

Location NER from Tweets

Stone Fang (Student ID: 19049045)
Computers and Information Sciences
Auckland University of Technology
Auckland, New Zealand
fkn7060@autuni.ac.nz

I. METRICS

- exact match: both entity type and boundaries are matched
- relaxed match

II. ALGORITHMS

A. Knowledge/Rule-based

hand-crafted rules. rely on lexicon resources and domain specific knowledge.

pros: do not require annotated training data high precision

cons: need domain experts construct and maintain the knowledge resources low recall

B. Unsupervised and bootstrapped systems

earliest systems,

pros: required very minimal training data.

C. Feature-engineered supervised learning

learn models from data, but need manual feature engineering common approaches:

- Hidden Markov Models (HMM)
- Support Vector Machines (SVM)
- Conditional Random Fields (CRF)
- decision trees

D. Feature-inferring neural networks

features are also learned from data, e.g. embeddings

1) Representation(Embedding):

- Word level Architecture
- Character level Architecture
- Character+Word level architectures
- Character + Word + affix model

2) Context encoder:

- CNN
- RNN -> LSTM -> bi-LSTM -> window bi-LSTM
- Recursive neural networks
- Neural Language Models
- Attention -> Transformer -> BERT/GPT/ELMo

3) Tag decoder:

- Multi-Layer Perceptron + Softmax
- Conditional Random Fields
- RNN
- Pointer Networks

III. ALGORITHM ANALYSIS

A. choosed algorithms

This paper conducted a two-stage location extraction system for tweets, where a location predictor followed by an extractor. In the first stage, a classification model predicts if an input text contains any location or not, where only inputs with positive predictions will be feed into next stage. In the second stage, a combined NER model from various tools extracts locations from the text filtered out by previous stage. By a variety of experiments, the author demonstrated that: 1) combining various NER tools can improve F-score of the model; 2) the final precision of location extraction can be improved if a proper predictor is set before it.

1) *Location Extraction*: This is the second stage in the whole pipeline but it was studied first. In this stage, a compound model is combined by some of the three NER tools: Ritter tool, the Gate NLP framework (Gate) and the Stanford NER. In addition, in some combinations the result will be post-filtered by DBPedia. To achieve this, each tool is applied to input texts to extract locations, then the individual outputs are merged as final result. To filter the locations, locations extracted from NER models are kept only if they appears in DBPedia.

In the experiment, combined NER extractors are compared to the Ritter tool as the baseline. The results showed that combining with either Gate or Stanford NER could improve the model performance under the evaluation of F-score. Moreover, the DBPedia post-filtering dramatically increased precision but also caused great decline in recall. The best F-score on Ritter dataset was obtained by "Ritter tool + Stanford NER + DBPedia", and "Ritter tool + Stanford NER" for MSM2013 dataset.

2) *Location Prediction*: In location prediction stage, the author first showed the usefulness of prior predictors by passing tweets through a perfect (that is, 100% correct results) location predictor before feeding into extractor, where the precision were significantly improved with the recall unchanged. Though perfect prediction models are impossible in real-world applications, this is a reasonable proof-of-concept. Then several features were defined for predictive models, including:

- **Geography gazetteer**: if any word is included by a geography gazetteer (Gate was chosen in the experiment)
- **Prep**: if input text contains any of the 7 prepositions regarding to place (*at, in, on, from, to, toward, towards*)

- **Prep + PP**: if input text contains a preposition directly followed by a proper noun (PP)
- **Place + PP**: if any word of place (*town, city, state, region, department, country*) appears directly after or before a proper noun
- **Time**: if any word regarding to time exists, including *today, tomorrow, weekend, tonight*, as well as all the names of months and the days of a week
- **DefArt + PP**: if any proper noun appears after the definitive article “the”
- **Htsh**: if the tweet contains any hash tag
- **PP, Adj, Verb**: the counts of proper nouns, adjectives and verbs in the text

After features generated, three machine learning algorithms are trained on Ritter and MSM2013 datasets with 10-folds cross validation, namely Naive Baiyes (NB), Support Vector Machine (SVM) and Random Forest (RF). Experiments under various parameters have been carried out.

The author found that some features are more useful than others. Though there are minor difference between performances on two datasets, most significant common features includes: Geography gazetteer, Prep+PP, PP, and Place+PP. On Ritter dataset, the best F-score was 65% with an accuracy of 94%, achieved by RF under threshold 0.5. On MSM2013 dataset, the best F-score was 61% along with accuracy 84%, also obtained by RF but with a different threshold 0.75. Naive Bayes is also a comparable solution which is preferred over SVM.

After that, various predictors are applied before extractor for evaluation. In this experiment, only Ritter tool was used instead of combined tools, and RF models with different thresholds were used as predictors. The result showed that the precision was significantly improved, while the recall decreased due to prediction error. Another advantage of predictor is reducing the number of tweets that are fed into extractor.

3) *Summary*: To summarise, this work showed that 1) combining different NER tools is a reasonable way to improve the location extraction model metrics such as F-score; 2) a preceding location predictor can benefit the precisions of final result but reduce the recall. Furthermore, from the experiment of perfect predictor conclusions can be drawn that better prediction model will mitigate the recall decline.

IV. PROPOSED APPROACH

A. Algorithm Description

Our proposed NER algorithm for locations on Twitter follows the two-stage model as studied by Hoang and Mothe [1], but the machine learning models in each stage will be replaced by deep learning ones which was employed by Mao, Thakur, Sparks, *et al.* [2]. The overall process of our proposal is shown as Fig 1.

- 1) *Location Prediction*:
- 2) *Location Tagging*:
- 3) *Post-processing*:



Fig. 1. Proposed Algorithm: Two-Stage Location Extraction

REFERENCES

- [1] T. B. N. Hoang and J. Mothe, “Location extraction from tweets,” *Information Processing & Management*, vol. 54, no. 2, pp. 129–144, Mar. 2018, ISSN: 03064573. DOI: [10.1016/j.ipm.2017.11.001](https://doi.org/10.1016/j.ipm.2017.11.001).
- [2] H. Mao, G. Thakur, K. Sparks, J. Sanyal, and B. Bhaduri, “Mapping near-real-time power outages from social media,” *International Journal of Digital Earth*, vol. 12, no. 11, pp. 1285–1299, 2019. DOI: [10.1080/17538947.2018.1535000](https://doi.org/10.1080/17538947.2018.1535000). [Online]. Available: <https://doi.org/10.1080/17538947.2018.1535000>.