# University of Arizona at SemEval-2019 Task 12: Deep-Affix Named Entity Recognition of Geolocation Entities

**Vikas Yadav, Egoitz Laparra, Ti-Tai Wang, Mihai Surdeanu, Steven Bethard**
University of Arizona
{vikasy,laparra,twang03,msurdeanu,bethard}@email.arizona.edu

## Abstract

We present the Named Entity Recognition (NER) and disambiguation model used by the University of Arizona team (UArizona) for SemEval 2019 task 12. We achieved fourth place on tasks 1 and 3. We implemented a deep-affix based LSTM-CRF NER model for task 1, which utilizes only character, word, prefix and suffix information for the identification of geolocation entities. Despite using just the training data provided by task organizers and not using any lexicon features, we achieved 78.85% strict micro F-score on task 1. We used the unsupervised population heuristics for task 3 and achieved 52.99% strict micro-F1 score in this task.

## 1 Introduction

Geoparsing is the task of detecting geolocation phrases in unstructured text and normalizing them to a unique identifier, e.g. GeoNames[1] IDs. Although many automatic resolvers have been released in the past years, their performance fluctuates when applied to different domains (Gritta et al., 2018b). Most have also not been applied to and evaluated on scientific publications. The SemEval 2019 Shared Task 12: Toponym Resolution in Scientific Papers (Weissenbacher et al., 2019) aims to boost the research on geoparsing for the scientific domain by focusing on epidemiology journal articles.

The task includes three sub-tasks: toponym detection, toponym disambiguation, and end-to-end toponym resolution. The first one requires participants to detect the text boundaries of all toponym mentions in articles. In toponym disambiguation, the toponym mentions are known, and the resolver has to align them to their precise coordinates through GeoNames IDs. For the last subtask, the resolver must perform both detection and disambiguation.

In this paper, we present the description of our system for SemEval 2019 Shared Task 12, in which we focus mainly on toponym detection. For this sub-task, we propose a recurrent neural network that combines word, character and affix information. By making use of the baseline provided by the organizers for toponym disambiguation, we also obtain results for the end-to-end sub-task.

## 2 Related Work

Toponym detection and resolution has been widely studied, and various systems (Gritta et al., 2018b) have been proposed for these tasks. Toponym detection has been implemented on texts from various sources like social media (Karagoz et al., 2016), PubMed articles (Magge et al., 2018) etc. Various named entity recognition (NER) systems including rule-based (Gritta et al., 2018b), machine learning-based (Karagoz et al., 2016), and deep learning-based (Magge et al., 2018) have been implemented for detecting toponyms.

The disambiguation step has been tackled previously using both supervised models and unsupervised heuristic based approaches. For example, Turton (2008) presented a rule based system for disambiguating locations from PubMed abstracts. Weissenbacher et al. (2015) presented results from *Population* and *Distance* heuristics (discussed in Section 4.3) for the disambiguation task on PubMed articles. The authors also presented an SVM model with population, distance and set of meta-data as input which achieved higher performance than both the individual heuristics. Gritta et al. (2018a) used a feedforward neural network approach for the disambiguation of geolocations.

[1] http://www.geonames.org/



*Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, pages 1319–1323
Minneapolis, Minnesota, USA, June 6–7, 2019. 

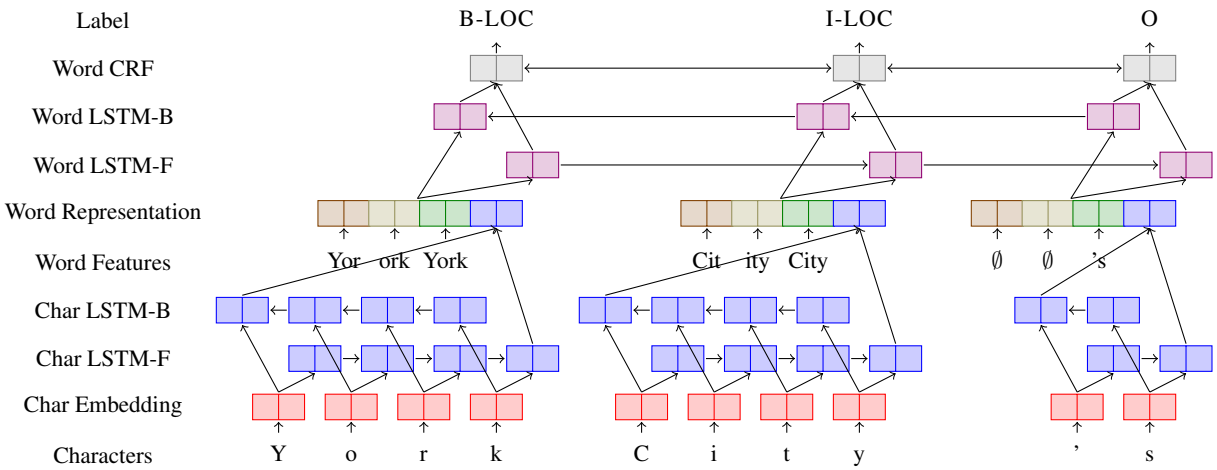


Figure 1: Word+character+affix neural network architecture from Yadav et al. (2018).

## 3 Data and Baseline

The corpus of the task is composed of 150 journal articles downloaded from PubMed Central. After removing the author names, acknowledgments and references, titles and body text were fully annotated. The annotators identified and labelled toponyms with their corresponding coordinates according to GeoNames. For cases not found in GeoNames, they used Google Maps and Wikipedia. If the coordinates of a toponym were not available in any of these resources the special value N/A was used. The data is provided in Brat format (Stenetorp et al., 2012). The organizers also released a strong baseline that combines the model by Magge et al. (2018) for toponomy detection and the *Population* heuristic described in (Weissenbacher et al., 2015) for disambiguation.[2]

## 4 Approach

### 4.1 Preprocessing

We used the tokenizer included in the baseline provided by the organizers as we observed it provided the best final results among other options (see Section 5.3). Again using baseline system preprocessing codes, we converted the data into CoNLL 2003 format (Tjong Kim Sang and De Meulder, 2003) for task 1. Following our prior work (Yadav and Bethard, 2018), we have used a BIO encoding instead of the IO encoding provided by the baseline system.

[2] https://github.com/amagge/semeval-ffnn-baseline

### 4.2 Toponym Detection

We used the model proposed by Yadav et al. (2018) for Named Entity Recognition (NER), shown in figure 1, which uses character, word and affix information. In this architecture, a word is represented by concatenating its word embedding, an LSTM representation over the characters of the word, and learned embeddings for prefixes and suffixes of the word[3]. Then another LSTM is used at the sentence level to give a contextual representation of each word. These representations of words in the sentence are given to a CRF layer to finally predict the NER label.

### 4.3 Toponym Resolution

Weissenbacher et al. (2015) presented two heuristics for disambiguation of geolocation - *Population* and *Distance*. These two heuristics are often used as features with other meta-data such as the user location meta-data in a Twitter account (Zhang and Gelernter, 2014), GenBank meta-data (Weissenbacher et al., 2015), etc.

In the *Population* heuristic, the system simply assigns the geonameID of the most populous[4] candidate for the current location. For the *Distance* heuristic, the system selects the candidate which is at the minimum distance from all candidates of all other toponyms in the same document. Many previous works (Weissenbacher et al., 2015; Zhang and Gelernter, 2014; Weissenbacher et al., 2019) have shown that the most populous location is often referenced more in the text documents and performs

[3] The affix vocabulary consisted of all three-character affixes that occurred at least 50 times in the training data.

[4] Population retrieved from the GeoNames database.

better than the distance heuristics. Thus, we use the $Population$ heuristic as our disambiguation model.

## 5 Experiments

Using the original fully annotated training set, we achieved 77.3% strict micro-Fscore (mean performance of 3 runs) on the validation set. However, the organizers provided two additional large (but weakly) annotated NER datasets: $POS$, which contains sentences having at least 1 location phrase, and $NEG$, which has sentences with no mention of location entities. We experimented with both these datasets in both joint and transfer learning.

### 5.1 Joint Learning

In the joint learning experiment, we trained the model on a training set by concatenating the $POS$ data with the original training data. In this configuration, we achieved 81.4% strict micro-F score (mean performance of 3 runs) on the validation set, a 4 point improvement over the original experiment.

### 5.2 Transfer Learning

In this experiment, we first trained our model on just the $POS$ set and further fine tuned it on the original training data provided for the task. The intuition here was to use the weakly annotated data only to get a good initialization for the "real" training on the manually annotated data, rather than training on both together and possibly getting misled by the noise in the weakly annotated data. We achieved 83.7% strict micro-F score (mean performance of 3 runs) on the validation set. This is an improvement of 2.3 F over the simple joint learning experiment, and 6.4 F over the model using only the original training data.

### 5.3 Effects of Tokenization

The effect of tokenization on NER performance has been shown in the past (Akkasi et al., 2016; Xu et al., 2018). For this reason, we evaluated our model trained on the original training data, using various custom tokenization functions, and saw the strict micro-F1 score vary from 72% to 77% in the validation set.

The NLTK regexp tokenizer resulted in 70% strict F1-score. We wrote several rules to improve this tokenizer which further improved the performance by 4%.

| Parameter | value |
|---|---|
| Word embedding (GloVe) size | 300 |
| Character embedding size | 50 |
| Affix embedding size | 30 |
| Word LSTM hidden state size | 50 |
| Character LSTM hidden state size | 25 |
| Learning rate | 0.15 |
| Learning rate decay | 0.99 |
| Batch size | 100 |
| Optimizer | SGD |

Table 1: Hyperparameters for training the model.

However, the custom tokenization implemented by the shared task organizers in the baseline model performed the best, achieving 77% on the validation set when trained on just the original training data. In this case, we also wrote a few additional rules to improve the tokenization but achieved marginal improvements in the overall performance.

### 5.4 Hyperparameters

We trained the Yadav et al. (2018) model using the parameters in Table 1. For transfer learning from $POS$ data, we first trained the model for 40 epochs. We then retrained this model on the original training data for 80 epochs with 20 as the early stopping patience. After training on the original training data, we retrained this model on train+development data for another 40 epochs. For the final evaluation, we submitted the models at epoch = 25, 35 and 40. Epoch 35 achieved the best performance among the three submissions.

The software is available at `https://github.com/vikas95/Pref_Suff_Span_NN`.

## 6 Results

We achieved the $4^{th}$ position in both task 1 (toponym detection) and task 3 (end-to-end toponym resolution) as shown in tables table 2 and table 3, respectively. Although it has been shown previously that adding lexicon features improves the overall performance of several NER models (Yadav and Bethard, 2018; Gritta et al., 2018b), we have focused on extraction of context information using LSTMs over character, word and affixes of the word. Hence, our resource-independent NER model achieves competitive results, despite not using any dictionary information. Also, we have just used the training data provided by the task organizers and did not use any external training data or

| Team | strict Micro F | strict Macro F |
|---|---|---|
| DM_NLP | 89.13 | 91.61 |
| QWERTY | 83.33 | 87.10 |
| Newbee | 80.92 | 87.11 |
| Our model | 78.75 | 84.52 |
| THU_NGN | 74.96 | 83.23 |
| UNH | 73.12 | 81.93 |
| RGCL-WLV | 49.13 | 61.96 |
| NLP_IECAS | 64.85 | 74.82 |

Table 2: Results of subtask 1 – toponym detection. We include the best Micro F-score and best Macro F-score of each team from their final 3 runs. Our model is ranked fourth, despited the fact that it uses no external knowledge.

| Team | strict Micro F | strict Macro F |
|---|---|---|
| DM_NLP | 0.7291 | 0.7749 |
| QWERTY | 0.7128 | 0.7551 |
| Newbee | 0.6545 | 0.7355 |
| Our model | 0.5299 | 0.6487 |
| THU_NGN | 0.5156 | 0.6131 |
| NLP_IECAS | 0.5223 | 0.6019 |

Table 3: Results of subtask 3 - end-to-end toponym resolution. Our system is again ranked fourth.

lexicon resources.

We used the unsupervised *Population* heuristic which is fast and simple to implement for disambiguating toponyms. As shown by Weissenbacher et al. (2015), feeding features like population, distance, and other meta-data to machine learning models often achieved higher performances. However, as shown here, the *Population* heuristic serves as a strong baseline for this disambiguation task.

## 7 Future Work

We plan to include the following features in our current model:

- Part of Speech (POS) features – as per the annotations guidelines, locations that were used as adjectives were not labelled in the annotation process. We will explore the effect of adding POS feature representation to the word, character and affix representations.

- Inclusion of geoname dictionary – our current approach is resource independent. We will include dictionary features in the next version of our model, to understand how much signal can be inferred from local information, and how much must come from world knowledge.

- Using domain-specific embeddings – we relied on pretrained GloVe embeddings for our submissions. In future versions of our software, we will explore domain-specific embeddings, i.e., trained on scientific texts, as well as contextualized embeddings such as FLAIR (Akbik et al., 2018).

## 8 Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the World Modelers program, grant number W911NF1810014. Mihai Surdeanu declares a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies.

## References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Abbas Akkasi, Ekrem Varoğlu, and Nazife Dimililer. 2016. Chemtok: a new rule based tokenizer for chemical named entity recognition. *BioMed research international*, 2016.

Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. 2018a. Which melbourne? augmenting geocoding with maps. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1285–1296.

Milan Gritta, Mohammad Taher Pilehvar, Nut Limsopatham, and Nigel Collier. 2018b. Whats missing in geographical parsing? *Language Resources and Evaluation*, 52(2):603–623.

Pinar Karagoz, Halit Oguztuzun, Ruket Cakici, Ozer Ozdikis, Kezban Dilek Onal, and Meryem Sagcan. 2016. Extracting location information from crowdsourced social network data. *European Handbook of Crowdsourced Geographic Information*, 195.

Arjun Magge, Davy Weissenbacher, Abeed Sarker, Matthew Scotch, and Graciela Gonzalez-Hernandez. 2018. Deep neural networks and distant supervision for geographic location mention extraction. *Bioinformatics*, 34(13):i565–i573.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: A web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 102–107, Stroudsburg, PA, USA. Association for Computational Linguistics.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Ian Turton. 2008. A system for the automatic comparison of machine and human geocoded documents. In *Proceedings of the 5th Workshop on Geographic Information Retrieval*, pages 23–24. ACM.

Davy Weissenbacher, Arjun Magge, Karen O'Connor, Matthew Scotch, and Graciela Gonzalez. 2019. Semeval-2019 task 12: Toponym resolution in scientific papers. In *Proceedings of The 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Davy Weissenbacher, Tasnia Tahsin, Rachel Beard, Mari Figaro, Robert Rivera, Matthew Scotch, and Graciela Gonzalez. 2015. Knowledge-driven geospatial location resolution for phylogeographic models of virus migration. *Bioinformatics*, 31(12):i348–i356.

Dongfang Xu, Vikas Yadav, and Steven Bethard. 2018. Uarizona at the made1. 0 nlp challenge. *Proceedings of machine learning research*, 90:57.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158.

Vikas Yadav, Rebecca Sharp, and Steven Bethard. 2018. Deep affix features improve neural named entity recognizers. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 167–172.

Wei Zhang and Judith Gelernter. 2014. Geocoding location expressions in twitter messages: A preference learning method. *Journal of Spatial Information Science*, 2014(9):37–70.