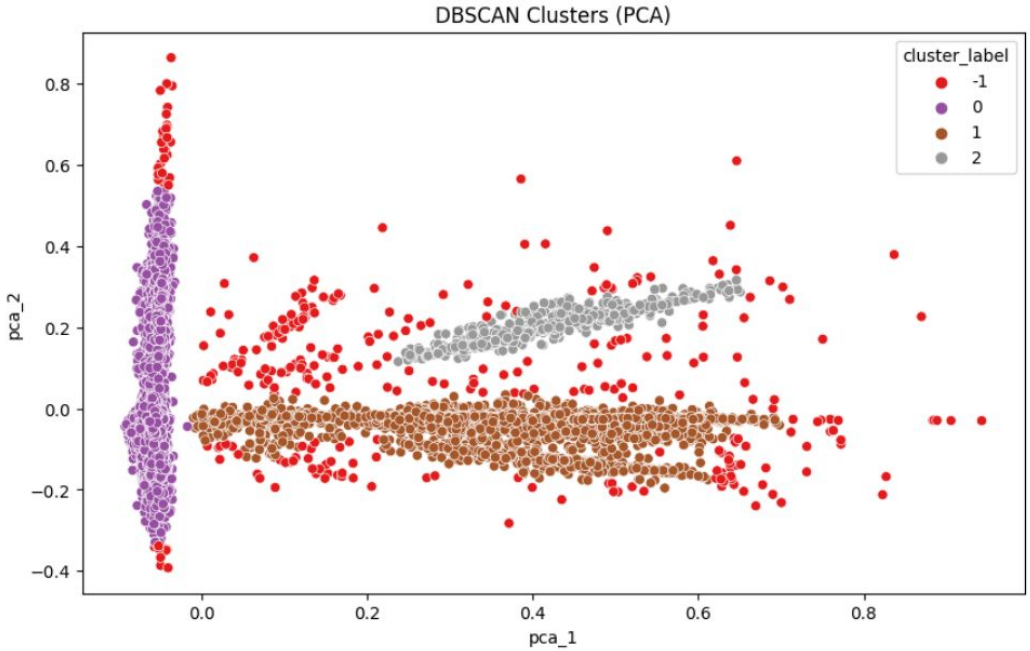
Fonte: Autores

Metodologia

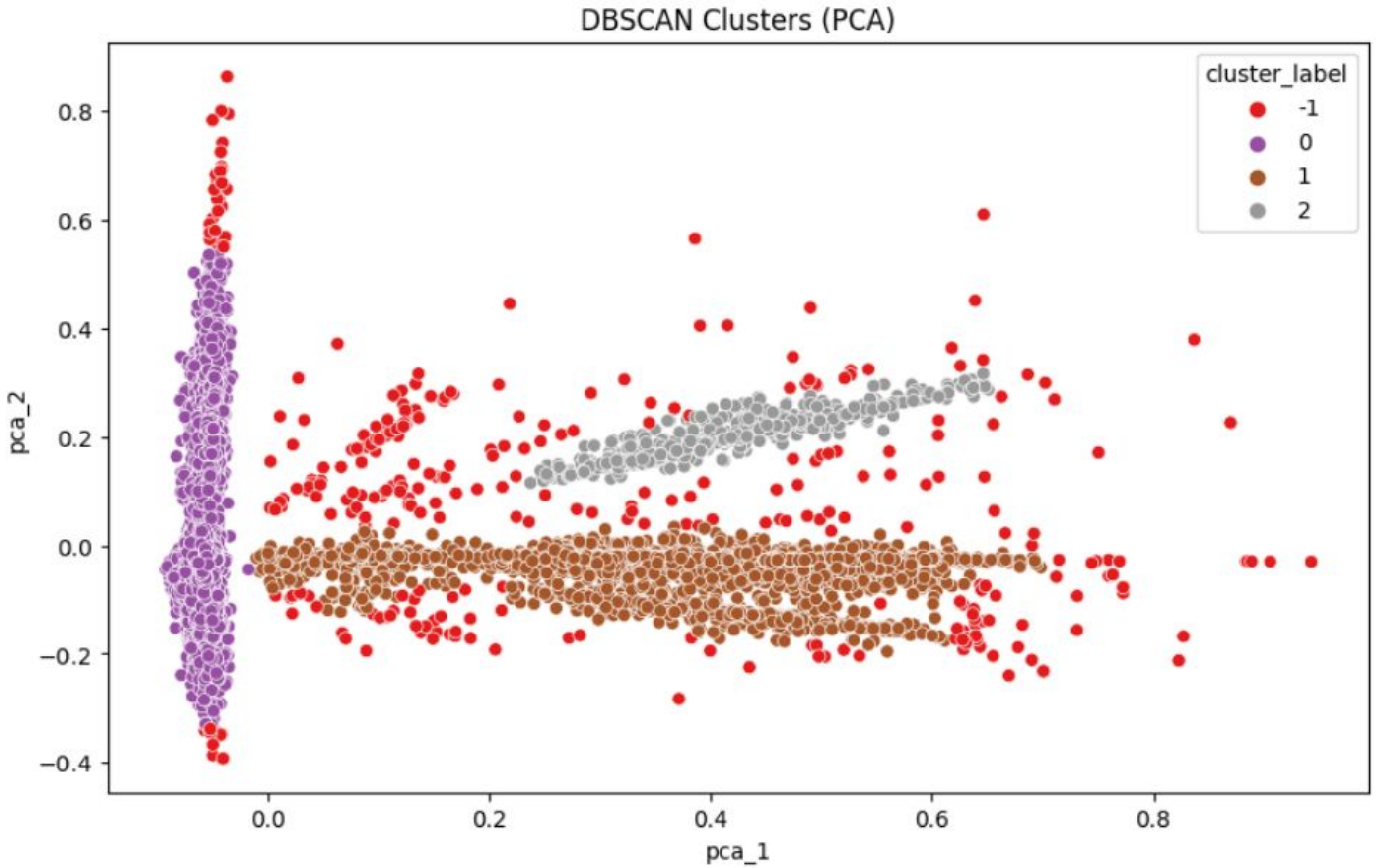
Construção do modelo de clustering

Algoritmo baseado em **densidade**

- Hiperparâmetros definidos e explorados manualmente e visualmente até um resultado satisfatório;
- Foi utilizado índice de silhueta para refinar os hiperparâmetros;



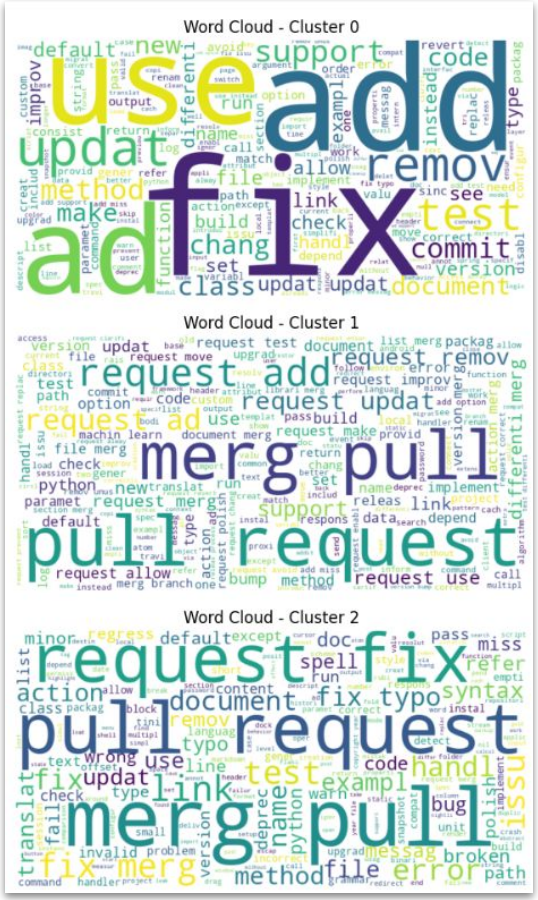
Fonte: Autores


DBSCAN Clusters (PCA)
cluster_label
-1
0
1
2
pca_2
pca_1
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
0.2
0.4
0.6
0.8

Metodologia

Pós-processamento

- Cálculo dos índices e comparação com execuções anteriores e de outros modelos;
- Análise visual do agrupamento de clusters (e comparação);
- Análises essenciais para refinar o processo de clustering;
- Exploração visual dos agrupamentos encontrados para encontrar a lógica do agrupamento;



Fonte: Autores

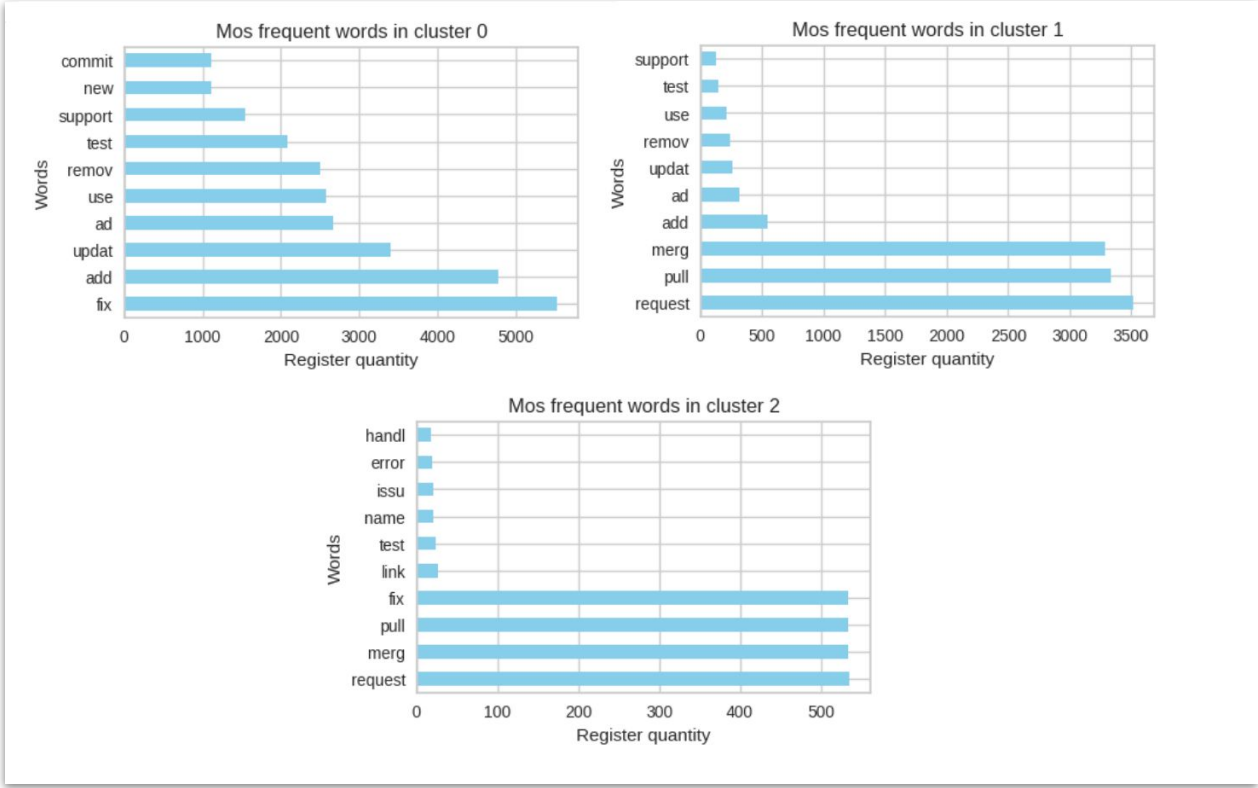
Metodologia

Pós-processamento

Cluster 0: atividade de adição, updates e fixes

Cluster 1: atividade de merge e pull requests relacionados a adições, remoções e updates

Cluster 2: atividades de merge e pull request relacionada a ajustes (fixes) testes e links

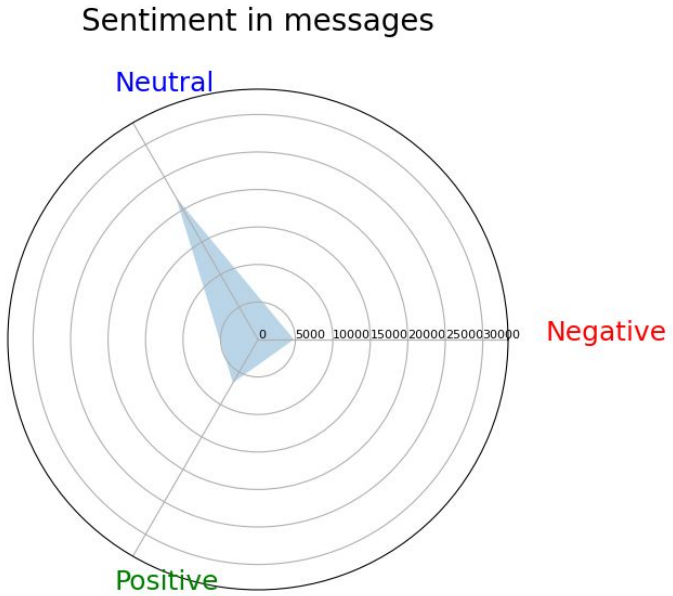


Fonte: Autores

Metodologia

Análise de sentimento

- biblioteca TextBlob em Python, uma ferramenta poderosa projetada para avaliar a polaridade;
- Pode ser classificada como positiva, negativa ou neutra;
- Relações de sentimento com as linguagens e em cada grupo definido nos clusters;

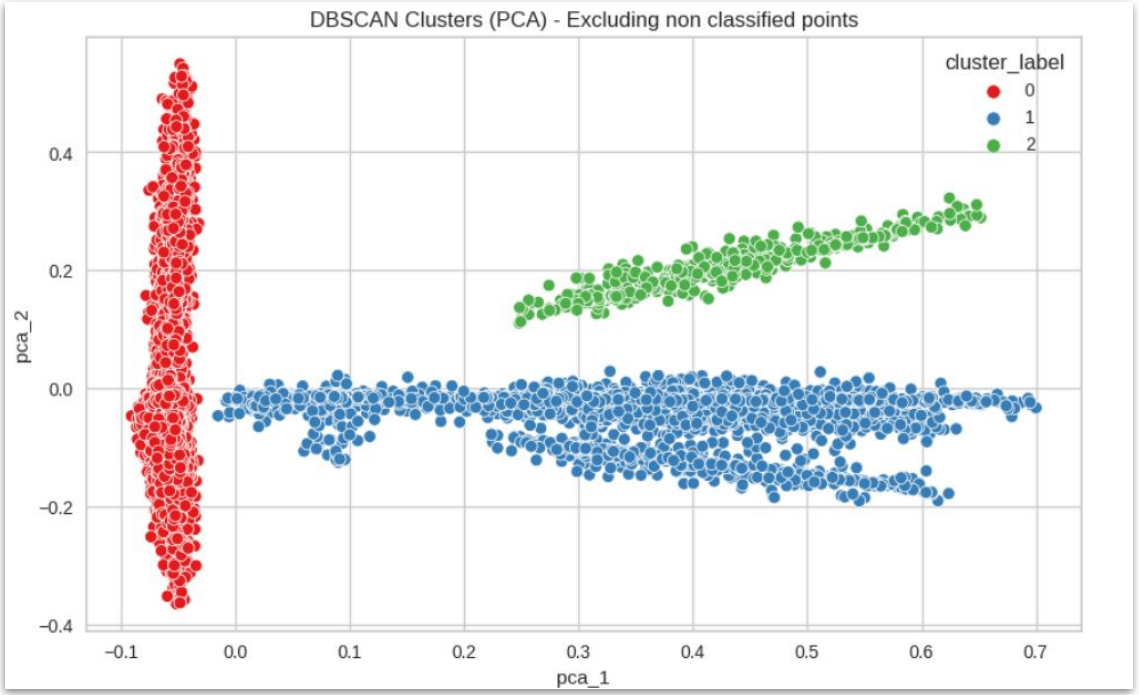


Fonte: Autores

Análise dos resultados

Análise dos Resultados

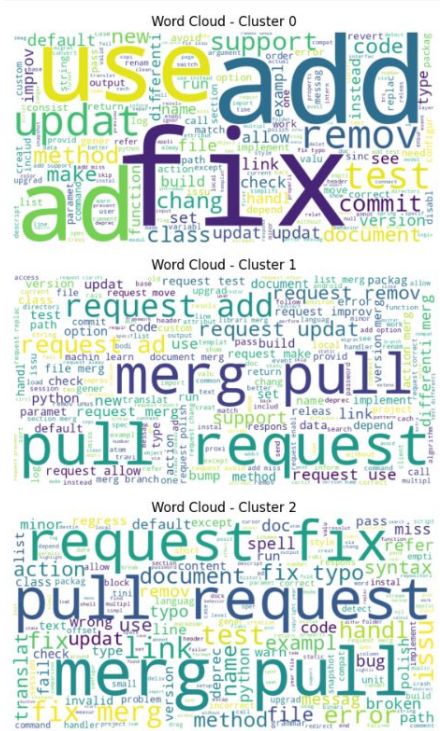
- Clusters definidos
- Remoção dos outliers



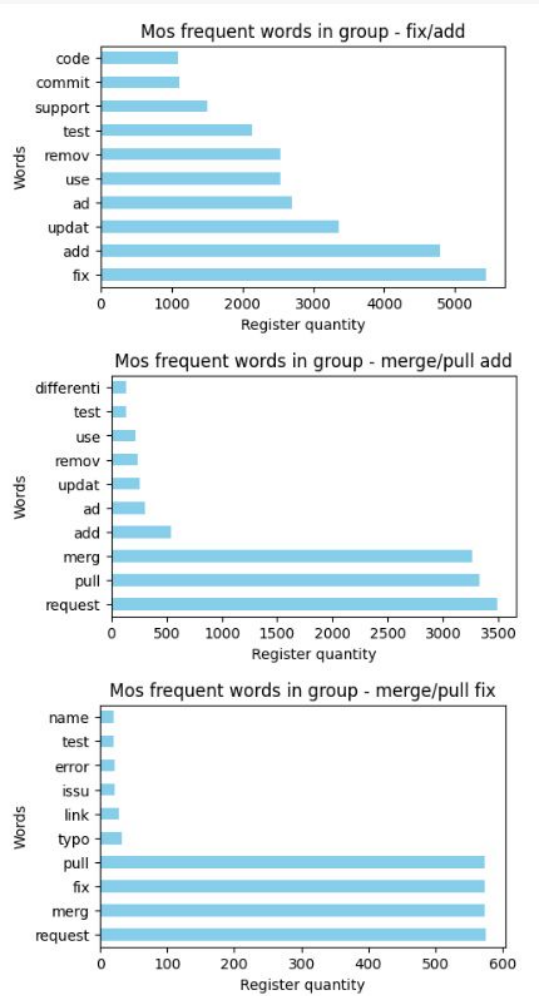
Fonte: Autores

Análise dos Resultados

- Nuvem de palavras de cada cluster para entender os agrupamentos
- Clusters 1 e 2 ainda com diferença muito pequena, mas encontrada por meio das palavras frequentes em cada um.



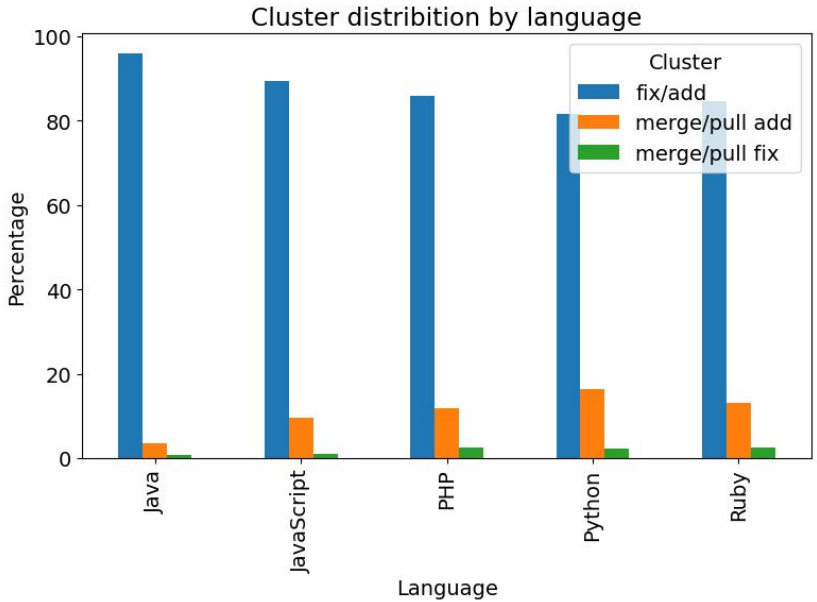
Fonte: Autores



Fonte: Autores

Análise dos Resultados

- **RQ1: as linguagens têm uma influência na distribuição das proporções dos grupos?**
- Linguagem de programação e grupos gerados;
- Java possui menos merge/pull;
- No geral quantidades balanceadas;

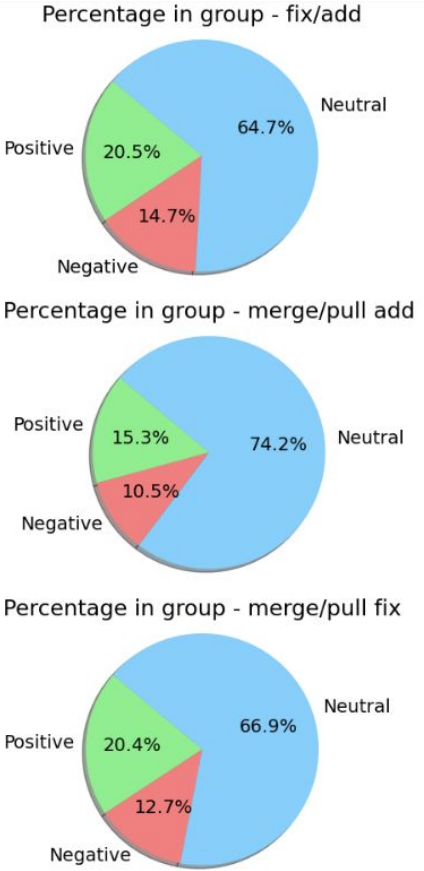


Fonte: Autores

Análise dos Resultados

- **RQ2: o tipo de atividade de desenvolvimento influencia os sentimentos expressos nos comentários?**
- Majoritariamente sentimentos neutros;
- Atividades de adição, updates, adds e fixes independentes (cluster 0) - mais sentimentos positivos;
- Atividades de merge e pull, sendo as de fixes e testes (cluster 1) com maior proporção de sentimentos positivos, considerando a área de sentimentos expressos.

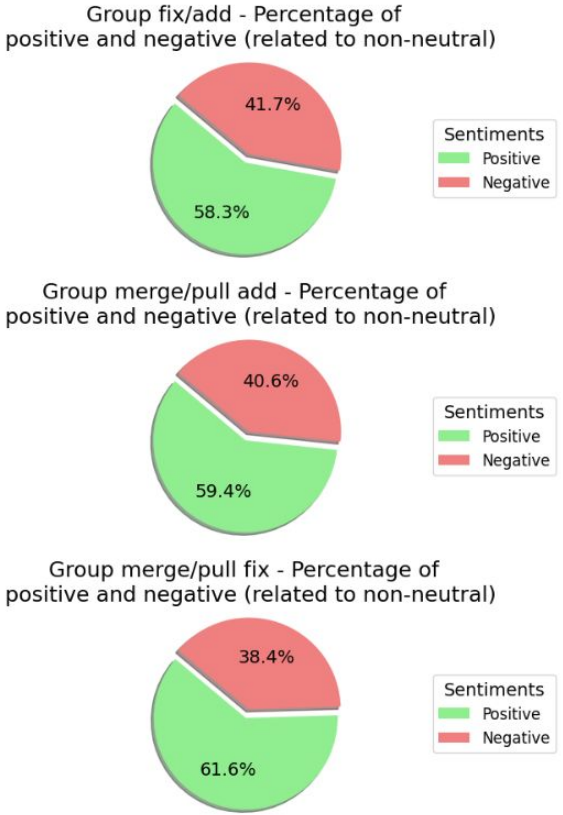
cluster 0 - fix, add, remove
cluster 1 - merge e pull request de add
cluster 2 - merge e pull request de fix



Fonte: Autores

Análise dos Resultados

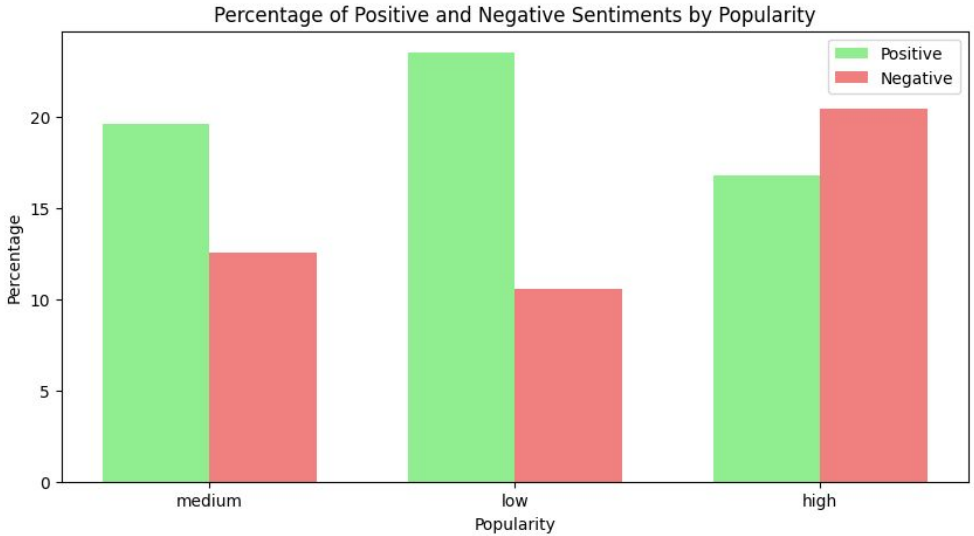
- Majoritariamente sentimentos neutros;
- Atividades de adição, updates, adds e fixes independentes (cluster 0) - mais sentimentos positivos;
- Atividades de merge e pull, sendo as de fixes e testes (cluster 1) com maior proporção de sentimentos positivos, considerando a área de sentimentos expressos.



Fonte: Autores

Análise dos Resultados

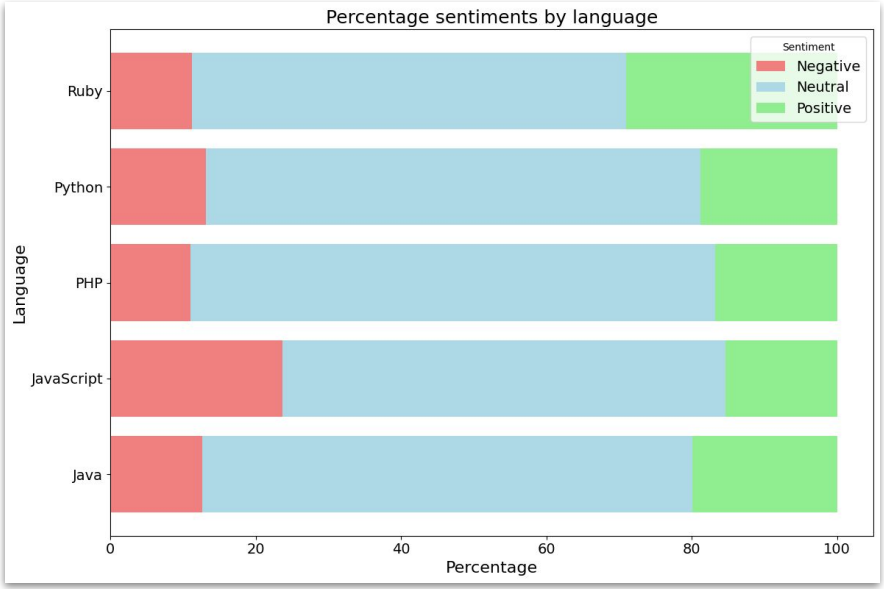
- **RQ3: Os sentimentos expressos em comentários de commits refletem na popularidade do projeto entre desenvolvedores?**
- Em relação aos sentimentos, os negativos são predominantes nos mais populares e maior porcentagem em relação aos demais.
- Menos populares possuem mais commits com mensagens positivas;



Fonte: Autores

Análise dos Resultados

- **RQ4: As linguagens de programação mais utilizadas têm influência nos sentimentos expressos nos comentários?**
- Apesar de ser uma linguagem popular, JavaScript tem muito mais commits com mensagens negativas do que os demais;
- Ruby muito mais mensagens positivas;
- Demais estão balanceadas entre si.

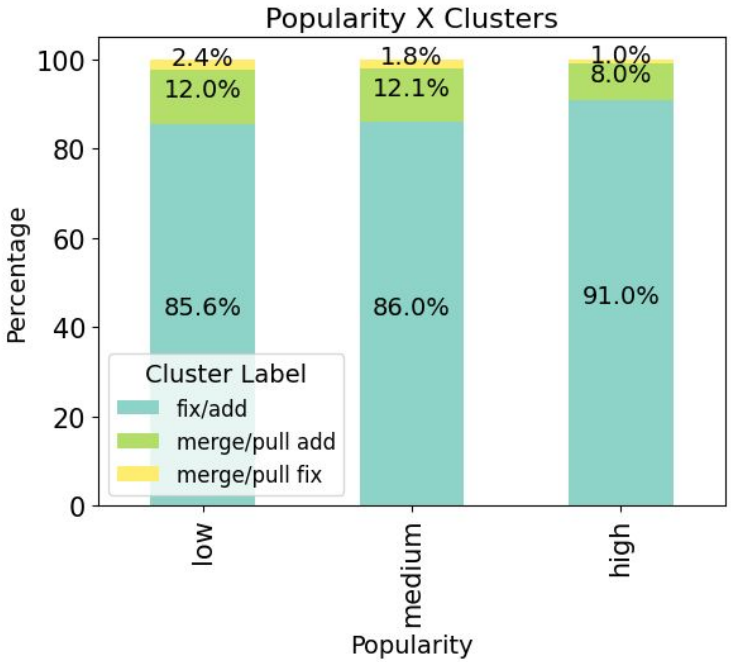


Fonte: Autores

Análise dos Resultados

- **RQ5: as atividades mais recorrentes dentro do projeto tem alguma relação com a popularidade dos repositórios?**
- Popularidade em relação aos clusters definidos. Os mais populares possuem menos merge e pull requests;
- No geral, porcentagem balanceada.

cluster 0 - fix, add, remove
cluster 1 - merge e pull request de add
cluster 2 - merge e pull request de fix



Fonte: Autores

Conclusão e trabalhos futuros

Conclusão e Trabalhos Futuros

- A partir deste estudo, é possível entender quais as tarefas que os usuários têm mais sentimentos negativos atrelados e até mesmo quais linguagens possuem mais dificuldades;
- Os sentimentos a respeito de atividades de desenvolvimento de features são mais expressos (não neutros) e mais positivos do que ao realizar um merge ou pull request;
- Futuramente deve ser feito um estudo aprofundado sobre as melhores features para a clusterização de commits;
- Ainda é um desafio a aquisição destes dados por conta da limitação de requisições na API disponibilizada pelo GitHub e falta de dados na base de dados BigQuery;
- Também podem ser analisados repositórios que utilizam a conversão de tipos de mensagens nos commits.

Bibliografia

[1] V. Costa and L. Ponciano, “Minerando padrões de interação de programadores com repositórios na plataforma github,” 2018.
[2] V. Cosentino, J.-L. C. Izquierdo, and J. Cabot, “*A systematic mapping study of software development with github*” IEEE access, vol. 5, pp. 7173–7192, 2017.
[3] S. F. Huq, A. Z. Sadiq, and K. Sakib, “*Is developer sentiment related to software bugs: An exploratory study on github commits*” in 2020 IEEE *27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2020, pp. 527–531.
[4] T. Ji, J. Pan, L. Chen, and X. Mao, “*Identifying supplementary bug-fix commits*” in 2018 IEEE *42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018, pp. 184–193.
[5] I. Keivanloo, C. Forbes, A. Hmood, M. Erfani, C. Neal, G. Peristerakis, and J. Rilling, “*A linked data platform for mining software repositories*” in 2012 *9th IEEE Working Conference on Mining Software Repositories (MSR)*. Zurich, Switzerland: IEEE, 2012, pp. 32–35.
[6] P. Anbalagan and M. Vouk, “*On mining data across software repositories*” in 2009 6th *IEEE International Working Conference on Mining Software Repositories*.
[7] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, “T*ext preprocessing for text mining in organizational research: Review and recommendations*,” *Organizational Research Methods*, vol. 25, no. 1, pp. 114–146, 2022
[8] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, “*A brief survey of text mining: Classification, clustering and extraction techniques*,” arXiv preprint arXiv:1707.02919, 2017.

Bibliografia

[9] G. V. B. Moser et al., “Análise de similaridade entre tf-idf e modelos contextualizados de linguagem baseados em tokens,” 2022.
[10] D. Sisodia, L. Singh, S. Sisodia, and K. Saxena, “*Clustering techniques: a brief survey of different clustering algorithms*,” International Journal of Latest Trends in Engineering and Technology (IJLTET), vol. 1, no. 3, pp. 82–87, 2012.
[11] G. Ahalya and H. M. Pandey, “*Data clustering approaches survey and analysis*,” in 2015 *International Conference on Futuristic Trends on Computational Analysis and Knowledge Management* (ABLAZE). IEEE, 2015, pp. 532–537.
[12]] I. L. da Silva, R. F. Mello, P. B. Miranda, A. C. Nascimento, I. W. Maldonado, and J. L. Coelho Filho, “Assessment of text clustering approaches for legal documents,” in Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional ˆ . SBC, 2021, pp. 37–48.
[13] V. Vargas, E. Amorim, J. A. d. M. Brito, and G. S. Semaan, “Um estudo de variantes do índice de validação silhueta,” in Anais da IV Escola Regional de Informática do Rio de Janeiro. SBC, 2021, pp. 123–126.
[14]X. Wang and Y. Xu, “*An improved index for clustering validation based on silhouette index and calinski-harabasz index*,” in IOP *Conference Series: Materials Science and Engineering*, vol. 569, no. 5. IOP Publishing, 2019, p. 052024
[15] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “*A survey on sentiment analysis methods, applications, and challenges*,” Artificial Intelligence Review, vol. 55, no. 7, pp. 5731–5780, 2022
[16] J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, and E. Herrera-Viedma, “Sentiment analysis: A review and comparative analysis of web services,” Information Sciences, vol. 311, pp. 18–38, 2015.