German Tweets Geolocation

# Problem spec

Given a dataset consisting of tweets and their geographical coordinates of the location where they originated as labels, the coordinated should be inferred for new, unlabeled tweets.

# Data

The data is given in standard csv format where a row consists of an Id, the latitude, the longitude and the text of a tweet. Example:

|  |  |  |  |
| --- | --- | --- | --- |
| 119165 | 51.810067114093954 | 10.191330935251802 | "Seit d Vase: ""Wenn ich kaputt gang, bringt das 1 Jahr Unglück."" Antwortet de Spiegel: ""Das isch doch gar nüt. Wenn ich kaputt gang, bringt das 7 Jahr Pech!"" Dadruf fangt s Kondom a lache. S'Atomkraftwerk au 😋 Uuuh nöd schlecht 👍 seisch anere frau sie isch fett und du hesch dis lebe lang unglück (au wenn i dem fall nüm lang lebsch)" |

# Preprocessing

Preprocessing has been applied to the data in the form of: removal of special characters – like emojis and emoticons, symbols, chinese characters etc.; removal of digits; removal of punctuation; lower-casing; lemmatizing; removal of stop-words. Other types of preprocessing have been attempted but have been deemed ineffective by result, such as converting the emojis to words describing them (i.e. 😋 -> smiley\_face\_tongue\_out) and removal of most and least common words in the set.

Example of a tweet before and after preprocessing:

**Before:** *Seit d Vase: ""Wenn ich kaputt gang, bringt das 1 Jahr Unglück."" Antwortet de Spiegel: ""Das isch doch gar nüt. Wenn ich kaputt gang, bringt das 7 Jahr Pech!"" Dadruf fangt s Kondom a lache. S'Atomkraftwerk au 😋 Uuuh nöd schlecht 👍 seisch anere frau sie isch fett und du hesch dis lebe lang unglück (au wenn i dem fall nüm lang lebsch)*

**After:** *kaputt gang bringen jahr unglück antworten isch gar nüt kaputt gang bringen jahr dadruf fangen lache satomkraftwerk schlecht seisch anere frau isch fett hesch leben langen unglück fall langen lebsch*

# Data representation

Several encodings have been used for the data, comparing performance, but also in relation with the model applied. These are Bag-Of-Words, TF-IDF and word, respectively character embeddings. Out of the aforementioned methods, Bag-Of-Words and TF-IDF are comparable on the same models, and result have shown a slight improvement (~0.2 MSE decrease) when using TF-IDF. (The string kernels method has been attempted but dropped due to lack of computing power)

# Models

The models with the best results have been a nu-SupportVectorRegressor, a character-level Convolutional Neural Network and a Bidirectional LSTM with partially trained word embeddings. Other regression models have been applied to the data in BOW or TF-IDF representation. These include: Stochastic Gradient Descent, Ridge, Lasso, Bayesian Ridge, Kernel Ridge, Passive Aggressive.

CNN architecture:

* Embedding(63, 63)
* Conv1D(filters = 256, filter size = 7, activation = RELU)
* MaxPooling1D(3),
* Conv1D(256, 7, RELU)
* MaxPooling1D(3),
* Conv1D(256, 3, RELU)
* Dropout(0.5),
* Conv1D(256, 3, RELU)
* Dropout(0.5),
* Conv1D(256, 3, RELU)
* Dropout(0.5),
* Conv1D(256, 3, RELU)
* Dropout(0.5),
* Conv1D(256, 3, RELU)
* Dropout(0.5),
* Conv1D(256, 3, RELU)
* Dropout(0.5),
* Conv1D(256, 3, RELU)
* Dropout(0.5),
* Conv1D(256, 3, RELU)
* MaxPooling1D(3),
* Flatten(),
* Dense(1024, RELU)
* Dense(1024, RELU)
* Dense(2, RELU)

LSTM architecture:

* Embedding(497, 300)
* Bidirectional(LSTM(64))
* Bidirectional(LSTM(64))
* Flatten()

# Results

The metrics observed have been MSE and MAE. The metric optimized was MSE.

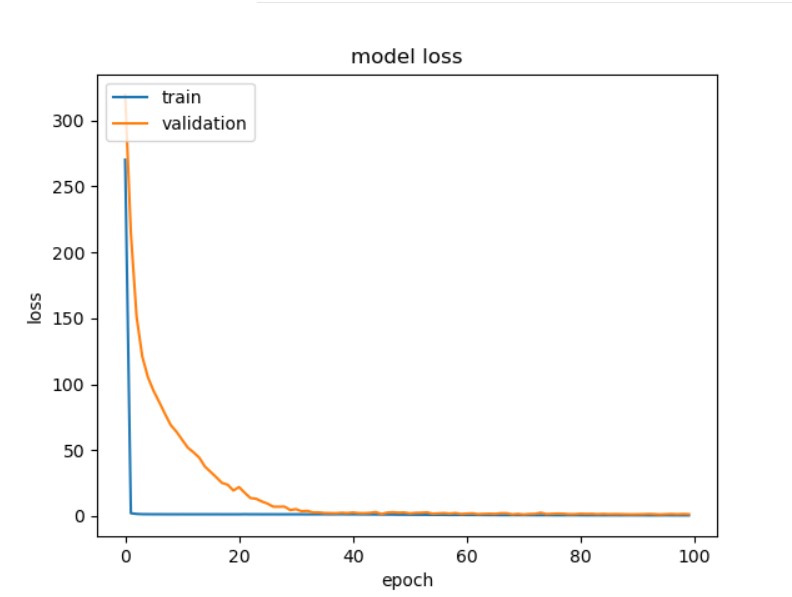
## Char-level CNN

Final Score: 0.74871

Validation scores:

|  |  |  |
| --- | --- | --- |
| Char CNN | MAE | MSE |
| Latitude | 0.888477768354080 | 1.104162363651997 |
| Longitude | 1.164460620534860 | 2.073622408656586 |
| Average | 1.026469194444470 | 1.588892386154292 |

Training loss:



## Bidirectional LSTM

Final Score: 0.74871

Validation scores:

|  |  |  |
| --- | --- | --- |
| BiLSTM | MAE | MSE |
| Latitude | 0.801884927227194 | 1.057395777483923 |
| Longitude | 1.060289460098271 | 2.020592883099761 |
| Average | 0.931087193662732 | 1.538994330291842 |

Training loss:

# 

## Nu-SVR

2 Grid searches have been conducted for optimizing the parameters of 2 nuSVR models, one predicting the latitude and one predicting the longitude separately and the best parameters found were:

|  |  |  |
| --- | --- | --- |
| Grid Search | Latitude | Longitude |
| C | 0.1 | 0.001 |
| nu | 0.3 | 0.7 |

However, it has proven more efficiently (probably due to the scikit-learn implementation) to train a single Multioutput Regressor with a NuSVR backbone with parameters C=1.0 and nu=0.5.

Final score: 0.72270

|  |  |  |
| --- | --- | --- |
| Nu-SVR | MAE | MSE |
| Latitude | 0.503936374886482 | 0.654933202981718 |
| Longitude | 0.987841194987705 | 1.276784455945984 |
| Average | 0.7458887849370935 | 0.965858829463851 |

## Other linear regressors

All 0ther regressors used (Stochastic Gradient Descent, Ridge, Lasso, Bayesian Ridge, Kernel Ridge, Passive Aggressive, Random Forest) have performed close to the Nu-SVR model or poorer, with scores ranging from 1.0 and even up to 700 (in the case of Kernel Ridge and Random Forest, for example).

# Conclusion

As a conclusion, it can be noted that the BiLSTM, NuSVR and character CNN have best fitted the data, results which are consistent with other published work in the field (*Gaman, Mihaela & Ionescu, Radu Tudor - Combining Deep Learning and String Kernels for the Localization of Swiss German Tweets).* It can be mentioned that better preprocessing, coordinates normalization or optimizing MAE score may possibly yield better results.