Bi-directional Recurrent Neural Network Models for Geographic Location Extraction in Biomedical Literature

Arjun Magge[1,2] and Davy Weissenbacher[3] and Abeed Sarker[3] and Matthew Scotch[1,2] † and Graciela Gonzalez-Hernandez[3]

[1] *College of Health Solutions,* [2] *Biodesign Center for Environmental Health Engineering, Arizona State University, Tempe, AZ 85281, USA*

[3] *Department of Biostatistics, Epidemiology and Informatics, The Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA.*

†*E-mail: Matthew.Scotch@asu.edu*

Phylogeography research involving virus spread and tree reconstruction relies on accurate geographic locations of infected hosts. Insufficient level of geographic information in nucleotide sequence repositories such as GenBank motivates the use of natural language processing methods for extracting geographic location names (toponyms) in the scientific article associated with the sequence, and disambiguating the locations to their co-ordinates. In this paper, we present an extensive study of multiple recurrent neural network architectures for the task of extracting geographic locations and their effective contribution to the disambiguation task using population heuristics. The methods presented in this paper achieve a strict detection F_1 score of 0.94, disambiguation accuracy of 91% and an overall resolution F_1 score of 0.88 that are significantly higher than previously developed methods, improving our capability to find the location of infected hosts and enrich metadata information.

Keywords: Named Entity Recognition; Toponym Detection; Toponym Disambiguation; Toponym Resolution; Natural Language Processing; Deep Learning;

1. Introduction

Nucleotide sequence repositories like GenBank contain millions of records from various organisms collected around the world that enables researchers to perform phylogenetic tree and spread reconstruction. However, a vast majority of the records (65-80%)[1,2] contain geographic information that is deemed to be at an insufficient level of granularity; information that is often present in the associated published article. This motivates the use of natural language processing (NLP) methods to find the geographic location (or toponym) of infected hosts in the full text. In NLP, this task of detecting toponyms from unstructured text, and then disambiguating the locations to their co-ordinates is formally known as toponym resolution.

Toponym resolution in scientific articles can be used to obtain precise geospatial metadata of infected hosts which is highly beneficial in building transmission models in phylogeography that could enable public health agencies to target high-risk areas. Improvement in geospatial metadata also enriches other scientific studies that utilize GenBank data, such as those in population genetics, environmental health, and epidemiology in general, as geographic location

is often used in addition to or as a proxy of other demographic data. Toponym Resolution is typically accomplished in two stages (1) toponym detection (geotagging), a named entity recognition (NER) task in NLP and (2) toponym disambiguation (geocoding).

For instance, given the sentence *"Our study mainly focused on pediatric cases with different outcomes from the most populated city in Argentina and one of the hospitals in Buenos Aires where patients are most often referred."*, the detection stage deals with extracting the locations *"Argentina"* and *"Buenos Aires"*.[3] The disambiguation stage deals with assigning the most likely, unique, identifiers from gazetteer resources like Geonames[a] to each location detected e.g. *"3865483:Argentina"* from 145 candidate entries containing the same name and *"3435910:Buenos Aires"* from 943 candidate entries with variations of the same name. Both tasks bring forth interesting NLP challenges with applications in a wide number of areas.

In this work, we present a system for toponym detection and disambiguation that improves substantially over previously published systems for this task, including our own.[4–6] Since detection is the first step in the process, its impact on the overall performance of the combined task is multiplied, as locations not detected can never be disambiguated. We use recurrent neural network (RNN) architectures that use word embeddings, character embeddings and case features as input for performing the detection task. In addition to these, we also experiment with the use of conditional random fields (CRF) on the output layer as they have known to improve performance. We perform ablation studies/leave-one-out analysis with repetitive runs with different seed values for drawing strong conclusions about the use of deep recurrent neural networks, their architectural variations and common features. We evaluate the impact of the results from the detection task on the upstream disambiguation task, performed using the commonly assumed *population heuristic*[7] whereby the location with the greatest population is chosen as the correct match.

The rest of the document is structured as follows. In Section 2, we summarize research efforts in the area of toponym detection and disambiguation and list the contributions of this paper in light of previous work. We distinguish the RNN architectures used for evaluation along with the population heuristic used for measurement in Section 3. Finally, we present and discuss the results of the toponym detection and disambiguation experiments in Sections 4 and discuss limitations and scope for improvements in Section 5.

2. Related Work

Toponym detection and toponym disambiguation have been widely researched by the NLP community, with numerous publications on both detection and disambiguation tasks.[8–10] Toponym detection is commonly tackled as a NER challenge where toponyms are recognized among other named entities like organization names and people's names. Previous studies[11] have identified the performance of the NER as an important source of errors in enhancing geospatial metadata in GenBank, motivating the development of tools for performing detection and resolution of named entities such as infected hosts and geographical locations.[12,13] The annotated dataset used in this work[4,11] includes both span and normalized Geonames ID

[a]http://www.geonames.org/ Accessed:Sept 30 2018

annotations. Since the performance of the overall resolution task is deeply influenced by the NER, some of the previous works using this dataset have looked specifically at improving the NER's performance. Our previous research on toponym detection have used rule-based methods,[4] traditional machine learning sequence taggers using conditional random fields (CRF)[5] and deep learning methods using feed forward neural networks.[6] NER performance since the introduction of the dataset has increased from an F1-score of 0.70 to 0.91 closing in on the human-level annotation agreement of 0.97. In the previous baseline for toponym resolution[4] a rule based extraction system was used to detect toponyms. In subsequent work, traditional machine learning algorithms such as conditional random fields (CRFs)[5] and feedforward neural nets[6] were introduced for improving the NER's performance. There exist some studies involving RNN experiments that explore the use of RNN architectures for sequence tagging tasks in the generic domain.[14,15] While these tasks measure the performance on specific tasks, the effect of optimal performances haven't been measured in upstream tasks.

On the other hand, toponym disambiguation has been commonly tackled as an information retrieval challenge by creating an inverted index of Geonames entries.[4,16] Given a toponym, candidate locations are first retrieved based on words used in the toponym and subsequently heuristics are used to pick the most appropriate location. Popular techniques use metrics such as entity co-occurrences, similarity measures, distance metrics, context features and topic modeling.[7,16–20] This approach is largely adopted due the large number of Geonames entries (about 12 million) to choose from. We also find that the most common baseline used for measuring the disambiguation performance is the population heuristic where the place with the most population is chosen as the correct match. Most research articles that focus specifically on the disambiguation problem use Stanford-NER or the Apache-NER tool[20–22] for detection which has been trained on datasets like CoNLL-2003, ACE-2005 and MUC. Some studies assume gold standard labels and proceed with the task of disambiguation which makes it difficult to assess the strength of the overall system. It is also important to note that a majority of efforts have been focused on texts from a general domain like Wikipedia or news articles.[20–22] Only a handful of publications deal with the problem in other domains like biomedical scientific articles[4,23] which contain a different and broader vocabulary. Similar to the previous disambiguation method developed for this dataset,[4] we build an inverted index using Geonames entries but use term expansion techniques to improve the performance and usability of the system in various contexts.

In light of previous work, the main contributions of this work can be summarized as follows:

(i) We perform a comprehensive and systematic evaluation of multiple RNN architectures from over 400 individual runs for the task of toponym detection in scientific articles and arrive at state-of-the-art results compared to previous methods.
(ii) We discuss the impact of significant performance improvement in toponym detection in the upstream task of toponym resolution.

3. Methods

Our approach for detection and disambiguation of geographic locations are tackled independently, as described in the following subsections. For the purposes of training and evaluation,

we use the publicly available human annotated corpus of 60 full-text PMC articles containing 1881 toponyms.[4] Of the 60, the standard test set for the corpus includes only 12 articles containing a total of 285 toponyms, a large majority of which are countries and major locations. The annotated dataset contains both span annotations and gazetteer ID annotations linking ISO-3166-1 codes for countries and GeonamesIDs for the remaining toponyms. For uniformity, we converted all ISO-3166-1 codes to equivalent GeonameIDs.

3.1. *Toponym Detection*

The task of toponym detection typically involves identifying the spans of the toponyms in an NER task where the sequence of actions is illustrated in Fig 1. As input features, we use publicly available pre-trained word embeddings that were trained on Wikipedia, PubMed abstracts and PubMed Central full text articles.[24] In addition to word embeddings, we experiment with orthogonal features such as (1) a case feature to explicitly distinguish all-uppercase, all-lowercase and camel-case words encoded as one-hot vectors that are appended to the word, and (2) fixed length character embeddings. Character embeddings have shown to improve the performances of deep neural networks and are employed in few different ways. One of the popular methods used involves the use of a CNN layer[25] or an LSTM layer[26] on vectors from a randomly initialized character embeddings that are fine tuned during training appended to the input word embedding layer. During initial experiments we found that implementation of this architecture added significantly to the training time and hence we employ the use of a simpler model where character embeddings are pre-trained using word2vec and appended directly to the input layer along with word embeddings and case features.

The proposed RNN units and their variations can be used on their own for NER purposes. However, bidirectional architectures are popularly employed for NER as they have the combined capability of processing input sentences in both directions and making tagging decisions collectively using an output layer as illustrated in figure 1. In this paper, we specifically look at bi-directional recurrent architectures. It is also common to observe the use of a CRF output layer on top of the output layer of bidirectional RNN architecture. CRF's are known to add consistency in making final tagging decisions using IOB or IOBES styled annotations. We experiment between combinations of the RNN variants along with the optional features in an ablation study to identify the impact of these additive layers on the NER's performance as well as its impact on the upstream resolution task.

3.1.1. *Recurrent Neural Networks*

RNN architectures have been widely used for auto-encoders and sequence labeling tasks such as part-of-speech tagging, NER, chunking among others.[27] RNNs are variants of feedforward neural networks that are equipped with recurrent units to carry signals from the previous output y^{t-1} for making decisions at time y^t as shown in equation 1.

$$y_t = \sigma\left(W \cdot x_t + U \cdot y_{t-1} + b\right) \tag{1}$$

Here, W and U are the weight matrices and b is the bias term that are randomly initialized and updated during training. σ represents the sigmoid activation function. In practice

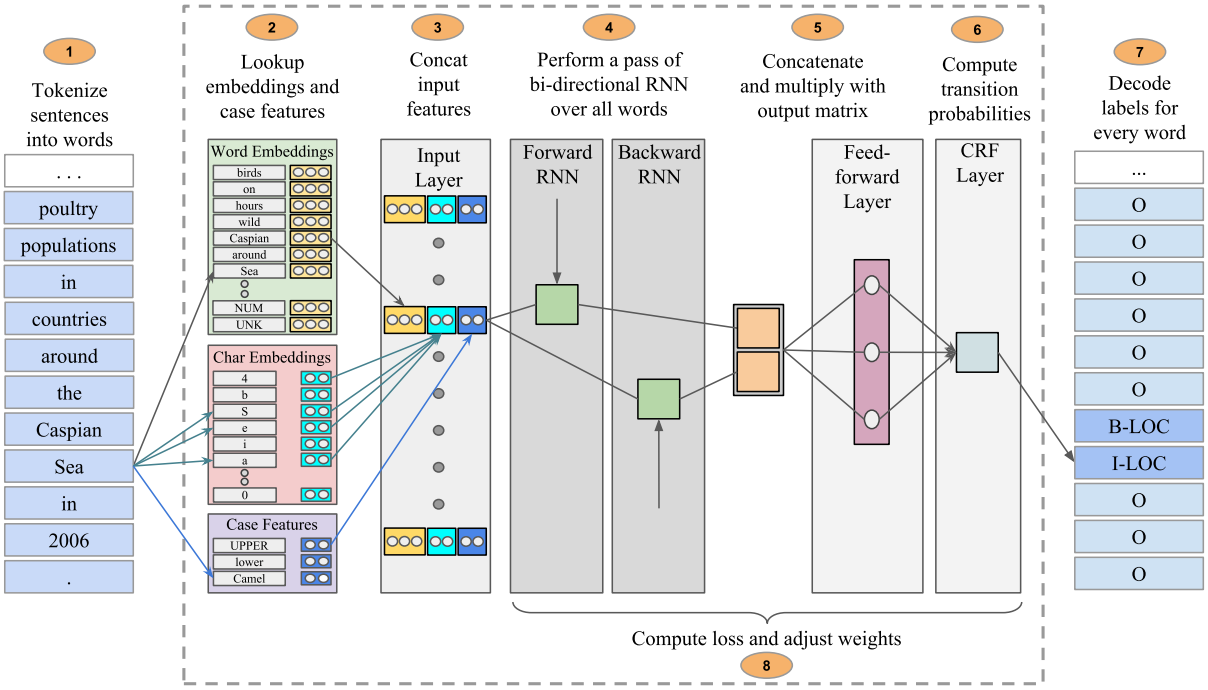

Fig. 1. A schematic representation of the sequence of actions performed in the NER equipped with bi-directional RNN layers and an output CRF layer. RNN variants discussed in this paper involve replacing RNN units with LSTM, LSTM-Peepholes, GRU and UG-RNN units.

other activation functions such as *tanh* and rectified linear units (*ReLU*) are also used. This characteristic recurrent feature simulates a memory function that makes it ideal for tasks involving sequential predictions dependent on previous decisions. However, learning long term dependencies that are necessary have been found to be difficult using RNN units alone.[28]

3.1.2. *LSTM*

LSTM networks[29] are variants of RNN that have proven to be fairly successful at learning long term dependencies. A candidate output g is calculated using an equation similar to equation 1 and further manipulated based on previous and current states of a cell that retains signals simulating long-term memory. The LSTM cell's state is controlled by *forget (f)*, *input (i)* and *output (o)* gates that control how much information flows from the input to the state and from state to the output. The gates themselves depend of current input and previous outputs.

$$g = tanh(W^g \cdot x_t + U^g \cdot y_{t-1} + b^g) \tag{2}$$

$$f = \sigma(W^f \cdot x_t + U^f \cdot y_{t-1} + b^f) \tag{3}$$

$$i = \sigma(W^i \cdot x_t + U^i \cdot y_{t-1} + b^i) \tag{4}$$

$$o = \sigma(W^o \cdot x_t + U^o \cdot y_{t-1} + b^o) \tag{5}$$

The future state of the cell c_t is calculated as a combination of (1) signals from forget gate g and the previous state of the cell c_{t-1} which determines the information to forget (or retain)

in the cell, and (2) signals from the *input* gate i and the candidate output g that determines the information from the input to be stored in the cell. Eventually the output y_t is calculated using signals from the output gate o and the current state of the cell c_t.

$$c_t = f \odot c_{t-1} + i \odot g \tag{6}$$

$$y_t = o \odot tanh(c_t) \tag{7}$$

In the above equations, $\odot$ indicates pointwise multiplication operation. While the above equations represent LSTM in its most basic form, many variations of the architecture have been introduced to simulate retention of long-term signals a few of which have been summarized in the following subsections and subsequently evaluated in the results section. For reasons of brevity, we do not include the formulas used for calculating the output y_t but they can be inferred from the works cited.

3.1.3. *Other Gated RNN Architectures*

We evaluate in our experiments one of the LSTM variations introduced for speech processing[30] that introduced the notion of peepholes (LSTM-Peep) where the idea is that state of the cell influences the *input*, *forget* and *output* gates. Here, signals for the *input* and *forget* gates i and f depend not only on the previous output y_{t-1} and current input x_t but also the previous state of the cell c_{t-1} and the *output* gate o depends on the current state of the cell c_t.

Gated Recurrent Unit (GRU)[31] also known as coupled input and forget gate LSTM (CIFG-LSTM)[15] is a simpler variation of LSTM with only two gates: update z and reset r. Their signals are determined based on the current input x and previous output y_{t-1} similar to the gates in LSTMs. The update gate z attempts to combine the functionality of input and forget gates of LSTMs i and f and eliminates the need for an output gate as well as an explicit cell state. A singular update gate signal z controls the information flow to the output value. Although it appears far more simple, GRU has gained a lot of popularity in the recent years in a variety of NLP tasks.[32,33]

Update gate RNN (UG-RNN)[34] is a much simpler variation of LSTM and GRU architectures containing only an update gate z is also included in our experiments. The importance of the update gate is often highlighted in RNN based architectures.[15] Hence, we include this model to perform a gate based ablation study to understand their contributions to the overall resolution task.

3.1.4. *Hyperparameter search and optimization*

The performance of deep neural networks relies greatly on optimization of its hyperparameters and the performance of the models have been found to be sensitive to changes in seed values used for initializing the weight matrices.[27] We first performed a grid search over the previously recommended optimal range of hyperparameter space for NER tasks[27] and to arrive at potential candidates of optimal configurations. We then performed up to 5 repetitions of experiments at the optimal setting for the model at different seed values to obtain the median performance scores. All models were developed using the TensorFlow framework and trained on NVIDIA Titan Xp GPUs equipped with an Intel Xeon CPU (E5-2687W v4).

3.2. *Toponym Disambiguation*

For toponym disambiguation, we use the Geonames gazetteer data to build an inverted index using Apache Lucene[b] and search for the toponym terms extracted in the toponym detection step in the index.

3.2.1. *Building Geonames Index*

Individual Geonames entries in the index are documents with common fields such as *GeonameID*, *LocationName*, *Latitude*, *Longitude*, *LocationClass*, *LocationCode*, *Population*, *Continent* and *AncestorNames*. Here, *LocationName* contains the common name of the place. For countries, we expand this field by using official names, ISO and ISO3 abbreviations (e.g. *United States of America*, *US* and *USA*, respectively, for *United States*). For ADM1 (Administrative Level 1) entries that have available abbreviations (e.g. *AZ* for *Arizona*, and *CA* for *California*), we add such alternate names to the *LocationName* field. In addition to the above fields we add the *County*, *State* and *Country* fields depending on the type of geoname entry. Fields such as *LocationName*, *County*, *State*, *Country* and *AncestorNames* are chosen to be reverse indexed such that partial matches of names offers the possibility of being matched with the right disambiguated toponym on a search.

3.2.2. *Searching Geonames Index*

Most cities and locations commonly have their parent locations listed as comma separated values (e.g. *Philadelphia, PA, USA*). In such cases, the index provides the capability to perform compound searches (e.g. *LocationName:"Philadelphia" AND AncestorNames:"PA, USA"*). We find that this method offers the best scalable framework for toponym disambiguation among approximately 12 million entries. Efficient search capabilities aside, the solution internally provides documents to be sorted by a particular field. In this case, we choose the *Population* field as the default sorting heuristic such that search results are sorted by highest population first. An additional motivation for the implementation of this solution is the flexibility of using external information to narrow down search results. For example, when Country information is available in the GenBank record, we can use queries like *LocationName:"Paris" AND Country:"France"* to narrow down the location of infected hosts.

4. Results and Discussion

For the NER task, we use the standard metric scores of precision, recall, and F_1-scores for toponym entities across two modes of evaluation:(1) *Strict* where the predicted spans of the toponym have to match exactly with the gold standard spans to be counted as a true positive and (2) *Overlapping* where predicted spans are true positives as long as one of its tokens overlap with gold standard annotations. For toponym disambiguation, we compare the predicted and gold standard GeonameIDs to measure precision, recall and f_1-scores as long as the spans overlap. We compare our scores with the previous systems that were trained and tested on the

[b]http://lucene.apache.org/ Accessed:Sept 30 2018

same dataset. To evaluate the performance of the overall resolution task, it is important to examine the performance of the individual systems to assess the cause of errors and identifying regions for improvement.

4.1. *Toponym Disambiguation*

Our toponym disambiguation system is unsupervised, giving us the capability to test its performance on the entire dataset assuming gold standard toponym terms to be available. Under this assumption, the accuracy of the disambiguation system was found to be 91.6% and 90.5% on training and test set respectively. Analyzing the errors, we found that comparing ids directly is a very strict mode of evaluation for the purposes of phylogeography as Geonames contains duplicate entries for many locations that belong to two or more classes of locations such as administrative division (ADM) and populated area or city (PPLA, PPLC) but refer to the same geographical location. For instance, when we look at the test set alone, which had 27 errors from a total of 285 locations, 19 appeared to be roughly the same location. These included locations like *Auckland, Lagos, St. Louis, Cleveland, Shantou, Nanchang, Shanghai, and Beijing* which were assigned the ID of the administrative unit by the system, while the annotated locations were assigned the ID of the populated area or city or *vice versa*. Given these reasons, we find that the performance of the resolution step exceeds the reported scores by 5% to arrive at an approximate accuracy of 95-96%. However, for the purposes of comparison with previous systems we report the overall resolution performance in Table 1 without making such approximations. We did however observe 8 errors where the system assigned GeonamesIDs were drastically different from their original locations due to the population heuristic. For example, a toponym of *Madison* was incorrectly assigned the ID of *Madison County, Alabama* which had a higher population than the gold standard annotation *Madison, Dane County, Wisconsin(WI)*.

4.2. *Toponym Resolution*

Analyzing the errors across the architectures, we find that 80-90% of the erroneous instances to be repeating across the RNN architectures making it challenging to use ensemble methods for reducing errors. These included false negative toponyms such as *Plateau, Borno, Ga, Gurjev, Sokoto* etc. which appear in tables and structured contexts making it difficult to recognize them. However, as discussed in our previous work,[6] we plan to handle table structures differently by employing alternative methods of conversions from pdf to text. Almost all false positives appeared to be geographic locations, however in the text they were found to be referring to other named entities like virus strains and isolates rather than toponyms.

We found that the LSTM-Peep based architecture appeared to have marginally better performance scores on the NER task and hence the overall resolution task. Feature ablation analysis shown in Figure 2 indicate that inclusion of the character embedding feature contributed to increase in the overall performance of RNN models. However, inclusion of case feature in combination with the character embeddings appeared to be redundant. Inclusion of the CRF output layer seemed to have a positive impact on most models while additive layers seemed to have more effect on GRU, LSTM and LSTM-Peep architectures.

Table 1. Median Precision(P), Recall(R) and F_1 scores for NER and Resolution. Bold-styled scores indicate highest performance. All recurrent neural network units were used in a bidirectional setup with inputs containing pre-trained word embeddings, character embeddings and case features, and an output layer with an additional CRF layer.

Method	NER-Strict			NER-Overlapping			Resolution		
	P	R	F_1	P	R	F_1	P	R	F_1
Rule-based[4]	0.58	0.876	0.698	0.599	0.904	0.72	0.547	**0.897**	0.697
CRF-All[5]	0.85	0.76	0.80	0.86	0.77	0.81	-	-	-
FFNN + DS[6]	0.90	0.93	0.91	-	-	-	-	-	-
RNN	0.910	0.891	0.901	0.931	0.912	0.922	0.896	0.817	0.855
UG-RNN	0.948	0.902	0.924	0.959	0.912	0.935	0.903	0.824	0.862
GRU	**0.952**	0.919	0.935	**0.967**	0.930	0.948	0.888	0.835	0.860
LSTM	0.932	0.926	0.929	0.954	0.947	0.950	0.892	0.842	0.866
LSTM-Peep	0.934	**0.944**	**0.939**	0.951	**0.961**	**0.956**	**0.907**	0.863	**0.884**

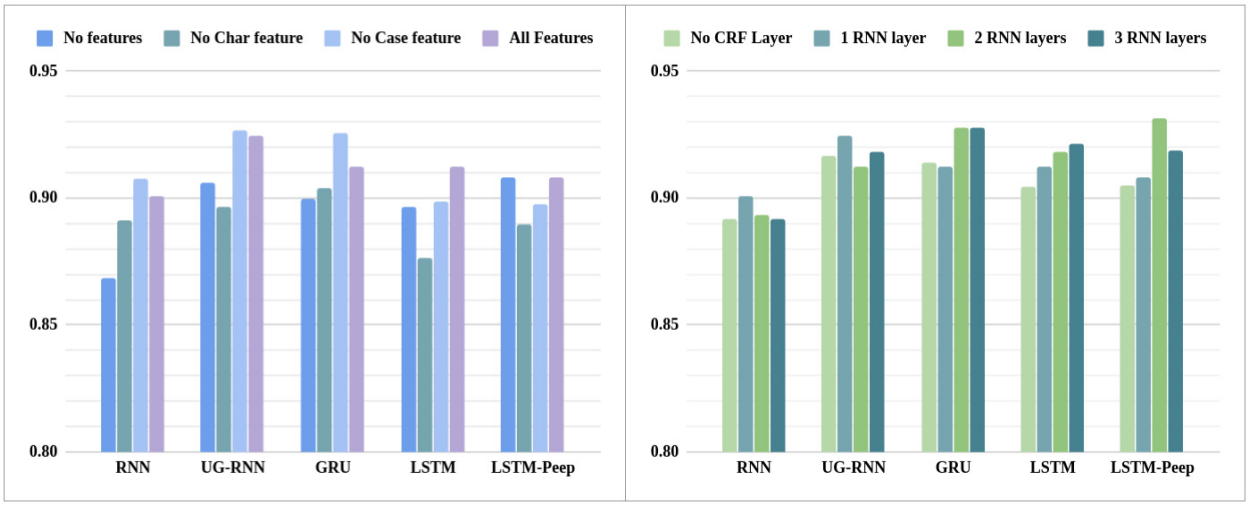

Fig. 2. *(Left)* Ablation/leave-one-out analysis showing the contribution of individual features to the NER performance across the RNN models. *(Right)* Impact of additive layers on the performance of the NER across the RNN models. Here, RNN layers refer to respective variants of RNN architectures. Y-axis shows strict F_1 scores.

5. Limitations and Future Work

In this work, we find that utilizing state-of-the-art NER architectures help us obtain performances that are inching close to human performance. However, we do find that the articles in the test set may perhaps be relatively easier than the average article for the detection task when we compare it to randomly selected validation/development set performances. As discussed in our previous work,[6] distance supervision datasets can contain toponym spans in close proximity to each other generating noisy training examples. This makes it challenging to

use distance supervision techniques to increase the size of training data for training sequence tagging models based on RNN architectures. Hence, to address this issue, we are in the process of expanding the annotation dataset from 60 articles to 150 articles for a more comprehensive training and evaluation of the system.

Irrespective of the ease of detection in the test set, there appear to be false negative toponyms (discussed in the previous section) that could possibly be the location of infected hosts(LOIH). While there are chances that toponyms that are LOIH appear repeatedly in the scientific article in varying contexts thus increasing the chances of them being detected, in our following work we wish to evaluate the impact of these false negatives on the overall task of identifying the LOIH. To reduce false positives where locations could infact refer to other named entities like virus strains and isolates than toponyms themselves, we intend to explore approaches from metonymy resolution[35] for filtering out such false positives.

6. Conclusion

Phylogeography research relies on accurate geographical metadata information from nucleotide repositories like GenBank. In records that contain insufficient metadata information, there is a motivation to extract the geographical location from the associated articles to determine the location of the infected hosts. In this work we present and evaluate methods built on recurrent neural networks that extract geographical locations from scientific articles with a substantial increase in performance from an F_1 score of 0.88 which improves significantly over the previous toponym resolution system F_1 of 0.69. Our implementations of the toponym detection and toponym disambiguation[c] systems along with the updated version of the annotations containing GeonameIDs[d] are available online.

Acknowledgments

AM designed and trained the neural networks, ran the experiments, performed the error analysis, and wrote most of the manuscript. DW and AS reviewed, restructured and contributed many sections and revisions of the manuscript. MS and GG provided overall guidance on the work and edited the final manuscript. The authors would also like to acknowledge Karen OConnor, Megan Rorison and Briana Trevino for their efforts in the annotation processes. The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The authors are also grateful to ASU-BMI's computing resources used for conducting the experiments in the paper.

Funding

Research reported in this publication was supported by the National Institute of Allergy and Infectious Diseases (NIAID) of the National Institutes of Health (NIH) under grant number R01AI117011 to MS and GG. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

[c] `https://bitbucket.org/pennhlp/toponym-resolution-using-rnns Accessed:30 Sept 2018`
[d] `https://healthlanguageprocessing.org/software-and-downloads/ Accessed:30 Sept 2018`

References

1. M. Scotch, I. N. Sarkar, C. Mei, R. Leaman, K.-H. Cheung, P. Ortiz, A. Singraur and G. Gonzalez, Enhancing phylogeography by improving geographical information from genbank *Journal of biomedical informatics* **44** (Elsevier, 2011).
2. T. Tahsin, R. Beard, R. Rivera, R. Lauder, G. Wallstrom, M. Scotch and G. Gonzalez, Natural language processing methods for enhancing geographic metadata for phylogeography of zoonotic viruses *AMIA Summits on Translational Science Proceedings* **2014** (American Medical Informatics Association, 2014).
3. P. Barrero, M. Viegas, L. Valinotto and A. Mistchenko, Genetic and phylogenetic analyses of influenza a h1n1pdm virus in buenos aires, argentina *Journal of virology* **85** (Am Soc Microbiol, 2011).
4. D. Weissenbacher, T. Tahsin, R. Beard, M. Figaro, R. Rivera, M. Scotch and G. Gonzalez, Knowledge-driven geospatial location resolution for phylogeographic models of virus migration *Bioinformatics* **31** (Oxford University Press, 2015).
5. D. Weissenbacher, A. Sarker, T. Tahsin, M. Scotch and G. Gonzalez, Extracting geographic locations from the literature for virus phylogeography using supervised and distant supervision methods *AMIA Summits on Translational Science Proceedings* **2017** (American Medical Informatics Association, 2017).
6. A. Magge, D. Weissenbacher, A. Sarker, M. Scotch and G. Gonzalez-Hernandez, Deep neural networks and distant supervision for geographic location mention extraction *Bioinformatics* **34**2018.
7. J. L. Leidner, Toponym resolution in text: annotation, evaluation and applications of spatial grounding, in *ACM SIGIR Forum*, (2)2007.
8. M. Gritta, M. T. Pilehvar, N. Limsopatham and N. Collier, Whats missing in geographical parsing? *Language Resources and Evaluation* **52** (Springer, 2018).
9. J. L. Leidner and M. D. Lieberman, Detecting geographical references in the form of place names and associated spatial natural language *SIGSPATIAL Special* **3** (ACM, 2011).
10. R. Tobin, C. Grover, K. Byrne, J. Reid and J. Walsh, Evaluation of georeferencing, in *proceedings of the 6th workshop on geographic information retrieval*, 2010.
11. T. Tahsin, D. Weissenbacher, R. Rivera, R. Beard, M. Firago, G. Wallstrom, M. Scotch and G. Gonzalez, A high-precision rule-based extraction system for expanding geospatial metadata in genbank records *Journal of the American Medical Informatics Association* **23** (Oxford University Press, 2016).
12. T. Tahsin, D. Weissenbacher, D. Jones-Shargani, D. Magee, M. Vaiente, G. Gonzalez and M. Scotch, Named entity linking of geospatial and host metadata in genbank for advancing biomedical research *Database* **2017** (Oxford University Press, 2017).
13. T. Tahsin, D. Weissenbacher, K. Oconnor, A. Magge, M. Scotch and G. Gonzalez-Hernandez, Geoboost: accelerating research involving the geospatial metadata of virus genbank records *Bioinformatics* **34** (Oxford University Press, 2017).
14. R. Jozefowicz, W. Zaremba and I. Sutskever, An empirical exploration of recurrent network architectures, in *International Conference on Machine Learning*, 2015.
15. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, Lstm: A search space odyssey *IEEE transactions on neural networks and learning systems* **28** (IEEE, 2017).
16. S. Overell and S. Rüger, Using co-occurrence models for placename disambiguation *International Journal of Geographical Information Science* **22** (Taylor & Francis, 2008).
17. A. Spitz, J. Geiß and M. Gertz, So far away and yet so close: augmenting toponym disambiguation and similarity with text-based networks, in *Proceedings of the third international ACM SIGMOD workshop on managing and mining enriched geo-spatial data*, 2016.

18. Y. Ju, B. Adams, K. Janowicz, Y. Hu, B. Yan and G. McKenzie, Things and strings: improving place name disambiguation from short texts by combining entity co-occurrence with topic modeling, in *European Knowledge Acquisition Workshop*, 2016.
19. M. D. Lieberman and H. Samet, Adaptive context features for toponym resolution in streaming news, in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012.
20. E. Kamalloo and D. Rafiei, A coherent unsupervised model for toponym resolution, in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 2018.
21. M. D. Lieberman and H. Samet, Multifaceted toponym recognition for streaming news, in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.
22. J. Hoffart, Discovering and disambiguating named entities in text, in *Proceedings of the 2013 SIGMOD/PODS Ph. D. symposium*, 2013.
23. J. Tamames and V. de Lorenzo, Envmine: A text-mining system for the automatic extraction of contextual information *BMC bioinformatics* **11** (BioMed Central, 2010).
24. S. Pyysalo, F. Ginter, H. Moen, T. Salakoski and S. Ananiadou, Distributional semantics resources for biomedical text processing (2013).
25. X. Ma and E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.
26. G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, Neural architectures for named entity recognition, in *Proceedings of NAACL-HLT*, 2016.
27. N. Reimers and I. Gurevych, Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Copenhagen, Denmark, 2017).
28. Y. Bengio, P. Simard and P. Frasconi, Learning long-term dependencies with gradient descent is difficult *IEEE transactions on neural networks* **5**1994.
29. S. Hochreiter and J. Schmidhuber, Long short-term memory *Neural computation* **9** (MIT Press, 1997).
30. H. Sak, A. Senior and F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in *Fifteenth annual conference of the international speech communication association*, 2014.
31. K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, Learning phrase representations using rnn encoder–decoder for statistical machine translation, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
32. Z. Che, S. Purushotham, K. Cho, D. Sontag and Y. Liu, Recurrent neural networks for multivariate time series with missing values *Scientific reports* **8** (Nature Publishing Group, 2018).
33. Y. Luo, Recurrent neural networks for classifying relations in clinical notes *Journal of biomedical informatics* **72** (Elsevier, 2017).
34. J. Collins, J. Sohl-Dickstein and D. Sussillo, Capacity and trainability in recurrent neural networks, in *Profeedings of the International Conference on Learning Representations (ICLR)*, 2017.
35. M. Gritta, M. T. Pilehvar, N. Limsopatham and N. Collier, Vancouver welcomes you! minimalist location metonymy resolution, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.