Feature Selection utilizando Particle Swarm Optimization

Leticia Freire de Figueiredo

29 de Agosto de 2019

Motivação

1. Utilizar o que foi apresentado em outras aulas
2. Mostrar uso de algoritmos evolucionários

Feature Selection: O que é feature selection?

1. Representação dos dados pode usar muitas features
2. Algumas são mais relevantes
3. Features redundantes degradam a performance: velocidade e acurácia

Feature Selection: O que é feature selection?

Definição

Feature Selection é o problema de escolher um subconjunto de features que é necessário para descrever o conceito alvo.

Feature Selection x Feature Extraction

1. **Feature selection**: subconjunto de features das features originais
2. **Feature extraction**: projeta as features num novo espaço de features, no qual as novas features são combinações das features originais

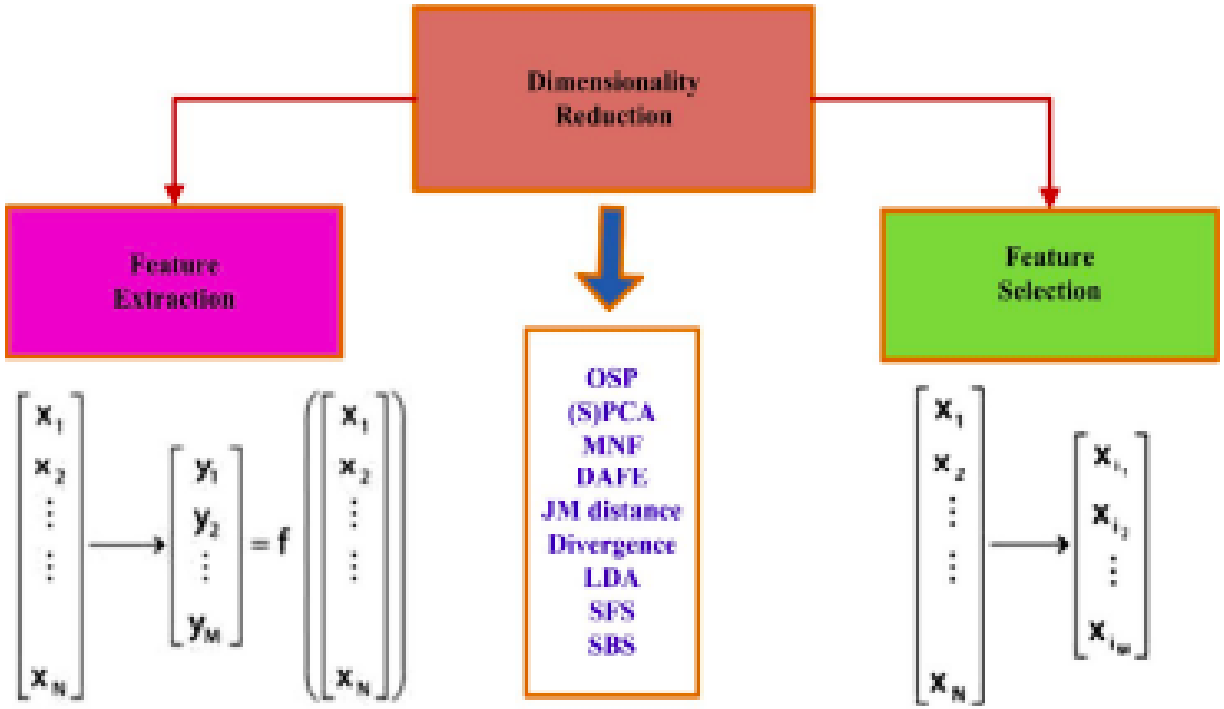

Figura 1: Feature Extraction x Feature Selection. Fonte: Merugu [1]

Feature Selection: Modelos

- **Filtro**
  1. Avaliam as features sem utilizar qualquer classificador
  2. Rankeia as features baseado em algum critério
  3. As features com maiores rankingssão usadas na classificação
  4. Desvantagem: ignora o feito do subconjunto escolhido na performance do algoritmo
  5. Utilizado SelectKBest

Feature Selection: Modelos

- **Wrapper**
  1. Utiliza um classificador específico para avaliar a qualidade da escolha das features
  2. Procura um subconjunto de features
  3. Avalia a performance deste subconjunto no algoritmo
  4. Repete os 2 itens acima até alcançar a qualidade desejada.
  5. Obtém estimativas acurácia melhores que o filtro
  6. Desvantagem: computacionalmente caro
  7. **Utilizado Particle Swarm Optimization**

Feature Selection: Modelos

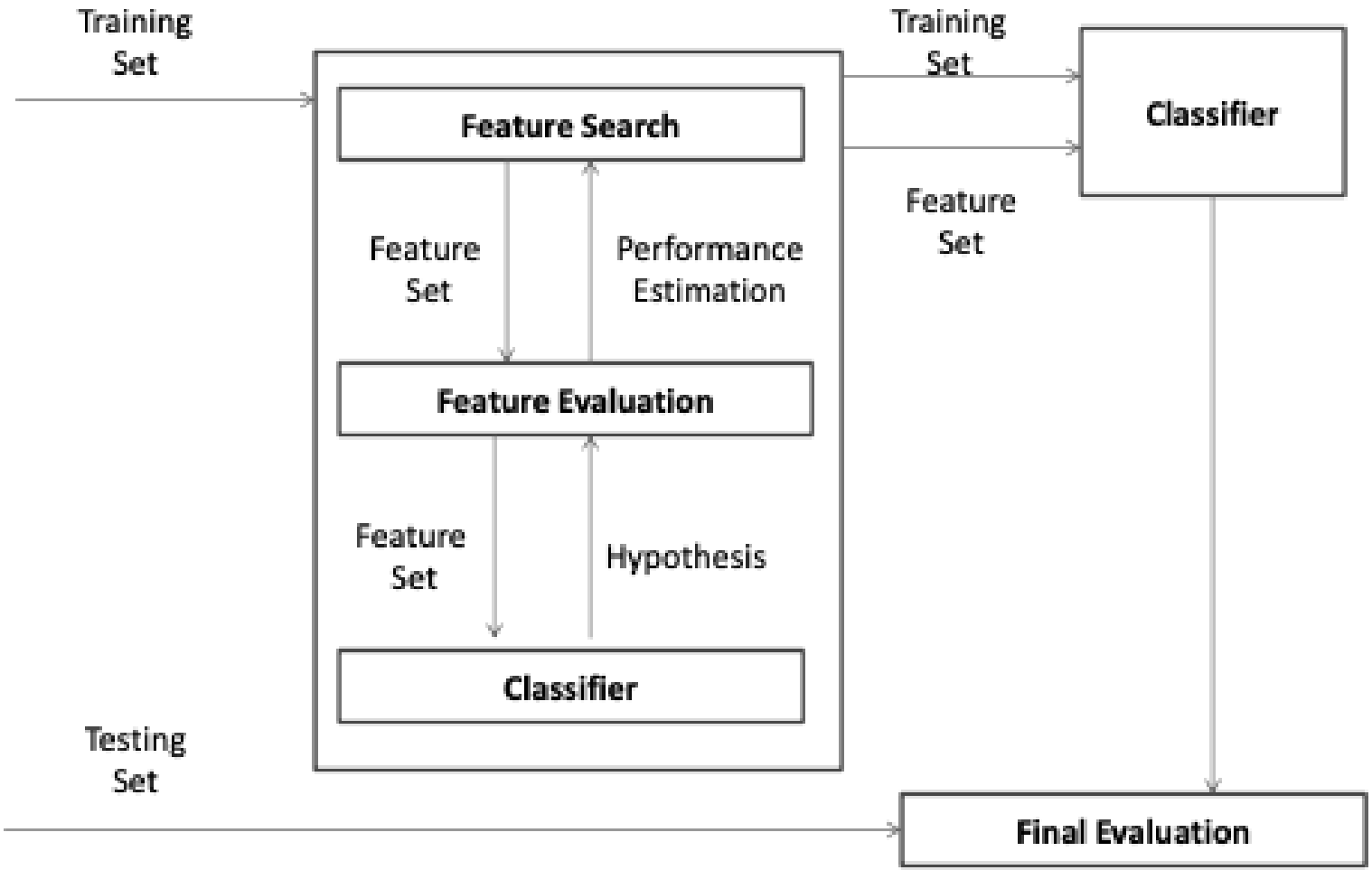

Figura 2: Exemplo de wrapper. Fonte: Tang, Alelyani e Liu [2]

Feature Selection: Modelos

- **Embedded**
  1. Utiliza um classificador para avaliar a qualidade das features
  2. Existem 3 tipos de métodos para este tipo
  3. Um deles: utiliza todas as features para treinar o modelo e tenta eliminar algumas features
  4. Tem vantagens do Wrapper e do filter
  5. **Utilizado SelectFromModel**

PSO para Feature Selection

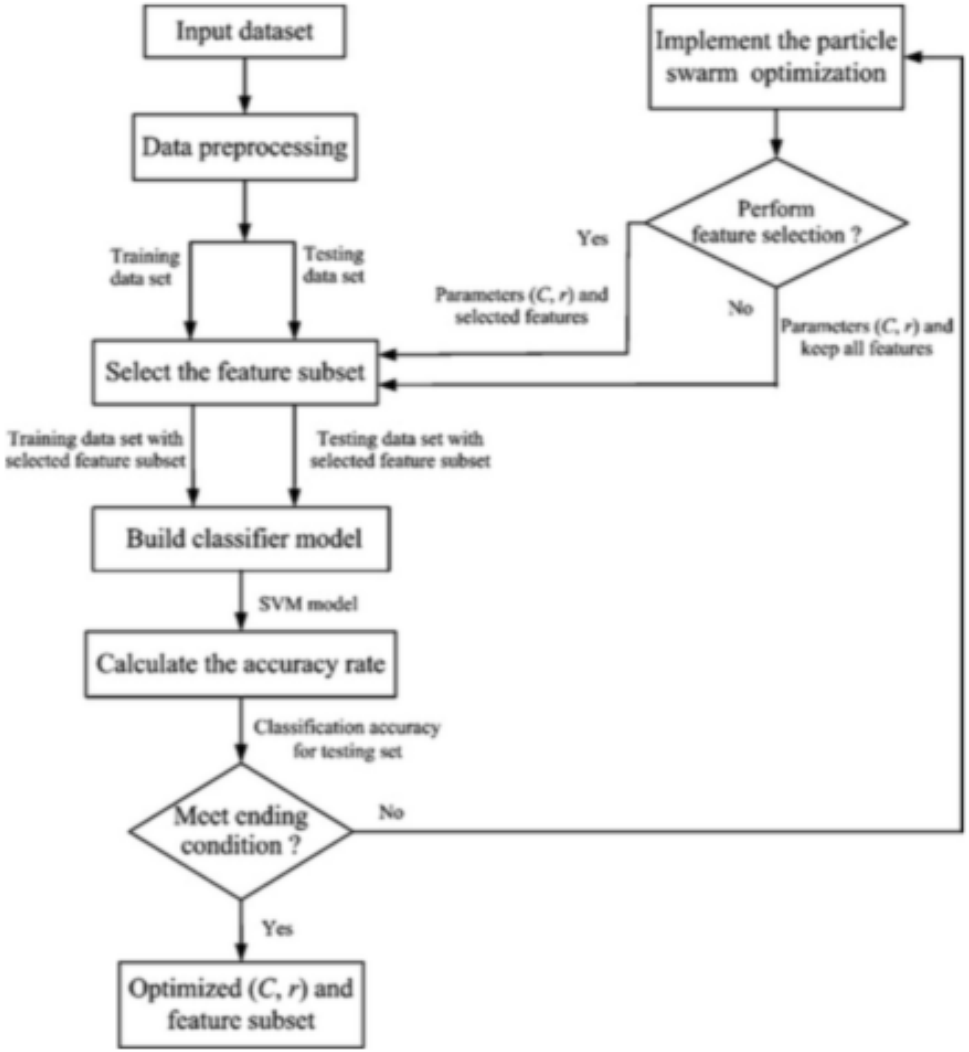

Figura 3: Esquema do PSO para seleção de features. Fonte: Lin et al. [3]

Resultados: Datasets

- Datasets utilizados: Pokemon. Fonte: *The Complete Pokemon Dataset / Kaggle* [4]
  1. Objetivo é verificar se um Pokemon é lendário ou não
  2. Possui 801 linhas, 41 colunas

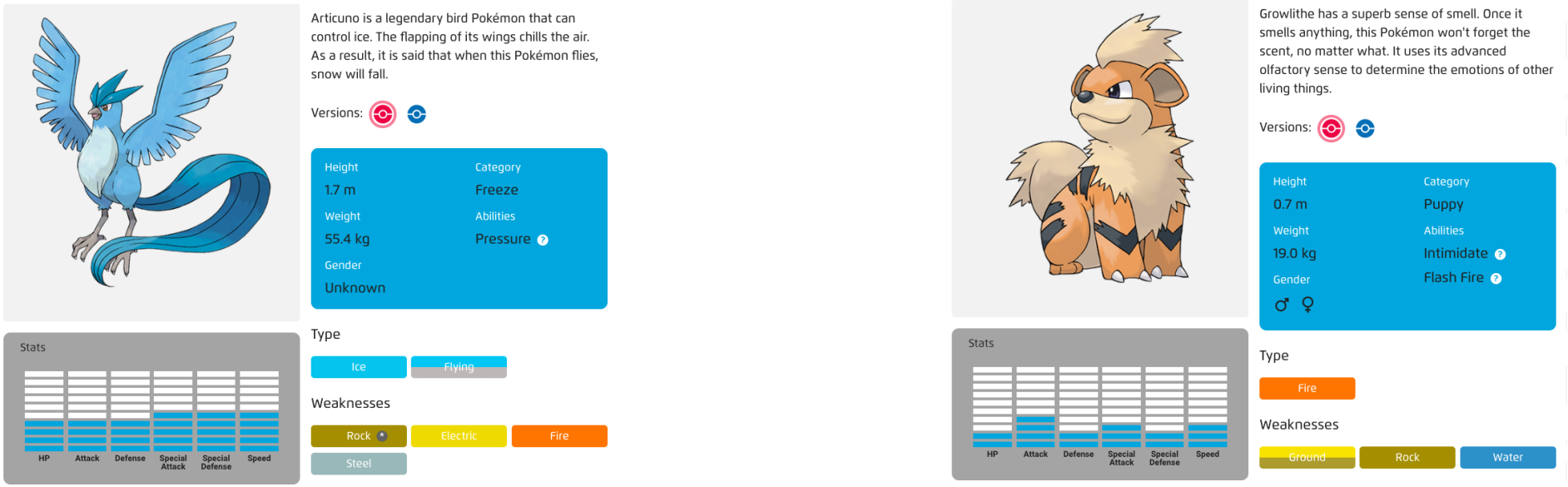

Figura 4: Exemplo de Pokemons.

Pokédex / www.pokemon.net.br [5]

Resultados: Dataset

- Datasets utilizados: Spam. Fonte: *UCI Machine Learning Repository: Spambase Data Set* [6]
  1. Verificar se um e-mail é spam ou não
  2. Possui 4601 linhas, 57 colunas
- Datasets utilizados: Doença de Parkinson. Fonte: *UCI Machine Learning Repository: Parkinson's Disease Classification Data Set* [7]
  1. Verificar se um paciente tem a doença ou não
  2. Possui 756 linhas, 754 colunas

Resultados: Algoritmos utilizados

- Feature Selection
  1. Filtro: SelectKBest, com k = 5, 15 e 20
  2. Wrapped: PSO, com 10 iterações
  3. Embedded: SelectFromModel
- Classificação
  1. XGBoost
  2. SVM
  3. Decision Tree

Resultados: Pokemon

Classificador	Sem Feature selection	SelectKBest			PSO	SelectFromModel
		5	15	20		
XGBoost	0.97928; 0.00250	0.97528; 0.00093	0.97378; 0.00137	0.97377; 0.00097	0.98390; 0.00102	0.97676; 0.00985
SVM	0.91261; 3.01e-05	0.91262; 2.96e-05	0.91260; 3.60e-05	0.91260; 2.76e-05	0.91269; 2.37e-06	0.96265; 0.01617
Decision Tree	0.98427; 0.00312	0.98626; 0.00097	0.98551; 0.00195	0.98476; 0.00221	0.99089; 0.00137	0.99253; 0.00483

Tabela 1: Resultados com dataset de Pokemon.

Resultados: Spam

Classificador	Sem Feature selection	SelectKBest			PSO	SelectFromModel
		5	15	20		
XGBoost	0.94151; 0.00103	0.86824; 0.00136	0.93036; 0.00114	0.94118; 0.00110	0.94333; 0.00025	0.93533; 0.00594
SVM	0.714258; 0.00154	0.71471; 0.00132	0.71478; 0.00181	0.71384; 0.00092	0.92342; 0.00072	0.89913; 0.00754
Decision Tree	0.91512; 0.00262	0.91417; 0.00306	0.91425; 0.00310	0.91577; 0.00370	0.92342; 0.00072	0.90043; 0.00805

Tabela 2: Resultados com dataset de spam.

Resultados: Parkinson

Classificador	Sem Feature selection	SelectKBest			PSO	SelectFromModel
		5	15	20		
XGBoost	0.86798; 0.00504	0.80399; 0.00357	0.81754; 0.00525	0.82475; 0.00683	0.87777; 0.00321	0.85462; 0.02158
SVM	0.74104; 0.00557	0.7407; 0.00338	0.7414; 0.00323	0.74152; 0.00230	0.75025; 0.00199	0.51762; 0.25606
Decision Tree	0.79934; 0.00996	0.80399; 0.00357	0.81754; 0.00525	0.82475; 0.00683	0.82844; 0.00451	0.80176; 0.03809

Tabela 3: Resultados com dataset Parkinson.

Resultados: valores corrigidos

- Nos próximos slides, serão apresentados os resultados com a implementação corrigida
- O dataset foi dividido em: 70% para treino e 30% para teste
- No conjunto de treino, foi utilizado Kfold com k=10
- Após aplicado o Kfold e encontrado o melhor conjunto de treino, é calculada a acurácia utilizando o conjunto de teste
- Nos resultados seguintes são apresentados a média e o desvio padrão, com 10 iterações de cada classificador

Resultados: Pokemon - corrigido

Classificador	Sem Feature selection	SelectKBest			PSO	SelectFromModel
		5	15	20		
XGBoost	0.94730; 0.04336	0.93983; 0.03703	0.95186; 0.03083	0.94854; 0.03660	0.96431; 0.03844	0.91286; 0.01710
SVM	0.92489; 0.01489	0.91369; 0.01678	0.90871; 0.01496	0.91327; 0.01251	0.98298; 0.03189	0.90207; 0.11143
Decision Tree	0.98755; 0.00694	0.98838; 0.00580	0.98672; 0.00689	0.98796; 0.00506	0.99543; 0.00433	0.99253; 0.00483

Tabela 4: Resultados com dataset de Pokemon.

Resultados: Spam - corrigido

Classificador	Sem Feature selection	SelectKBest			PSO	SelectFromModel
		5	15	20		
XGBoost	0.94011; 0.00546	0.86466; 0.01044	0.92729; 0.00484	0.93765; 0.00863	0.93569; 0.00581	0.92614; 0.00974
SVM	0.70159; 0.00898	0.70803; 0.01575	0.70412; 0.01218	0.70050; 0.01254	0.87385; 0.05227	0.90094; 0.00653
Decision Tree	0.91013; 0.00665	0.84974; 0.00817	0.90796; 0.00988	0.91397; 0.00613	0.91556; 0.00956	0.90181; 0.00853

Tabela 5: Resultados com dataset de spam.

Resultados: Parkinson - corrigido

Classificador	Sem Feature selection	SelectKBest			PSO	SelectFromModel
		5	15	20		
XGBoost	0.85242; 0.02357	0.80176; 0.01970	0.80088; 0.01846	0.82158; 0.02745	0.87621; 0.00912	0.79339; 0.03454
SVM	0.75022; 0.02584	0.79515; 0.01992	0.74140; 0.01848	0.75550; 0.02570	0.75682; 0.01824	0.59207; 0.23417
Decision Tree	0.80528; 0.01318	0.68325; 0.02188	0.73127; 0.02322	0.75418; 0.02007	0.84449; 0.02306	0.79691; 0.03499

Tabela 6: Resultados com dataset Parkinson.

Resultados: Análise das features

	PSO	SelectFromModel
XGBoost	20.9; 3.38969	5.4; 0.48989
SVM	24.3; 4.14849	11.1; 1.51327
Decision Tree	19.9, 1.7	3.3; 0.45825

Tabela 7: Resultados com dataset Pokemon.

Resultados: Análise das features

SelectKBest: k = 20

abilities, attack, base_egg_steps, base_happiness, base_total, capture_rate, classfication, defense, experience_growth, height_m, hp, japanese_name, name, pokedex_number, sp_attack, sp_defense, speed, type1, weight_kg, generation

PSO (XGBoost): selecionando as 20 features mais escolhidas

abilities, against_bug, against_fire, against_flying, against_ghost, against_grass, against_ice, against_poison, against_psychic, against_steel, against_water, attack, base_egg_steps, base_happiness, capture_rate, defense, height_m, hp, sp_attack, type2

Feature Selection utilizando Particle Swarm Optimization

Leticia Freire de Figueiredo

29 de Agosto de 2019

Referências

[1] Suresh Merugu. "A Review of Some Information Extraction Methods, Techniques and their Limitations for Hyperspectral Dataset". Em: *International Journal of Advanced Research in Computer Engineering Technology (IJARCET)* 3 (jul. de 2014), pp. 2394–2400.

[2] Jiliang Tang, Salem Alelyani e Huan Liu. *Feature Selection for Classification: A Review*.

[3] Shih-Wei Lin et al. "Particle swarm optimization for parameter determination and feature selection of support vector machines". Em: *Expert Syst. Appl.* 35 (nov. de 2008), pp. 1817–1824. DOI: 10.1016/j.eswa.2007.08.088.

[4] *The Complete Pokemon Dataset | Kaggle*. https://www.kaggle.com/rounakbanik/pokemon#pokemon.csv. (Accessed on 08/28/2019).

Referências

[5] *Pokédex | www.pokemon.net.br*. https://www.pokemon.com/br/pokedex/. (Accessed on 08/28/2019).

[6] *UCI Machine Learning Repository: Spambase Data Set*. https://archive.ics.uci.edu/ml/datasets/Spambase. (Accessed on 08/28/2019).

[7] *UCI Machine Learning Repository: Parkinson's Disease Classification Data Set*. https://archive.ics.uci.edu/ml/datasets/Parkinson%27s+Disease+Classification#. (Accessed on 08/28/2019).

[8] *1.13. Feature selection — scikit-learn 0.21.3 documentation*. https://scikit-learn.org/stable/modules/feature_selection.html. (Accessed on 08/28/2019).

Referências

[9] *Welcome to PySwarms's documentation! — PySwarms 1.1.0 documentation*. https://pyswarms.readthedocs.io/en/latest/index.html. (Accessed on 08/28/2019).

[10] *lefreire/teia*. https://github.com/lefreire/teia. (Accessed on 09/04/2019).