UNH at SemEval-2019 Task 12: Toponym Resolution in Scientific Papers

Matthew Magnusson
Department of Computer Science
University of New Hampshire
mfm2@cs.unh.edu

Laura Dietz
Department of Computer Science
University of New Hampshire
dietz@cs.unh.edu

Abstract

The SemEval-2019 Task 12 is toponym resolution in scientific papers. We focus on Subtask 1: Toponym Detection which is the identification of spans of text for place names mentioned in a document. We propose two methods: 1) sliding window convolutional neural network using ELMo embeddings (CNN-ELMo), and 2) sliding window multi-Layer perceptron using ELMo embeddings (MLP-ELMo). We also submit a bi-directional LSTM with Conditional Random Fields (bi-LSTM) as a strong baseline given its state-of-art performance in Named Entity Recognition (NER) task. Our best performing model is CNN-ELMo with a F1 of 0.844 which was below bi-LSTM F1 of 0.862 when evaluated on overlap macro detection. Eight teams participated in this subtask with a total of 21 submissions.

1 Introduction

Toponyms are textual spans of text identifying geospatial locations. This can range from the canonical name of populated places, such as "London" to direct or indirect mentions of geographic entities. The parsing of geographic locations from unstructured text is considered an open challenge due to domain diversity, place name ambiguity, metonymic language and often limited leveraging of context (Gritta et al., 2018). Many scientific publications contain toponyms which can be challenging to extract automatically. Specifically, names of institutions and viruses often contain geographic references which may confuse the extractor. Often, the extractor needs to handle noisy text parsed from PDF versions of scientific articles which can introduce artifacts.

In Task 12, a toponym is defined to include proper names and geographic entities but to exclude indirect mentions of places and metonyms. Additional discussion of the motivation and task description is available at the task web site.[1]

2 Related Work

There is significant work in the area of toponym detection (Matsuda et al., 2015; D. Lieberman et al., 2010) and the closely related fields of named entity recognition (NER) (Li et al., 2018) and entity mention detection (EMD) (Shen et al., 2015) with many different approaches. State-of-the-art named entity detection models have historically employed a combination of hand-crafted features, rules, natural language processing (NLP), string-pattern matching, and domain knowledge using supervised learning on manually annotated corpora (Piskorski and Yangarber, 2018). A common approach to toponym detection has been to utilize place name gazetteers which are directories of geographic names and their corresponding geolocations to perform string matching of place names in text (D. Lieberman et al., 2010).

Contemporary approaches in entity detection have utilized neural-based architectures. (Collobert et al., 2011) propose a window-based, multi-layer, dense feed-forward neural architecture using word embeddings concatenated with orthographic features and a gazetteer as an input layer with a hard Tanh output layer for superior performance on a standard NER task. Huang et al. (2015) utilise a bi-directional LSTM with a sequential conditional random layer using a gazetteer and Senna word embeddings to obtain superior performance. Magge et al. (2018) achieves state-of-the-art results in toponym detection by utilizing a window-based deep neural network, word embeddings trained on a domain-specific corpus, orthographic features, and a gazetteer.

[1] https://competitions.codalab.org/competitions/19948

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1308–1312
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

Table 1: Gold Standard Corpus Statistics

	Documents	Tokens	Toponyms
Train	72	396,668	3,637
Valid	32	179,443	2,141
Test	45	253,159	4,616
Total	149	829,720	10,394

3 Data

A gold standard corpus, composed of 150 full text journal articles in open access from PubMed Central (PMC), is provided by the task organizers.[2] Additional information can be found at Weissenbacher et al. (2017) for the general approach followed by the task organizers for developing the corpus. Table 1 highlights the gold standard corpus statistics.

4 Approach

Our approach is motivated by the simplicity and strong performance of windows-based approaches on the NER and toponym task with the strong performance of deep contextual embeddings on related NLP tasks. Two different neural based approaches are undertaken by the team: 1) sliding windows convolutional neural network using deep embeddings, and 2) sliding window multi-layer perceptron using deep embeddings. The embeddings are composed of an ELMo contextual embedding concatenated with hand-crafted features.

4.1 Embeddings

The ELMo embeddings (Gardner et al., 2018) are learned functions of the internal states of a deep bidirectional language model (biLM) that has been pre-trained on the 1B Word Benchmark. These vectors are developed from the concatenation of each of the 1,024 length vector outputs from the model for each token and are a function of the complete input sentence.

For each token in the context of its sentence, a vector representation is generated by concatenating the ELMo model embedding with the one-hot encoding of orthographic features and an additional flag bit indicating if the token was contained within the set of gazetteer tokens. This resulted in a vector of length 3,081 for each token. A padding vector of all 0s was used for the sliding window neural models.

4.2 Hand-crafted Features

Hand-crafted features were added as they slightly improve model performance when compared to using the ELMo embeddings alone for the input layer to the neural models.

Orthographic Features: a one hot encoding is assigned to each token based on its orthographic structure: only numeric, all lower case alphabetic characters, all upper case alphabetic characters, title-case alphabetic characters, mixed case (not title-case) alphabetic characters, alphabetic characters with numeric, padding token, and the "other" for the remaining tokens not matched by previously listed features. Alphabetic characters are UTF-8.

Gazeteer Features: a set of toponynm tokens is generated from the entries in GeoNames.[3] For example, for the entry in Geonames,"Gulf of Mexico", the tokens "Gulf", "of", and "Mexico" are added to the toponym set. This is used as a binary feature for the presence of the parsed token in the constructed Geonames token set.

4.3 Implementation details

The documents are parsed into sentences and tokenized using the open-source NLP library Spacy.[4]

Pre-trained embeddings are provided by Pyysalo et al. (2013).[5] which are generated from Wikipedia, PubMed, and PMC texts using the word2vec tool. They are 200-dimensional vectors trained using the skip-gram model with a window size of 5, hierarchical softmax training, and a frequent word subsampling threshold of 0.001. These vectors are used in the baseline and the bi-LSTM with CRF models.

ELMo embeddings are generated using the AllenNLP tool .[6] The deep learning library Keras 2.2.1 [7] is used for training the neural models. Tensorflow 1.12[8] is the backend used for training and evaluating all of the models attempted. In training the models, the Adam optimizer in Keras is used. Additional code and data will be available in an on-line appendix.[9]

[2] We are unable to successfully parse one of the documents from the train set due to an encoding error

[3] https://www.geonames.org/export/

[4] https://spacy.io/

[5] http://bio.nlplab.org/

[6] https://github.com/allenai/allennlp/

[7] https://keras.io/

[8] https://www.tensorflow.org/

[9] https://cs.unh.edu/ mfm2/index.html

4.4 Models

We compare the following models:

MLP-ELMo: A sliding window (size = 5) is applied to each sentence with padding vectors applied to boundary tokens. The input layer to the neural models is a 5 x 3081 matrix using the ELMo-based embeddings. The input layer is connected to two fully connected layers with 128 hidden units each and relu activation. The output is a sigmoid with a binary output to indicate if the token is part of a toponym.

CNN-ELMo: A sliding window (size = 5) is applied to each sentence with padding vectors applied to boundary tokens. The input layer is a 5 x 3081 layer. The input layer is two 1d convolutional layers with filter sizes of 250 and a kernel size of 3. A global 1-d max pooling layer follows the convolutional layers. Two fully connected layers with 100 hidden units each and relu activation follow max pooling. A sigmoid function is applied in output layer to indicate if the token is part of a toponym.

4.5 Baseline

Two models are used for evaluation: 1) a sliding window mlp provided by the task organizers, and 2) bi-LSTM with CRF. The bi-LSTM with CRF model demonstrates state-of-the-art results on NER and is used as an additional strong benchmark for model comparison.

MLP-Baseline: The task organizers provide a state-of-the-art geoparser as a strong baseline. The system has a specific component for toponym detection using a two-layer feedforward neural network (200 hidden units per layer) as described in Magge et al. (2018). The baseline features a sliding window (size = 5) over each sentence using Wikipedia-Pubmed-PMC word2vec embeddings for token encoding. The baseline did not include a gazetter-based lookup but did incorporate orthographic structure of the tokens: 1) All Caps - ASCII, 2) First letter capitalized - ASCII, and 3) first letter not-capitalized - ASCII. The baseline also uses separately trained vectors if the token contained a digit or unknown token in the vocabulary.

Bi-LSTM-Baseline: This strong baseline implementation utilizes the code developed by Reimers and Gurevych (2017).[10] Input sentences for the model are generated in the CoNLL format with the IOB representation for labeled toponyms in the training data. The embeddings are the Wikipedia-Pubmed-PMC word2vec vectors. Each LSTM has a size of 100 and is trained with a dropout of 0.25. Character embeddings are generated using a convolutional neural network and the maximum character length is 50. The model is fit over 25 training epochs.

5 Experiment Evaluation

5.1 Metrics

The metrics for evaluation are precision, recall and F-measure. Two variants are provided for toponym detection: strict and overlapping measures. In the strict measure, mentions match the gold standard if they match exact span boundaries. In the overlapping measure, a match occurs when the mention and gold standard share any common span of text.

There are two methods for computing precision and recall: micro averaging, and macro averaging. In micro averaging the corpus of documents is treated as one large document when calculating precision and recall. In macro averaging precision, recall and f-measure are calculated on a per document basis, and then the results are averaged.

5.2 Results

The model results are shown in Tables 2, 3, 4, and 5. The best model in the task by the Team "DM_NLP" (Davy Weissenbacher, 2019) is provided for comparison with the results our team achieves. The F1 scores of the two sliding window models and the bi-LSTM benchmark outperforms the task benchmark on all metrics. Each model is run once with a random model parameter initialization. The MLP-ELMo model had a similar feedforward structure and approach as the baseline neural model. The primary difference is the embedding vectors used. For both strict and overlap toponym detection, MLP-ELMo model achieves the same or higher precision and recall than the baseline.

The convolutional network using the ELMo-based embeddings exhibits higher performance on the f1 score relative to MLP-ELMo, however precision is higher and recall is lower for both strict and overlap measures.

The best performing model is the bi-LSTM with CRF method. This shows that the sliding window models with deep contextual embeddings did

[10] https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf

Table 2: Overlap Macro

Run	P	R	F1
Bi-LSTM-Baseline	**0.910**	**0.819**	**0.862**
CNN-ELMo	0.908	0.788	0.844
MLP-ELMo	0.886	0.798	0.840
MLP-Baseline	0.864	0.797	0.829
DM_NLP	0.946	0.924	0.935

Table 3: Overlap Micro

Run	P	R	F1
Bi-LSTM-Baseline	0.897	**0.704**	**0.789**
CNN-ELMo	**0.913**	0.697	0.791
MLP-ELMo	0.890	0.737	0.807
MLP-Baseline	0.880	0.687	0.772
DM_NLP	0.954	0.880	0.915

Table 4: Strict Macro

Run	P	R	F1
Bi-LSTM-Baseline	**0.862**	**0.781**	**0.819**
CNN-ELMo	0.836	0.737	0.784
MLP-ELMo	0.811	0.740	0.774
MLP-Baseline	0.791	0.740	0.764
DM_NLP	0.927	0.906	0.916

Table 5: Strict Micro

Run	P	R	F1
Bi-LSTM-Baseline	**0.835**	**0.650**	**0.731**
CNN-ELMo	0.807	0.618	0.700
MLP-ELMo	0.782	0.646	0.707
MLP-Baseline	0.775	0.603	0.678
DM_NLP	0.929	0.856	0.891

not achieve state-of-art performance on this task. However, bi-LSTM with CRF has lower performance than the best model submitted for the task which indicates that other approaches can exceed the performance of a state-of-the-art Named Entity Recognition model.

Fine tuning shows that a window size of 5 yields the best performance on the validation set during training for the sliding window neural models. The sliding window neural models do not require many epochs of training with approximately only three required before overfitting of the training data becomes evident. Adding dropout to training did not appear to improve sliding window model performance.

6 Conclusion

The best performing submission by our team is bi-LSTM with CRF. This is not surprising as this technique has achieved state-of-the-art results in NER NLP tasks. The sliding window models we propose are similar in the approach as the task baseline model. The ELMo-based embeddings do achieve a boost in performance relative to baseline given the richer context and character structure they embed. This indicates that the ELMo-derived embeddings are superior in the task to embeddings trained on a domain-specific corpus using word2vec. However, for both overlap and strict macro the recall for MLP-ELMo is identical to the baseline model.

Bi-LSTM with CRF and the baseline neural model are noteworthy in that they are both able to extract toponym mentions only using context from embeddings to acheive high-quality results without relying on the presence of a gazetteer. An open question is if a gazetter or other knowledge graph structure could be incorporated into a deep neural model using contextual embeddings to achieve superior performance. It is also not clear why CNN-ELMo has lower recall than MLP-ELMo and baseline.

The results suggest that a sliding window model can be enhanced by better-quality embeddings and a convolutional component. The sliding window model approach is attractive due to its relatively straight-forward implementation and quick training time. The results achieved by bi-LSTM with CRF and the model submitted by DM_NLP, suggest that other approaches may ultimately generate superior performance on the toponym detection task.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1846017. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Michael D. Lieberman, Hanan Samet, and Jagan Sankaranarayanan. 2010. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups.

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

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *ACL workshop for NLP Open Source Software*.

Milan Gritta, Mohammad Taher Pilehvar, Nut Limsopatham, and Nigel Collier. 2018. What's missing in geographical parsing? *Lang. Resour. Eval.*, 52(2):603–623.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *CoRR*, abs/1508.01991.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2018. A survey on deep learning for named entity recognition. *CoRR*, abs/1812.09449.

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

Koji Matsuda, Akira Sasaki, Naoaki Okazaki, and Kentaro Inui. 2015. Annotating geographical entities on microblog text. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 85–94. Association for Computational Linguistics.

Jakub Piskorski and Roman Yangarber. 2018. Chapter 2 information extraction : Past , present and future.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing.

Nils Reimers and Iryna Gurevych. 2017. 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)*, pages 338–348, Copenhagen, Denmark.

W. Shen, J. Wang, and J. Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge & Data Engineering*, 27(2):443–460.

Davy Weissenbacher, Abeed Sarker, Tasnia Tahsin, Matthew Scotch, and Graciela Gonzalez. 2017. Extracting geographic locations from the literature for virus phylogeography using supervised and distant supervision methods. *AMIA Jt Summits Transl Sci Proc*, 2017:114–122. 28815119[pmid].