Using N-grams to detect Bots on Twitter

Juan Pizarro
Universitat Politècnica de València

jpizarrom@gmail.com

Bots and Gender Profiling, PAN at CLEF 2019
Lugano, Switzerland, September 10, 2019

Outline

- Task
- Dataset
- Methods
  - Preprocessing
  - Feature Extraction
  - Models
  - Parameter Optimization
- Results
- Other Methods
- Conclusions and Future Work

Bots and Gender Profiling

- Predict
    - Author: Bot or Human
    - Gender: male or Female
- Lang:
    - English
    - Spanish
- 100 tweets per author
- Evaluation
    - Accuracy average
- TIRA platform

Dataset

	(EN) English				(ES) Spanish			
	Bots	Female	Male	Total	Bots	Female	Male	Total
Training	2,060	1,030	1,030	4,120	1,500	750	750	3,000
Test	1,320	660	660	2,640	900	450	450	1,800
Total	3,380	1,690	1,690	6,760	2,400	1,200	1,200	4,800

Lang	Train all	Train	Dev
es	3000	2080	920
en	4120	2880	1240

Rangel, F., Rosso, P.: Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In: Cappellato, L., Ferro, N., Losada, D., M¨uller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2019)

Preprocessing

- Concat tweets by author
- Replace with single token
  - urls
  - user mentions
  - hashtags
- NLTK [1] TweetTokenizer

Based on [2]

[1] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.
[2] Daneshvar, S., Inkpen, D.: Gender Identification in Twitter using N-grams and LSA: Notebook for PAN at CLEF 2018. In: CEUR Workshop Proceedings. vol. 2125 (2018),

Preprocessing

```
xxnew “El niño no apre
```

```
l Whately xxnew “El tra
```

```
avés del despertar del Alma https://t.co/ueFzKsjkuL @PlataformaAutor #relatos #desarrolloPersonal #lectura https://t.
co/bvI6dVRGY6 xxnew “Un hombre soberbio es siempre difícil de contentar, porque siempre espera de los otros mucho má
s.”
Richard Baxter xxnew RT @dmrshal: RT @CitasDeEscritor: “Eres maestro de lo que has vivido,artesano de lo que estás vi
```

```
dad a través del despertar del alma <URLURL> <UsernameMention> <HASHTAG> <HASHTAG> <HASHTAG> <URLURL> xxnew “ un homb
re soberbio es siempre difícil de contentar, porque siempre espera de los otros mucho más . ” richard baxter xxnew rt
<UsernameMention>: rt <UsernameMention>: “ eres maestro de lo que has vivido, artesano de lo que estás viviendo y apr
endiz de lo que vivirás . ” rich … xxnew rt <UsernameMention>: <UsernameMention> <UsernameMention> excelente novela!
```

Feature Extraction

- Char N-grams (1, 6)
- Word N-grams (1, 3)
- Tf-idf

Using [1]

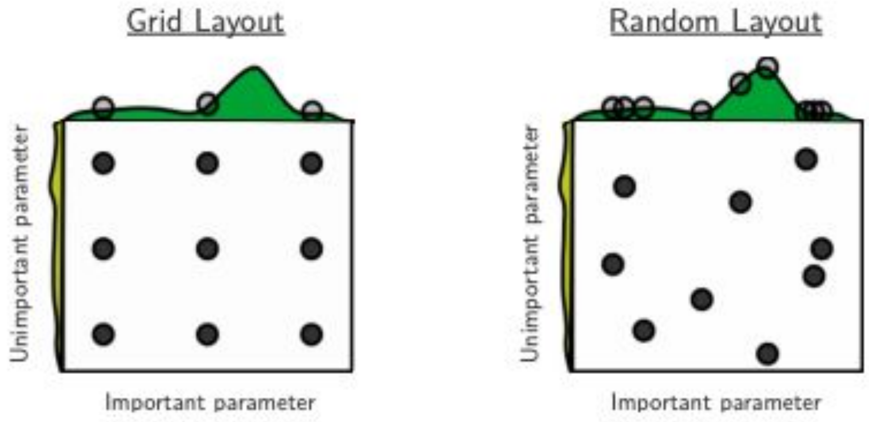
Models

- SVM LinearSVC
- MultinomialNB
- LogisticRegression

[1] Pedregosa et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, JMLR 12, 2825–2830 (2011)

Parameter Optimization

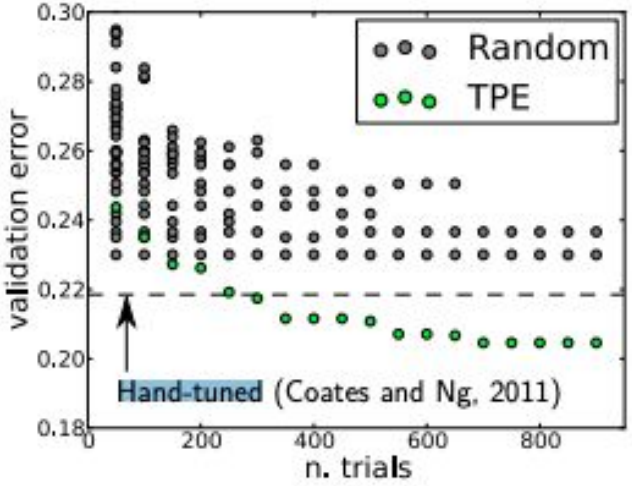
- Hand-tuning
- Grid Search
- Random Search [1]



[1] James Bergstra, Yoshua Bengio; Random Search for Hyper-Parameter Optimization.13(Feb):281−305, 2012.

Parameter Optimization

- Sequential model-based optimization (SMBO, also known as Bayesian optimization) with **hyperopt** [1,2]
  - Domain or Search Space
  - Objective Function
  - Optimization Algorithm



[1] Bergstra, J. Hyperopt: Distributed asynchronous hyperparameter optimization in Python. http://jaberg.github.com/hyperopt, 2013.
[2] Bergstra, J., Yamins, D., Cox, D. D. (2013) Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. To appear in Proc. of the 30th International Conference on Machine Learning (ICML 2013).

Parameter Optimization

Table 2. SVM hyperparameters.

Param	Values
C	hp.loguniform('C', np.log(1e-5), np.log(1e5))
tol	hp.loguniform('tol', np.log(1e-5), np.log(1e-2))
intercept_scaling	hp.loguniform('intercept_scaling', np.log(1e-1), np.log(1e1))

Table 3. MultinomialNB hyperparameters.

Param	Values
alpha	hp.loguniform('nb_alpha', -3, 5)

Table 4. Logistic Regression hyperparameters.

param	values
C	hp.choice('lr_C', [0.25, 0.5, 1.0])

Table 5. Feature representation hyperparamenters.

N-gram type	Param	Values
word	ngram_range	(1, 2),(1, 3),(2, 3)
word	max_df	0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0
word	min_df	0.0001, 0.001, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.1, 1, 2, 5
char	ngram_range	(1, 3),(1, 5),(2, 5),(3, 5),(1, 6),(2, 6)
char	max_df	0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0
char	min_df	0.0001, 0.001, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.1, 1, 2, 5

Parameter Optimization

- Precision tp/(tp+fp)
- Recall tp/(tp+fn)
- F-beta score

params.word.min_df	feats.params.word.ngram_range	loss	metric_classifier_accuracy	metric_classifier_fscore_macro	metric_classifier_fscore_micro
0.04	(2, 3)	-0.936290	0.936290	0.936221	0.936290
0.10	(1, 3)	-0.942742	0.942742	0.942696	0.942742
0.04	(1, 2)	-0.937097	0.937097	0.937031	0.937097
0.04	(1, 3)	-0.937903	0.937903	0.937842	0.937903
0.04	(1, 3)	-0.752419	0.752419	0.736253	0.752419

Results on Dev

lang	task	classifier	loss	feats	word.ngram_range	char.ngram_range
en	human_or_bot	LinearSVC	-0.945968	word_char	(1, 2)	(2, 6)
en	human_or_bot	LinearSVC	-0.945968	word_char	(2, 3)	(1,3)
en	human_or_bot	LinearSVC	-0.945968	word_char	(1, 2)	(2, 6)
en	human_or_bot	LinearSVC	-0.945968	word_char	(1, 2)	(2, 6)
en	human_or_bot	LinearSVC	-0.945161	word_char	(2, 3)	(1, 5)
en	gender	LinearSVC	-0.804839	word_char	(1,3)	(1,3)
en	gender	LinearSVC	-0.803226	word_char	(1, 2)	(1, 3)
en	gender	LinearSVC	-0.801613	word_char	(1, 2)	(1, 3)
en	gender	LinearSVC	-0.801613	word_char	(1, 2)	(1, 3)
en	gender	LinearSVC	-0.801613	word_char	(1, 2)	(1, 3)
es	human_or_bot	LinearSVC	-0.922826	word_char	(1, 3)	(3,5)
es	human_or_bot		-0.918478	word_char	(1, 2)	(1, 5)
es	human_or_bot	LinearSVC	-0.910870			
es	human_or_bot	LinearSVC	-0.909783			
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)
es	gender	LinearSVC	-0.691304	word_char	(1, 3)	(3,5)

Table 6. Best model parameters by language and task

Results on Test

Table 3. Accuracy per language and global ranking as average per language.

Ranking	Team	Bots vs. Human		Gender		Average
		EN	ES	EN	ES	
1	Pizarro	0.9360	**0.9333**	0.8356	**0.8172**	**0.8805**
2	Srinivasarao & Manu	0.9371	0.9061	0.8398	0.7967	0.8699
3	Bacciu et al.	0.9432	0.9078	0.8417	0.7761	0.8672
4	Jimenez-Villar et al.	0.9114	0.9211	0.8212	0.8100	0.8659
5	Fernquist	0.9496	0.9061	0.8273	0.7667	0.8624
6	Mahmood	0.9121	0.9167	0.8163	0.7950	0.8600
7	Ipsas & Popescu	0.9345	0.8950	0.8265	0.7822	0.8596
8	Vogel & Jiang	0.9201	0.9056	0.8167	0.7756	0.8545
9	Johansson & Isbister	**0.9595**	0.8817	0.8379	0.7278	0.8517
10	Goubin et al.	0.9034	0.8678	0.8333	0.7917	0.8491
11	Polignano & de Pinto	0.9182	0.9156	0.7973	0.7417	0.8432
12	Valencia et al.	0.9061	0.8606	**0.8432**	0.7539	0.8410
13	Kosmajac & Keselj	0.9216	0.8956	0.7928	0.7494	0.8399
14	Fagni & Tesconi	0.9148	0.9144	0.7670	0.7589	0.8388

Rangel, F., Rosso, P.: Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In: Cappellato, L., Ferro, N., Losada, D., M¨uller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2019)

Other Methods: NN Preprocessing

- Concat tweets by author
- Replace
  - urls
  - user mentions
  - hashtags
  - **number**
  - **demojify (demojize [1])**
- NLTK TweetTokenizer

```
print(demojify('👍 ddd', False))
print(demojify('🙈 ddd', False))
print(demojify('👍 ddd', True))
print(demojify('sss 🙈 1 :s_-s: ddd', True))
#print(remove_handles('@hola dd@ss'))
#print(modify_hashtags('#hola #hola'))
#print(remove_urls('http://hhhh.com'))
#print(remove_numbers('http://hhhh1.com'))
```

```
 :thumbs_up:  ddd
 :see-no-evil_monkey:  ddd
 xxemj  ddd
sss  xxemj  1  xxemj  ddd
```

[1] https://github.com/carpedm20/emoji/

Other Methods: NN Model

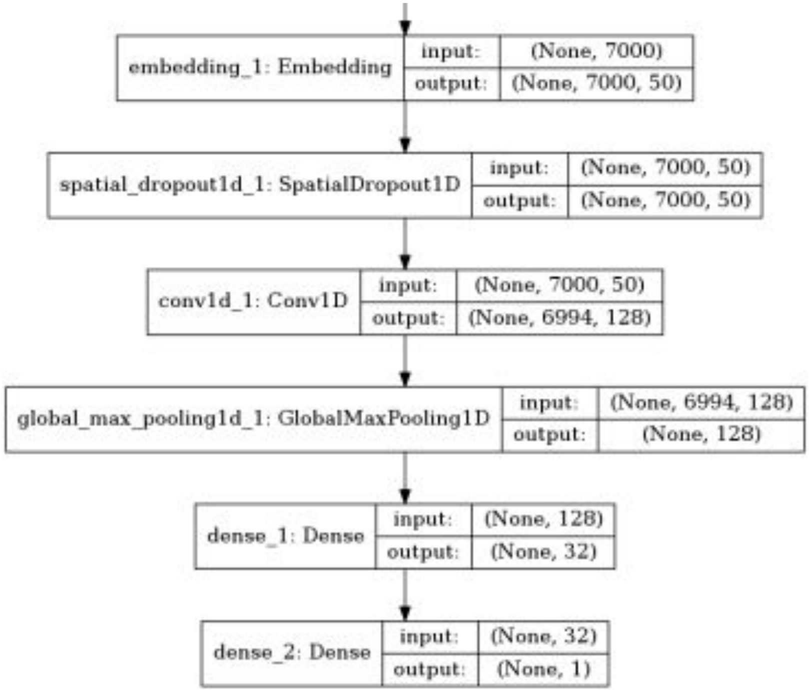
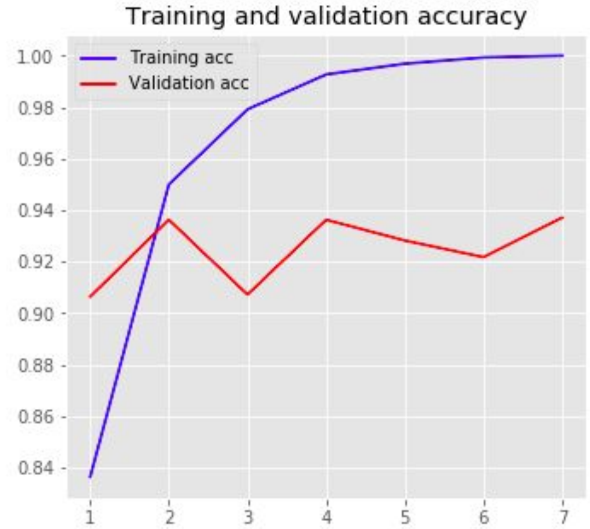
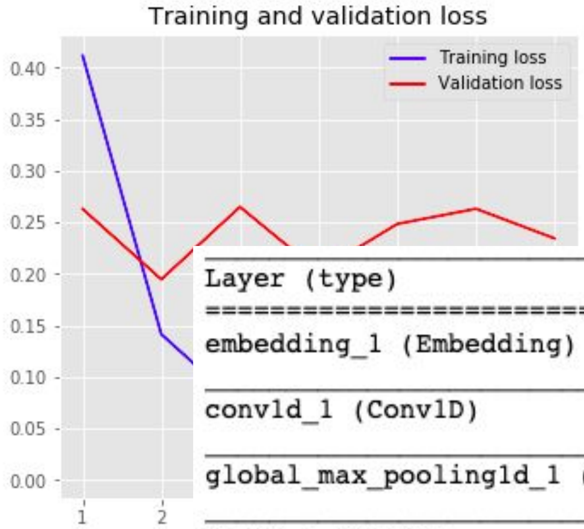


Fig. 3. Deep learning model.

Other Methods: Conv+Embedding

```
Training Accuracy: 1.0000
Testing Accuracy:  0.9371
```



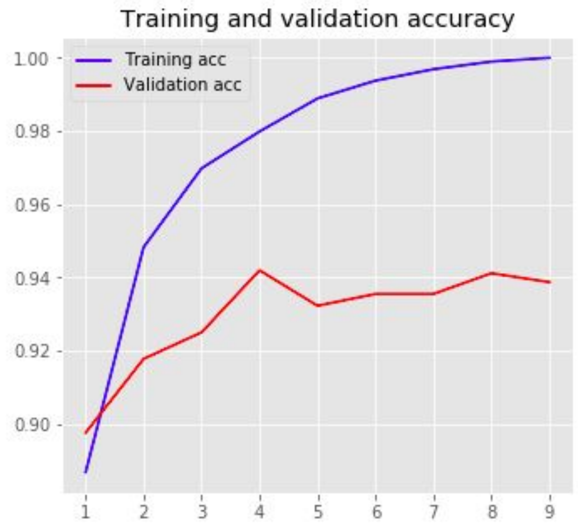


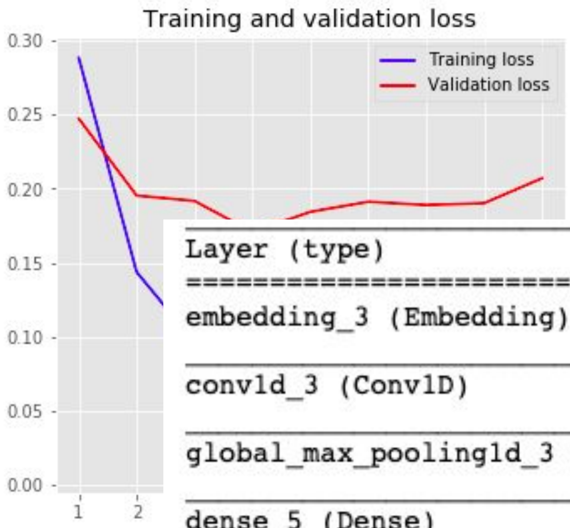
Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 7000, 50)	1000050
conv1d_1 (Conv1D)	(None, 6996, 128)	32128
global_max_pooling1d_1 (Glob	(None, 128)	0
dense_1 (Dense)	(None, 10)	1290
dense_2 (Dense)	(None, 1)	11

```
Total params: 1,033,479
Trainable params: 1,033,479
Non-trainable params: 0
```

Other Methods: Conv+Pretrained Embedding

```
Training Accuracy: 1.0000
Testing Accuracy:  0.9387
```



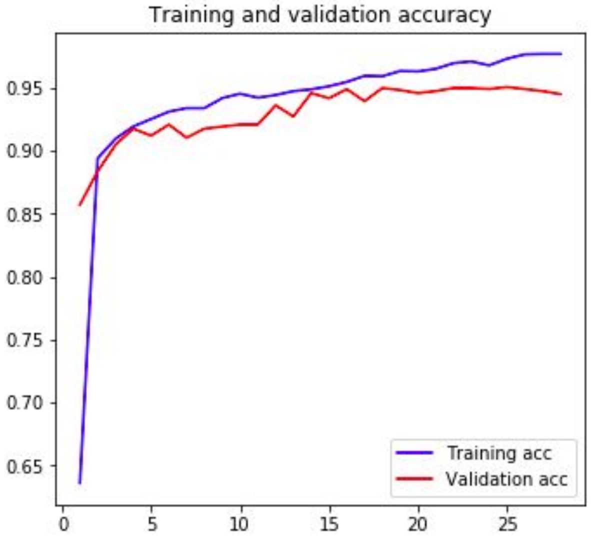


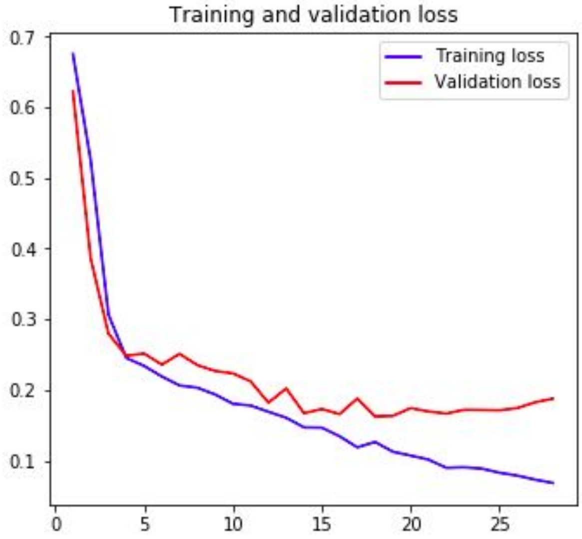

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 7000, 50)          1000050
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 6996, 128)         32128
_________________________________________________________________
global_max_pooling1d_3 (Glob (None, 128)               0
_________________________________________________________________
dense_5 (Dense)              (None, 10)                1290
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 11
=================================================================
Total params: 1,033,479
Trainable params: 33,429
Non-trainable params: 1,000,050
```

Other Methods: Conv+Embedding

- vocab_size=max_features+1
- embedding_dim=50
- maxlen=maxlen,
- embedding_matrix_weights=None
- trainable=False
- dropout1_rate=0.6
- conv1_filters=128
- conv1_kernel_size=7
- dropout2_rate=0.
- dense1_units=32
- dropout3_rate=0.

```
Epoch 00028: early stopping
Training Accuracy: 0.9892
Testing Accuracy:  0.9452
```





Conclusions

- SVM classifier with n-grams and TF-IDF features obtained good results
- Hyperparameter tuning is fundamental

Future Work

- why
- emoji
- lexicon
- word embeddings
- NN

Q&A

Environment Setup

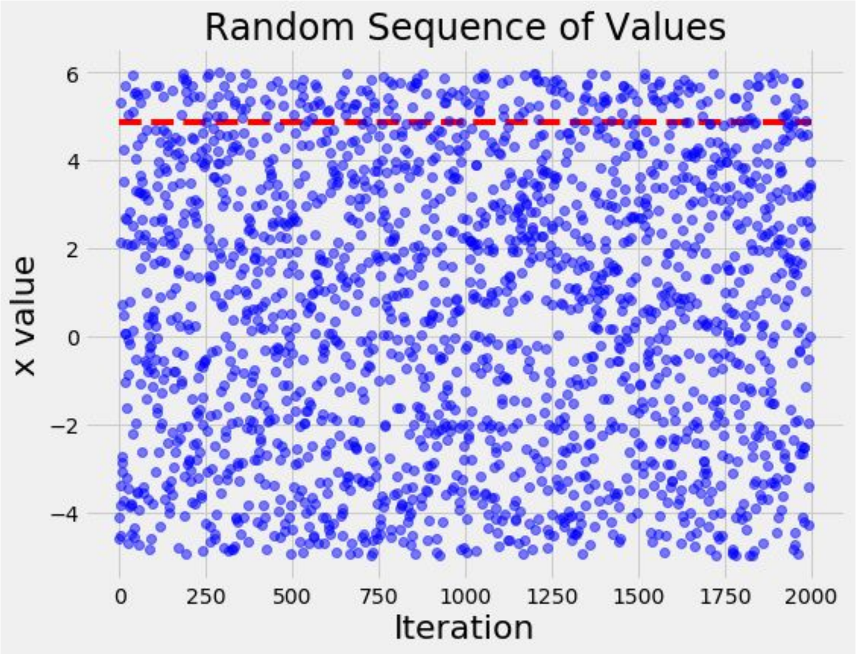
- NLTK [1]
- scikit-learn [2]
- hyperopt [3,4]
- Google Colaborator [5]
- Keras [6]

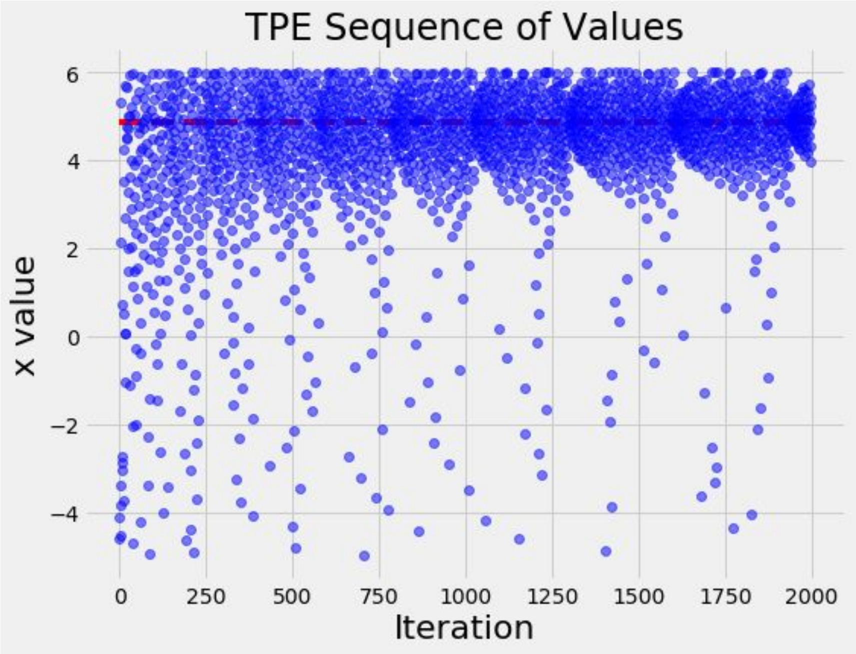
[1] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.
[2] Pedregosa et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, JMLR 12, 2825–2830 (2011)
[3] Bergstra, J. Hyperopt: Distributed asynchronous hyperparameter optimization in Python. http://jaberg.github.com/hyperopt, 2013.
[4] Bergstra, J., Yamins, D., Cox, D. D. (2013) Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. To appear in Proc. of the 30th International Conference on Machine Learning (ICML 2013).
[5] https://colab.research.google.com
[6] Chollet, F., et al.: Keras. https://keras.io (2015)

Other Methods

- build_model_emb_culstm_dense
- build_model_emb_lstm_dense
- build_model_emb_conv_maxpool_lstm_dense
- build_model_emb_conv_globmaxpool_dense_dense
- build_model_emb_sdrop_conv_maxpool_conv_maxpool_conv_maxpool_fln_dense_dense
- build_model_emb_globmaxpool_dense_dense
- build_model_emb_sdrop_fln_dense_dense
- build_model_emb_sdrop_biculstm_fln_sdrop_globmaxpool_dense
- build_model_emb_fln_dense_dense

Bayesian Optimization





https://towardsdatascience.com/an-introductory-example-of-bayesian-optimization-in-python-with-hyperopt-aae40fff4ff0

en-human?

{

#-0.9459677419354838 en human

'classifier': {'name': 'LinearSVC', 'params': {'C': 5153.874075307478, 'class_weight': 'balanced', 'dual': False, 'fit_intercept': True, 'intercept_scaling': 3.5918302677809204, 'loss': 'squared_hinge', 'max_iter': 1000, 'multi_class': 'ovr', 'penalty': 'l2', 'random_state': 2, 'tol': 0.0009950531254749422, 'verbose': False}},

'feats': {'name': 'word_char', 'params': {'char': {'max_df': 0.7, 'min_df': 0.02, 'ngram_range': (1, 3)}, 'word': {'max_df': 0.6, 'min_df': 0.1, 'ngram_range': (2, 3)}}}

}

en-gender

{

#-0.8 en gender

'classifier': {'name': 'LinearSVC', 'params': {'C': 14.332165053225301, 'class_weight': None, 'intercept_scaling': 0.215574951334565, 'loss': 'squared_hinge', 'max_iter': 2000, 'random_state': 42, 'tol': 3.798724613314342e-05}},

'feats': {'name': 'word_char', 'params': {'char': {'max_df': 0.7, 'min_df': 0.02, 'ngram_range': (1, 3)}, 'word': {'max_df': 0.7, 'min_df': 0.04, 'ngram_range': (1, 3)}}}

}

es-human?

{

-0.9228260869565217 es human

'classifier': {'name': 'LinearSVC', 'params': {'C': 5153.874075307478, 'class_weight': 'balanced', 'dual': False, 'fit_intercept': True, 'intercept_scaling': 3.5918302677809204, 'loss': 'squared_hinge', 'max_iter': 1000, 'multi_class': 'ovr', 'penalty': 'l2', 'random_state': 2, 'tol': 0.0009950531254749422, 'verbose': False}},

'feats': {'name': 'word_char', 'params': {'char': {'max_df': 0.8, 'min_df': 5, 'ngram_range': (3, 5)}, 'word': {'max_df': 0.7, 'min_df': 0.04, 'ngram_range': (1, 3)}}}

}

es-genger

{

-0.691304347826087 es gender

'classifier': {'name': 'LinearSVC', 'params': {'C': 83.52500216960948, 'class_weight': 'balanced', 'intercept_scaling': 0.40890443833718515, 'loss': 'hinge', 'max_iter': 2000, 'random_state': 42, 'tol': 0.005399650774898681 4}},

'feats': {'name': 'word_char', 'params': {'char': {'max_df': 0.7, 'min_df': 5, 'ngram_range': (3, 5)}, 'word': {'max_df': 0.6, 'min_df': 0.04, 'ngram_range': (1, 3)}}}}

Feature Extraction

- Char N-grams (1, 6)
- Word N-grams (1, 3)
- Tf-idf

Using [1]

[1] Pedregosa et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, JMLR 12, 2825–2830 (2011)

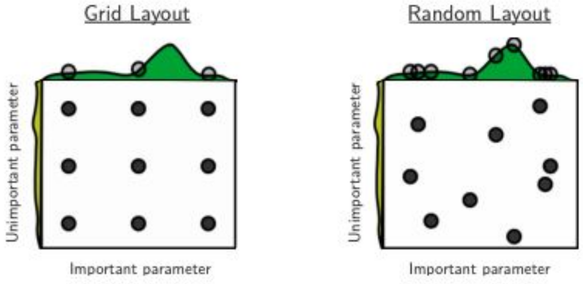
Models

- SVM LinearSVC
- MultinomialNB
- LogisticRegression

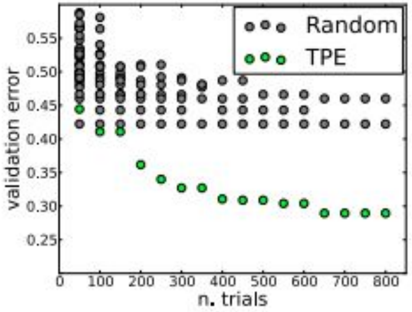
Using [1]

[1] Pedregosa et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, JMLR 12, 2825–2830 (2011)

Parameter Optimization



- Hand-tuning
- Grid Search
- Random Search [1]
- Sequential model-based optimization (SMBO, also known as Bayesian optimization) with **hyperopt** [2,3]
    - Domain or Search Space
    - Objective Function
    - Optimization Algorithm



[1] James Bergstra, Yoshua Bengio; Random Search for Hyper-Parameter Optimization.13(Feb):281−305, 2012.
[2] Bergstra, J. Hyperopt: Distributed asynchronous hyperparameter optimization in Python. http://jaberg.github.com/hyperopt, 2013.
[3] Bergstra, J., Yamins, D., Cox, D. D. (2013) Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. To appear in Proc. of the 30th International Conference on Machine Learning (ICML 2013).

Results

	BOTS vs. HUMAN		GENDER	
	en	es	en	es
pan19-author-profiling-test-dataset1-2019-03-20	0.9394	0.9278	0.7879	0.7611
pan19-author-profiling-test-dataset2-2019-04-29	0.9360	0.9330	0.8356	0.8172
MAJORITY	0.5000	0.5000	0.5000	0.5000
RANDOM	0.4905	0.4861	0.3716	0.3700
LDSE	0.9054	0.8372	0.7800	0.6900

Table 7. Results in the test set