Baseline v1

Como parâmetro de comparação, levar em consideração a v1 submetida para o CBIC 2017

- Melhor resultado todas as mensagens: F1 **50.54** (modelo B)

Evoluções v2

- Constraint na criação e em cada geração de indivíduos
  - Mais de uma massive function: fitness zero
  - Não ter if_then_else ou polaritySum* na raíz – penalização de 20% do fitness
- Pesos para os dicionários

Dicionários/Pesos						
W1	**W2**	**W3**	**W4**	**W5**	**W6**	**W7**
LIU	Sentiwordnet	AFFIN	Vader	Slang	Effect	Semeval2015

- Parâmetros principais [1]
  - Gerações: 51
  - População: 500

V2.1 - Testes 1

Usando valores discretos como terminais (0.0, 0.5, 1.0, 1.5, 2.0) para possíveis pesos e um range de valor real [0,2]

#	**Indivíduo**
1	polaritySumAVGUsingWeights(replaceNegatingWords(removeAllPonctuation(replaceBoosterWords(removeStopWords(replaceNegatingWords(replaceBoosterWords(boostUpperCase(x)))))), **0.5**, **0.0**, **0.31063506398117546**, **0.0**, **0.0**, **0.0**, **0.0**)
2	polaritySumAVGUsingWeights(removeAllPonctuation(replaceBoosterWords(removeStopWords(replaceNegatingWords(x)))), **0.0**, **0.0**, **1.6561909157376327**, positiveWordsQuantity(replaceNegatingWords(removeAllPonctuation(x))), negativeWordsQuantity(x), **0.0**, **0.0**)
3	polaritySumAVGUsingWeights(replaceBoosterWords(removeAllPonctuation(removeStopWords(replaceNegatingWords((x))))), **0.5**, **0.0**, **0.5**, positiveWordsQuantity(removeStopWords(replaceBoosterWords(removeLinks(removeAllPonctuation(replaceBoosterWords(replaceBoosterWords(removeLinks(removeAllPonctuation(replaceNegatingWords(removeAllPonctuation(replaceBoosterWords(removeAllPonctuation(removeAllPonctuation(replaceNegatingWords(x))))))))))))))), mul(mul(**0.5**, negativeWordsQuantity(x)), mul(**0.5**, **0.5**)), **0.0**, **0.0**)

Resultados
3 models evaluated - 7 dictionaries

AVG All F1 SemEval: **60.11**
Best All F1 value: **61.01**
Desvio padrão: **0.64**

Algumas observações:

- Geração em que foi obtido o melhor indivíduo
  - Modelo 1: geração 40
  - Modelo 2: geração 38
  - Modelo 3: geração 44

V2.2 - Testes 2

Usando somente o valor discreto 0.0 e mantendo os valores reais [0,2]

#	Indivíduo
1	polaritySumAVGUsingWeights(removeStopWords(removeAllPonctuation(replaceNegatingWords(replaceBoosterWords(x)))), sub(**1.4917314878762928**, **1.0666968653301865**), if_then_else(hasURLs(removeAllPonctuation(removeStopWords(x))), **1.695040482181927**, **0.0**), if_then_else(hasURLs(removeLinks(replaceNegatingWords(removeAllPonctuation(x)))), **0.5791933567949965**, **0.252301850185894**), if_then_else(hasURLs(removeLinks(boostUpperCase(removeAllPonctuation(replaceNegatingWords(x))))), mul(**1.4917314878762928**, **1.5050939233561567**), sub(**1.4917314878762928**, **1.4917314878762928**)), **0.0**, **0.0**, **0.0**)
2	polaritySumAVGUsingWeights(removeAllPonctuation(replaceNegatingWords(boostUpperCase(removeStopWords(replaceBoosterWords(removeLinks(x)))))), **0.4405453203256887**, **0.0**, **0.2494501229744468**, **0.0**, **0.0**, **0.0**, **0.0**)
3	if_then_else(hasURLs(x), add(sub(sub(**0.0**, add(negativeWordsQuantity(boostUpperCase(removeStopWords(removeStopWords(removeAllPonctuation(x))))), mul(**0.0**, **0.0**)), add(negativeWordsQuantity(removeStopWords(boostUpperCase(removeStopWords(removeStopWords(removeAllPonctuation(x)))))), mul(**0.0**, **0.0**))), positiveWordsQuantity(replaceNegatingWords(x))), add(polaritySumAVGUsingWeights(replaceNegatingWords(removeLinks(removeAllPonctuation(removeStopWords(x))))), **0.32170186496512987**, **0.0**, **1.7518934561906048**, **0.32170186496512987**, **0.0**, **0.0**, **0.0**), add(sub(add(**0.0**, mul(**0.0**, **0.0**)), add(negativeWordsQuantity(boostUpperCase(removeStopWords(removeAllPonctuation(x)))), positiveWordsQuantity(x))), positiveWordsQuantity(x))))

3 models evaluated - 7 dictionaries

AVG All F1 SemEval: **60.8**
Best All F1 value: **61.05**
Desvio padrão: **0.26**

Algumas observações:

- Geração em que foi obtido o melhor indivíduo
  - Modelo 1: geração 45
  - Modelo 2: geração 33
  - Modelo 3: geração 38

V2.3 - Testes 3

Usando uma mutação especial somente para os pesos dos dicionários (mutEphemeral)

#	Indivíduo
1	polaritySumAVGUsingWeights(replaceNegatingWords(removeAllPonctuation(replaceNegatingWords(removeAllPonctuation(removeStopWords(replaceBoosterWords(replaceNegatingWords(replaceNegatingWords(boostUpperCase(replaceNegatingWords(x)))))))))), **0.3033264031723448**, **0.0**, **0.18051739429602653**, **0.0**, **0.0**, if_then_else(False, **0.24205151141961956**, **0.0**), **0.0**)
2	polaritySumAVGUsingWeights(removeAllPonctuation(replaceBoosterWords(removeAllPonctuation(removeAllPonctuation(removeStopWords(removeAllPonctuation(removeAllPonctuation(replaceBoosterWords(removeAllPonctuation(replaceNegatingWords(removeAllPonctuation(replaceNegatingWords(replaceNegatingWords(replaceNegatingWords(removeLinks(x)))))))))))))), **0.2129113158880589**, **0.0**, **0.15868236997005292**, **0.0**, **0.0**, **0.0**, **0.0**)
3	polaritySumAVGUsingWeights(boostUpperCase(removeStopWords(replaceNegatingWords(removeAllPonctuation(replaceBoosterWords(removeLinks(replaceNegatingWords(x)))))))), if_then_else(True, **0.2844238174009517**, **0.0**), **0.0**, mul(**0.8411836359183982**, **0.2728149468300072**), **0.0**, **0.0**, **0.0**, **0.0**)

3 models evaluated - 7 dictionaries

AVG All F1 SemEval: **60.98**
Best All F1 value: **61.02**
Desvio padrão: **0.04**

Algumas observações:

- Geração em que foi obtido o melhor indivíduo
  - Modelo 1: geração 24
  - Modelo 2: geração 27
  - Modelo 3: geração 21

Próximos passos

- Modificação dos parâmetros de população e gerações
  - Bons indivíduos estão sendo obtidos em gerações muito próximas do limite, o que sugere que ainda pode haver espaço para melhorias nos indivíduos se houverem mais gerações
  - Manter 25 mil ciclos (50 * 500)

	Melhor modelo			
	CBIC 2017	**V2.1**	**V2.2**	**V2.3**
Twitter 2013	51.93	60.87	**61.43**	60.81
Twitter 2014	45.07	59.56	59.47	**59.67**
Sarcasm	24.12	**42.75**	39.74	39.74
SMS	45.49	56.97	**57.12**	56.92
LiveJournal	60.52	67.86	68.26	**68.43**
TODAS	50.54	61.01	**61.05**	61.02

[1] Sean Luke, Gabriel Catalin Balan, and Liviu Panait. Population Implosion in Genetic Programming - Department of Computer Science, George Mason University 4400 University Drive MSN 4A5, Fairfax, VA 22030, USA