Toponym Identification in Epidemiology Articles – A Deep Learning Approach

MohammadReza Davari, Leila Kosseim, and Tien D. Bui

Dept. Computer Science and Software Engineering
Concordia University, Montreal QC H3G 1M8, Canada
mohammadreza.davari@mail.concordia.ca
{leila.kosseim,tien.bui}@concordia.ca

Abstract. When analyzing the spread of viruses, epidemiologists often need to identify the location of infected hosts. This information can be found in public databases, such as GenBank [3], however, information provided in these databases are usually limited to the country or state level. More fine-grained localization information requires phylogeographers to manually read relevant scientific articles. In this work we propose an approach to automate the process of place name identification from medical (epidemiology) articles. The focus of this paper is to propose a deep learning based model for toponym detection and experiment with the use of external linguistic features and domain specific information. The model was evaluated using a collection of 105 epidemiology articles from PubMed Central [33] provided by the recent SemEval task 12 [28]. Our best detection model achieves an F1 score of 80.13%, a significant improvement compared to the state of the art of 69.84%. These results underline the importance of domain specific embedding as well as specific linguistic features in toponym detection in medical journals.

Keywords: Named entity Recognition · Toponym Identification · Deep Neural Network · Epidemiology Articles.

1 Introduction

With the increase of global tourism and international trade of goods, phylogeographers, who study the geographic distribution of viruses, have observed an increase in the geographical spread of viruses [9,12]. In order to study and model the global impact of the spread of viruses, epidemiologists typically use information on the DNA sequence and structure of viruses, but also rely on meta data. Accurate geographical data is essential in this process. However, most publicly available data sets, such as GenBank [3], lack specific geographical details, providing information only at the country or state level. Hence, localized geographical information has to be extracted through a manual inspection of medical journals.

The task of toponym resolution is a sub-problem of named entity recognition (NER), a well studied topic in Natural Language Processing (NLP). Toponym

resolution consists of two sub-tasks: toponym identification and toponym disambiguation. Toponym identification consists of identifying the word boundaries of expressions that denote geographic expressions; while toponym disambiguation focuses on labeling the expression with their corresponding geographical locations. Toponym resolution has been the focus of much work in recent years (e.g. [2,7,30]) and studies have shown that the task is highly dependent on the textual domain [1,25,26,14,8]. The focus of this paper is to propose a deep learning based model for toponym detection and experiment with the use of external linguistic features and domain specific information. The model was evaluated using the recent SemEval task 12 datatset [28] and shows that domain specific embedding as well as some linguistic features do help in toponym detection in medical journals.

2 Previous Work

The task of toponym detection consists in labeling each word of a text as a toponym or non-toponym. For example, given the sentence:

(1) WNV entered Mexico through at least 2 independent introductions[1].

The expected output is shown in Figure 1.

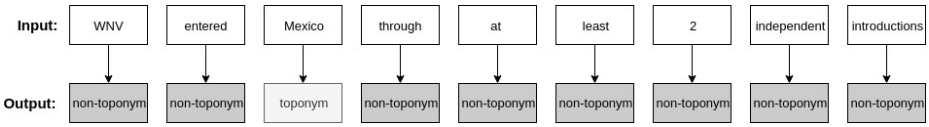

Fig. 1. An example of input and expected output of toponym detection task. Example from the [33] dataset.

Toponym detection has been addressed using a variety of methods: rule based approaches (e.g. [29]), dictionary or gazetteer-driven (e.g. [18]), as well as machine learning approaches (e.g. [27]). Rule based techniques try to manually capture textual clues or structures which could indicate the presence of a toponym. However, these handwritten rules are often not able to cover all possible cases, hence leading to a relatively large number of false negatives. Gazetteer driven approaches (e.g. [18]), suffer from a large number of false positive identifications, since they cannot disambiguate entities that refer to geographical locations from other categories of named entities. For example in the sentence,

(2) Washington was unanimously elected President by the Electoral College in the first two national elections.

[1] Example from the [33] dataset.

the word *Washington* will be recognized as a toponym since it is present in geographic gazetteers but in this context, the expression refers to a person name. Finally, standard machine learning approaches (e.g. [27]), require large datasets of labeled texts and carefully engineered features. Collecting such large datasets is costly and feature engineering is a time consuming task, with no guarantee that all relevant features have been modeled. This motivated us to experiment with automatic feature learning to address the problem of toponym detection. Deep Learning approaches to NER (e.g. [5,6,16,17,31]) have shown how a system can infer relevant features and lead to competitive performances in that domain.

The task of toponym resolution for the epidemiology domain is currently the object of the SemEval 2019 shared task 12 [28]. Previous approaches to toponym detection in this domain includes rule based approach [33], Conditional Random Fields [32], and a mixture of deep learning and rule based approaches [19]. The baseline model used at the SemEval 2019 task 12 [28] is modeled after the deep feed forward neural network (DFFNN) architecture presented in [19]. The network consists of 2 hidden layers with 150 rectified linear unit (ReLU) activation functions per layer. The baseline F1 performance is reported to be 69.84%. Building upon the work of [28,19] we propose a DFFNN that uses domain-specific information as well as linguistic features to enhance the state of the art performance.

3 Our Proposed Model

Figure 2 shows the architecture of our toponym recognition model. The model is comprised of 2 main layers: an embedding layer, and a deep feed-forward network.

3.1 Embedding Layer

As shown in Figure 2, the model takes as input a word (e.g. *derived*) and its context (i.e. n words around it). Given a document, each word is converted into an embedding along with its context. Specifically, two types of embeddings are used: word embeddings and feature embeddings.

For word embeddings, our basic model uses the pretrained Wikipedia-PubMed embeddings[2]. This embedding model was trained on a vocabulary of $201,380$ words and each word is represented by a 200 dimensional feature vector. This embedding model was used as opposed to more generic Word2vec [20] or GloVe [23] in order to capture more domain specific information (see Section 4). Indeed, the corpus used for training the Wikipedia-PubMed embedding consists of Wikipedia pages and PubMed articles [21]. This entails that the embeddings should be more appropriate when processing medical journals, and domain specific words. Moreover, the embedding model can better represent the closeness and relation of words in medical articles. The word embeddings for the target word and

[2] `http://bio.nlplab.org/`

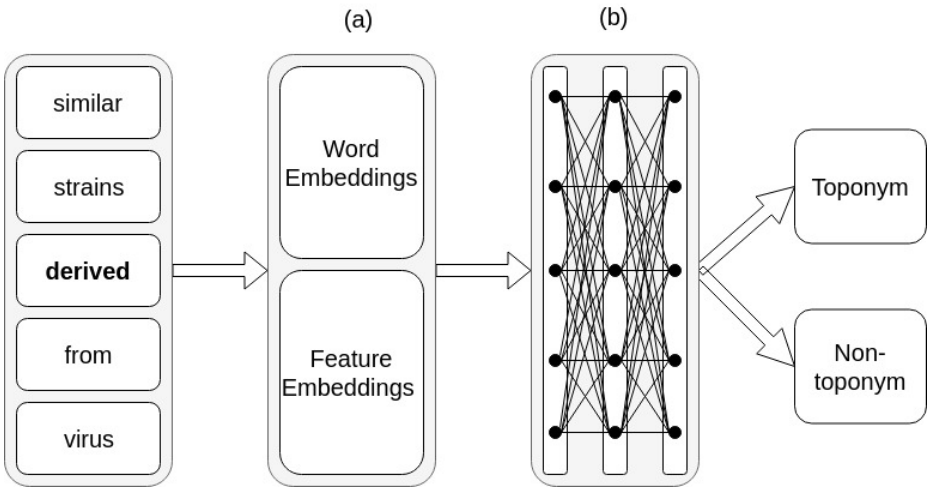

Fig. 2. Toponym recognition model. Input: words are extracted with a fixed context window (a) Embeddings: For each window, an embedding is constructed (b) Deep Neural Network: A feed-forward neural network with 3 layers and 500 neurons per layer outputs a prediction label indicating whether the word in the center of the context window is a toponym or not.

its context are concatenated to form a single word embedding vector of size $200 \times (2c + 1)$, where c is the context size.

Specific linguistic features have been shown to be very useful in toponym detection [19]. In order to leverage this information, our model is augmented using embedding for these features. These include the use of capital letters for the first character of the word or for all characters of the word. These features are encoded as a binary vector representation. If a word starts with a capitalized letter, its feature embedding is $[1, 0]$ otherwise it is $[0, 1]$ and if all of its letters are capitalized then its feature embedding is $[1, 1]$. Other linguistic features we observed to be useful (see Section 4) include part of speech tags, and the word embedding of the lemma of the word. The feature embedding of the input word and its context are combined to the word embedding via concatenation to form a single vector and passed to the next layer.

3.2 Deep Feed Forward Neural Network

The concatenated embeddings formed in the embedding layer (Section 3.1) are fed to a deep feed forward network (DFFNN) (see Figure 2) whose task is to perform binary classification. This component is comprised of 3 hidden layers and one output layer. Each hidden layer is comprised of 500 ReLU activation nodes. Once an input vector x enters a hidden layer h, the output $h(x)$ is computed as:

$$h(x) = \text{ReLU}(Wx + b) \tag{1}$$

The model is defined using the above equation recursively for all 3 hidden layers. The output layer contains 2 dimensional softmax activation functions. Upon receiving the input x, this layer will output $O(x)$ as follows:

$$O(x) = \text{Softmax}(Wx + b) \tag{2}$$

The Softmax function was chosen for the output layer since it provides a categorical probability distribution over the labels for an input x, i.e.:

$$p(x = \text{toponym}) = 1 - p(x = \text{non-toponym}) \tag{3}$$

We employed 2 mechanisms to prevent overfitting: drop-out and early-stopping. In each hidden layer the probability of drop-out was set to 0.5. The early-stopping caused the training to stop if the loss on the development set (see Section 4) started to rise preventing over-fitting and poor generalization. Norm clipping [22] scales the gradient when its norm exceeds a certain threshold and prevents the occurrence of exploding gradient; we experimentally found the best performing threshold to be 1 for our model.

We experimented with variations of the model architecture both in depth and number of hidden units per layer as well as other hyper-parameters listed in Table 1. However, deepening the model lead to immediate over-fitting due to the small size of the dataset used [13] (see Section 4) even with the presence of a dropout function to prevent it. The optimal hyper-parameter configuration with the development set used to fine tune them can be found in Table 1.

Table 1. Optimal hyper-parameters of the neural network.

Parameters	Value
Learning Rate	0.01
Batch Size	32
Optimizer	SGD
Momentum	0.1
Loss	Weighted Categorical cross-entropy
Loss weights	$(2, 1)$ for toponym vs. nontoponym

4 Experiments and Results

Our model has been evaluated as part of the recent SemEval 2019 task 12 shared task [28]. As such, we used the dataset and the scorer[3] provided by the organisers. The dataset consists of 105 articles from PubMed annotated with toponym mentions and their corresponding geographical locations. The dataset was split into 3 sections: training, development, and test set containing 60%, 10%, and 30% of the dataset respectively. Table 2 shows statistics of the dataset.

[3] `https://competitions.codalab.org/competitions/19948#learn_the_details-evaluation`

Table 2. Statistics of the dataset.

	Training	Development	Test
Size	2.8MB	0.5MB	1.5MB
Number of articles	63	10	32
Average size of each article (in words)	6422	5191	6146
Average number of toponyms per article	43	44	50

A baseline model for toponym detection was also provided by the organizers for comparative purposes. The baseline, inspired by [19], also uses a DFFNN but only uses 2 hidden layers and 150 ReLU activation functions per layer.

Table 3 shows the results of our basic model presented in Section 3.2 (see #4) compared to the baseline (row #3).[4]We carried out a series of experiments to evaluate a variety of parameters. These are described in the next sections.

Table 3. Performance score of the baseline, our proposed model and its variations. The suffixes represent the presence of a feature, P.:Punctuation marks, S: Stop words, C: Capitalization features, POS: Part of speech tags, W: Weighted loss, L: Lemmatization feature. For example DFFNN Basic+P+S+C+POS refers to the model that only takes advantage of capitalization feature and part of speech tags and does not ignore stop words or punctuation marks.

#	Model	Context	Precision	Recall	F1
8	DFFNN Basic+P+S+C+POS+W+L	5	80.69%	79.57%	80.13%
7	DFFNN Basic+P+S+C+POS+W	5	76.84%	77.36%	77.10%
6	DFFNN Basic+P+S+C+POS	5	77.55%	70.37%	73.79%
5	DFFNN Basic+P+S+C	2	78.82%	66.69%	72.24%
4	DFFNN Basic+P+S	2	79.01%	63.25%	70.26%
3	Baseline	2	73.86%	66.24%	69.84%
2	DFFNN Basic+P−S	2	74.70%	63.57%	68.67%
1	DFFNN Basic+S−P	2	64.58%	64.47%	64.53%

4.1 Effect of Domain Specific Embeddings

As [1,25,26,14,8] showed, the task of toponym detection is dependent on the discourse domain; this is why our basic model used the Wikipedia-PubMed embeddings. In order to measure the effect of such domain specific information, we experimented with 2 other pretrained word embedding models: Google News Word2vec [11], and a GloVe Model trained on Common Crawl [24]. Table 4 shows the characteristics of these pretrained embeddings. Although, the Wikipedia-PubMed has a smaller vocabulary in comparison to the other embedding models, it suffers from the smallest percentage of out of vocabulary (OOV) words within our dataset since it was trained on a closer domain.

[4] At the time of writing this paper, the results of the other teams were not available. Hence only a comparison with the baseline can be made at this point.

Table 4. Specifications of the word embedding models.

Model	Vocabulary Size	Embedding Dimension	OOV words
Wikipedia-PubMed	201,380	200	28.61%
Common Crawl GloVe	2.2M	300	29.84%
Google News Word2vec	3M	300	44.36%

We experimented with our DFFNN model with each of these embeddings and optimized the context window size to achieve the highest F-measure on the development set. The performance of these models on the test set is shown in Table 5. As predicted, we observe that Wikipedia-PubMed performs better than the other embedding models. This is likely due to its small number of OOV words and its domain-specific knowledge. As Table 5 shows, the performance of the GloVe model is quite close to the performance of Wikipedia-PubMed. To investigate this further, we decided to combine the two embeddings and train another model and evaluate performance. As shown in Table 5, the performance of this model (Wikipedia-PubMed + GloVe) is higher than the GloVe model alone but lower than the Wikipedia-PubMed. This decrease in performance suggests that because GloVe embeddings are more general, when the network is presented with a combination of GloVe and Wikipedia-PubMed, they dilute the domain specific information captured by the Wikipedia-PubMed embeddings, hence the performance suffers. From here on, our experiments were carried on using Wikipedia-PubMed word embeddings alone.

Table 5. Effect of word embeddings on the performance of our proposed model architecture.

Model	Context Window	Precision	Recall	F1
Wikipedia-PubMed	2	79.01%	63.25%	70.26%
Wikipedia-PubMed + GloVe	2	73.09%	67.22%	70.03%
Common Crawl GloVe	1	75.40%	64.05%	69.25%
Google News Word2vec	3	75.14%	58.96%	66.07%

4.2 Effect of Linguistic Features

Although deep learning approaches have lead to significant improvements in many NLP tasks, simple linguistic features are often very useful. In the case of NER, punctuation marks constitute strong signals. To evaluate this in our task, we ran the DFFNN Basic without punctuation information. As Table 3 shows, the removal of punctuation, decreased the F-measure from 70.26% to 64.53% (see Table 3 #1). A manual error analysis showed that many toponyms appear inside parenthesis, near a dot at the end of a sentence, or after a comma. Hence, as shown in [10] punctuation is a good indicator of toponyms and should not be ignored.

As Table 3 (#2) shows, the removal of stop words, did not help the model either and lead to a decrease in F-measure (from 70.26% to 68.67%). We hypothesize that some stop words such as *in* do help the system detect toponyms as they provide a learnable structure for detection of toponyms and that is why the model accuracy suffered once the stop words were removed.

As seen in Table 3 our basic model suffers from low recall. A manual inspection of the toponyms in the dataset revealed that either their first letter is capitalized (e.g. *Mexico*) or all their letters are capitalized (e.g. *UK*). As mentioned in Section 3.1 we used this information in an attempt to help the DFFNN learn more structure from the small dataset. As a result the F1 performance of the model increased form 70.26% to 72.27% (see Table 3 #5).

In order to help the neural network better understand and model the structure of the sentences, we experimented with part of speech (POS) tags as part of our feature embeddings. We used the NLTK POS tagger [4] which uses the Penn Treebank tagset. As shown in Table 3 (#6), the POS tags significantly improve the recall of the network (from 66.69% to 70.37%) hence leading to a higher performance in F1 (from 72.24% to 73.79%). The POS tags help the DFFNN to better learn the structure of the sentences and take advantage of more contextual information (see Section 4.3).

4.3 Effect of Window Size

In order to measure the effect of the size of the context window, we varied this value using the basic DFFNN. As seen in Figure 3, the best performance is achieved at $c = 2$. With values over this threshold, the DFFNN overfits as it cannot extract any meaningful structure. Due to the small size of the data set, the DFFNN is not able to learn the structure of the sentences, hence increasing the context window alone does not help the performance. In order to help the neural network better understand and use the contextual structure of the sentences in its predictions, we experimented with part of speech (POS) tags as part of our feature embeddings. As shown in Figure 3, the POS tags help the DFFNN to take advantage of more contextual information as a result the DFFNN with POS embeddings achieves a higher performance on larger window sizes. The context window for which the DFFNN achieved its highest performance on the development set was $c = 5$, and on the test set the performance was increased from 72.24% to 77.10% (see Table 3 #6).

4.4 Effect of the Loss Function

As shown in Table 2 most models suffer from a lower recall than precision. The dataset is quite imbalanced, that is the number of non-toponyms are much higher than toponyms (99% vs 1%). Hence, the neural network prefers to optimize its performance by concentrating its efforts on correctly predicting the labels for the dominant class (non-toponym). In order to minimize the gap between recall than precision, we experimented with a weighted loss function. We adjusted the importance of predicting the correct labels experimentally and found that

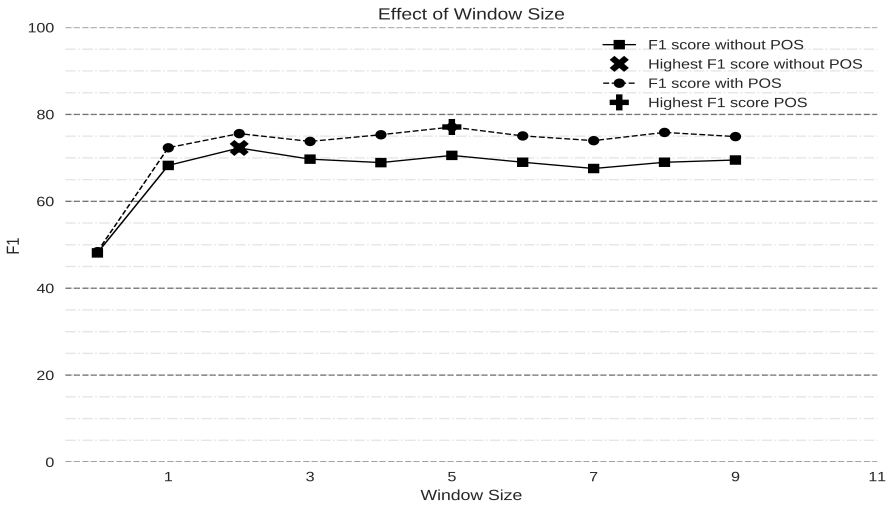

Fig. 3. Effect of context window on performance of the model with and without POS features. (DFFNN Basic+P+S and DFFNN Basic+P+S+C+P)

by weighing the toponyms 2 times more than the non-toponyms, the system reaches an equilibrium in the precision and recall measure, leading to a higher F1 performance. (This is indicated by "w" in Table 3 row #7)

4.5 Use of Lemmas

Neural networks require large datasets to learn structures and they learn better if the dataset contains similar examples so that the system can cluster them in its learning process. Since our dataset is small and the Wikipedia-PubMed embeddings suffer from 28.61% OOV words (see Table 4), we tried to help the network better cluster the data by adding the lemmatized word embeddings of the words to the feature embeddings and see how our best model reacts to it. As shown in Table 3 (#8), this improved the F1 measure significantly (from 77.10% to 80.13%).

Furthermore, we picked 2 random toponyms and 2 random non-toponyms to visualize the confidence of our best model and the baseline model in their prediction as given by the softmax function (see Equation 2). Figure 4 shows that our model produces much sharper confidence in comparison to the baseline model.

5 Discussion

Overall our best model (DFFNN #8 in Table 3) is made out of the basic DFFNN plus capitalized feature, POS embeddings, weighted loss function, and lemma-

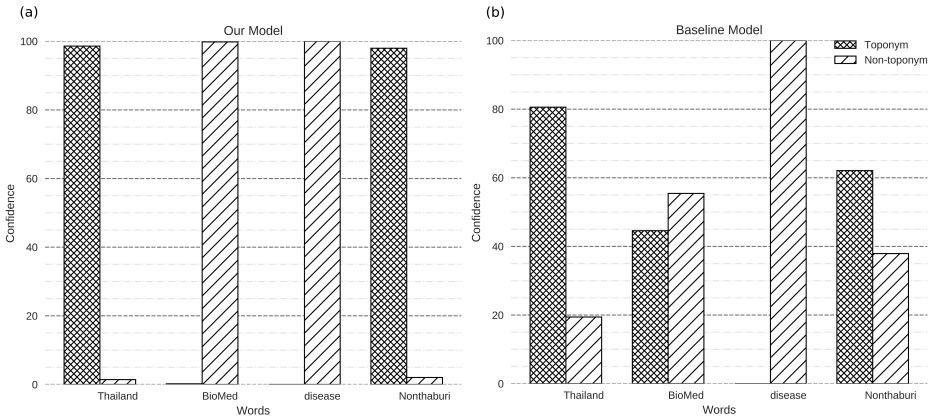

Fig. 4. (a) Confidence of our proposed model in its categorical predictions. (b) Confidence of the baseline in its categorical predictions.

tization feature. The experiments and results described in Section 4 underlines the importance of linguistic insights in the task of toponym detection. Ideally the system should learn all these insights and features by itself given access to enough data. However, when the data is scarce, as in our case, we should take advantage of the linguistic structure of the data for better performance.

Our experiments also underline the importance of domain specific word embedding models. These models reduce OOV words and also present us with embeddings that capture the relation of the words in the specific domain of study.

6 Conclusion and Future Work

This paper presented the approach we used to participate to the recent SemEval task 12 shared task on toponym resolution [28]. Our best DFFNN approach took advantage of domain specific embeddings as well as linguistic features. It achieves a significant increase in F-measure compared to the baseline system (from 69.74% to 80.13%). However, as the official results were not available at the time of writing, comparison with other approaches cannot be done at this time.

The focus of this paper was to propose a deep learning based model for toponym detection and experiment with the use of external linguistic features and domain specific information. The model was evaluated using the recent SemEval task 12 datatset [28] and shows that domain specific embedding as well as some linguistic features do help in toponym detection in medical journals.

One of the main factors preventing us from exploring deeper models, was the small size of the data set. With more human annotated data the models could be extended for better performance. However, since human annotated data is expensive to produce, we suggest distant supervision [15] to be explored for

further increasing performance. As our experiments pointed out, the model could heavily benefit from linguistic insights, hence equipping the model with more linguistic driven features could potentially lead to a higher performing model. We did not have the time or computational resources to explore recurrent neural architectures, however future work could be done focusing on these models.

Acknowledgments

This work was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Amitay, E., Har'El, N., Sivan, R., Soffer, A.: Web-a-where: geotagging web content. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 273–280. ACM (2004)
2. Ardanuy, M.C., Sporleder, C.: Toponym disambiguation in historical documents using semantic and geographic features. In: Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage. pp. 175–180. ACM (2017)
3. Benson, D.A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Sayers, E.W.: Genbank. Nucleic acids research **41**(D1), D36–D42 (2012)
4. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. "O'Reilly Media, Inc." (2009)
5. Chiu, J.P., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics **4**, 357–370 (2016)
6. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning. pp. 160–167. ACM (2008)
7. DeLozier, G., Baldridge, J., London, L.: Gazetteer-independent toponym resolution using geographic word profiles. In: AAAI. pp. 2382–2388 (2015)
8. Garbin, E., Mani, I.: Disambiguating toponyms in news. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. pp. 363–370. Association for Computational Linguistics (2005)
9. Gautret, P., Botelho-Nevers, E., Brouqui, P., Parola, P.: The spread of vaccine-preventable diseases by international travellers: A public-health concern. Clinical Microbiology and Infection **18**, 77 – 84 (2012). https://doi.org/https://doi.org/10.1111/j.1469-0691.2012.03940.x, `http://www.sciencedirect.com/science/article/pii/S1198743X14613653`
10. Gelernter, J., Balaji, S.: An algorithm for local geoparsing of microtext. GeoInformatica **17**(4), 635–667 (2013)
11. Google: Pretrained word and phrase vectors. `https://code.google.com/archive/p/word2vec/` (2019), accessed: 2019-01-10
12. Green, A.D., Roberts, K.I.: Recent trends in infectious diseases for travellers. Occupational Medicine **50**(8), 560–565 (2000). https://doi.org/10.1093/occmed/50.8.560, `http://dx.doi.org/10.1093/occmed/50.8.560`

13. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
14. Kienreich, W., Granitzer, M., Lux, M.: Geospatial anchoring of encyclopedia articles. In: Tenth International Conference on Information Visualisation (IV'06). pp. 211–215 (July 2006). https://doi.org/10.1109/IV.2006.57
15. Krause, S., Li, H., Uszkoreit, H., Xu, F.: Large-scale learning of relation-extraction rules with distant supervision from the web. In: International Semantic Web Conference. pp. 263–278. Springer (2012)
16. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
17. Li, L., Jin, L., Jiang, Z., Song, D., Huang, D.: Biomedical named entity recognition based on extended recurrent neural networks. In: 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 649–652 (Nov 2015). https://doi.org/10.1109/BIBM.2015.7359761
18. Lieberman, M.D., Samet, H.: Multifaceted toponym recognition for streaming news. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. pp. 843–852. ACM (2011)
19. Magge, A., Weissenbacher, D., Sarker, A., Scotch, M., Gonzalez-Hernandez, G.: Deep neural networks and distant supervision for geographic location mention extraction. Bioinformatics **34**(13), i565–i573 (2018)
20. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
21. Moen, S., Ananiadou, T.S.S.: Distributional semantics resources for biomedical text processing. In: Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan. pp. 39–43 (2013)
22. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning. pp. 1310–1318 (2013)
23. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
24. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. `https://nlp.stanford.edu/projects/glove/` (2014), accessed: 2019-01-10
25. Purves, R.S., Clough, P., Jones, C.B., Arampatzis, A., Bucher, B., Finch, D., Fu, G., Joho, H., Syed, A.K., Vaid, S., et al.: The design and implementation of spirit: a spatially aware search engine for information retrieval on the internet. International journal of geographical information science **21**(7), 717–745 (2007)
26. Qin, T., Xiao, R., Fang, L., Xie, X., Zhang, L.: An efficient location extraction algorithm by leveraging web contextual information. In: proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. pp. 53–60. ACM (2010)
27. Santos, J., Anastácio, I., Martins, B.: Using machine learning methods for disambiguating place references in textual documents. GeoJournal **80**(3), 375–392 (2015)
28. SemEval: Toponym resolution in scientific papers. `https://competitions.codalab.org/competitions/19948#learn_the_details-overview` (2018), accessed: 2019-01-20
29. Tamames, J., de Lorenzo, V.: Envmine: A text-mining system for the automatic extraction of contextual information. BMC bioinformatics **11**(1), 294 (2010)

30. Taylor, M.: Reduced Geographic Scope as a Strategy for Toponym Resolution. Ph.D. thesis, Northern Arizona University (2017)
31. Wang, P., Qian, Y., Soong, F.K., He, L., Zhao, H.: A unified tagging solution: Bidirectional lstm recurrent neural network with word embedding. arXiv preprint arXiv:1511.00215 (2015)
32. Weissenbacher, D., Sarker, A., Tahsin, T., Scotch, M., Gonzalez, G.: Extracting geographic locations from the literature for virus phylogeography using supervised and distant supervision methods. AMIA Summits on Translational Science Proceedings **2017**, 114 (2017)
33. Weissenbacher, D., Tahsin, T., Beard, R., Figaro, M., Rivera, R., Scotch, M., Gonzalez, G.: Knowledge-driven geospatial location resolution for phylogeographic models of virus migration. Bioinformatics **31**(12), i348–i356 (2015). https://doi.org/10.1093/bioinformatics/btv259, `http://dx.doi.org/10.1093/bioinformatics/btv259`