Dataset mention extraction in scientific articles using a BiLSTM-CRF model

Tong Zeng[1,2] and Daniel Acuna[1]*

[1]School of Information Studies, Syracuse University, Syracuse, USA
[2]School of Information Management, Nanjing University, Nanjing, China

Abstract. Datasets are critical for scientific research, playing a role in replication, reproducibility, and efficiency. Researchers have recently shown that datasets are becoming more important for science to function properly, even serving as artifacts of study themselves. However, citing datasets is not a common or standard practice in spite of recent efforts by data repositories and funding agencies. This greatly affects our ability to track their usage and importance. A potential solution to this problem is to automatically extract dataset mentions from scientific articles. In this work, we propose to achieve such extraction by using a neural network based on a BiLSTM-CRF architecture. Our method achieves $F_1 = 0.885$ in social science articles released as part of the Rich Context Dataset. We discuss limitations of the current datasets and propose modifications to the model to be done in the future.

1 Introduction

Science is fundamentally an incremental discipline that depends on previous scientist's work. Datasets form an integral part of this process and therefore should be shared and cited as any other scientific output. This ideal is far from reality; the credit that datasets currently receive does not correspond to their actual usage. One of the issues is that there is no standard approach for citing them. Interestingly, while datasets are still used and mentioned in articles, we lack methods to extract such mentions and properly reconstruct dataset citations. The Rich Context Competition challenge aims at closing this gap by inviting scientists to produce automated dataset mention and linkage detection algorithms. In this article, we detail our proposal to solve the dataset mention step.

* Corresponding author: deacuna@syr.edu

Our approach attempts to provide a first approximation to better give credit and keep track of datasets and their usage.

The problem of dataset extraction has been explored before. Ghavimi et al. (2016) and Ghavimi et al. (2017) use a relatively simple tf-idf representation with cosine similarity for matching dataset identification in social science articles. Their method consists of four major steps: preparing a curated dictionary of typical mention phrases, detecting dataset references, and ranking matching datasets based on cosine similarity of tf-idf representations. This approach achieved an impressive $F_1 = 0.84$ for mention detection and $F_1 = 0.83$, for matching. Singhal and Srivastava (2013) proposed a method using normalized Google distance to screen whether a term is in a dataset. However, this method relies on external services and is not computational efficient. They achieve a good $F_1 = 0.85$ using Google search and $F_1 = 0.75$ using Bing. A somewhat similar project was proposed by Lu et al. (2012). They built a dataset search engine by solving the two challenges: identification of the dataset and association to a URL. They build a dataset of 1000 documents with their URLs, containing 8922 words or abbreviations representing datasets. They also build a web-based interface. This shows the importance of dataset mention extraction and how several groups have tried to tackle the problem.

In this article, we describe a method for extracting dataset mentions based on a deep recurrent neural network. In particular, we used a Bidirectional Long short-term Memory (BiLSTM) sequence to sequence model paired with a Conditional Random Field (CRF) inference mechanism. We tested our model on a novel dataset produced for the Rich Context Competition challenge. We achieve a relatively good performance of $F_1 = 0.885$. We discuss the current noise and duplication present in the dataset and limitations of our model.

2 The dataset

The Rich Context Dataset challenge was proposed by the New York University's Coleridge Initiative (Coleridge Initiative, 2019). The challenge comprised several phases, and participants moved through the phases depending on their performance. We only analyze data of the first phase. This phase contained a list of datasets and a labeled corpus of around 5K publications. Each publication was labeled indicating whether a dataset was mentioned within it and which part of the text mentioned it. The challenge used the accuracy for measuring the performance of the competitors and also the quality of the code, documentation, and efficiency.

We adopt the CoNLL 2003 format (Tjong Kim Sang and De Meulder, 2003) to annotate whether a token is a part of dataset mention. Concretely, we use we use B-DS denotes a token is the first token of a dataset mention, I-DS denote a token is inside of dataset mention, and O means a token is not a part of dataset mention. We then put each token and its corresponding labels in one line and use a empty line as separator between sentences. All the sentences was split by 70%, 15%, 15% as training set, validation set and testing set.

3 The Proposed Method

3.1 Overall view of the architecture

In this section, we propose a model for detecting mentions based on a BiLSTM-CRF architecture. At a high level, the model uses a sequence-to-sequence recurrent neural network that produces the probability of whether a token belongs to a dataset mention. The CRF layer takes those probabilities and estimates the most likely sequence based on constrains between label transitions (i.e., mention–to–no-mention–to-mention has low probability). While this is a standard architecture for modeling sequence labeling, the application to our particular dataset and problem is new.

We now describe in more detail the choices of word representation, hyper-parameters, and training parameters. A schematic view of the model is in Fig 1 and the components are as follows:

1. Character embedding layer: treat a token as a sequence of characters and encode the characters by using a bidirectional LSTM to get a vector representation.
2. Word embedding layer: mapping each token into fixed sized vector representation by using a pre-trained word vector.
3. One BiLSTM layer: make use of Bidirectional LSTM network to capture the high level representation of the whole token sequence input.
4. Dense layer: project the output of the previous layer to a low dimensional vector representation of the the distribution of labels.
5. CRF layer: find the most likely sequence of labels.

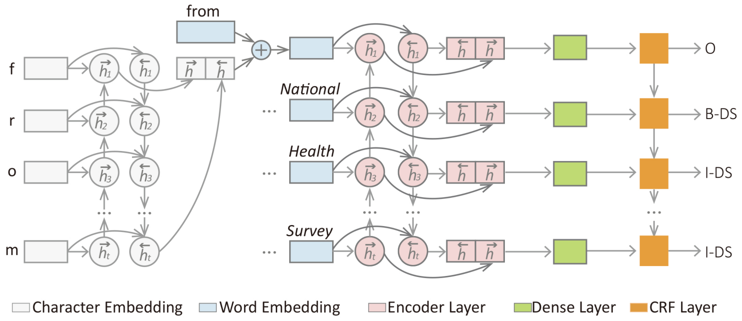

Fig. 1. Network Architecture of BiLSTM-CRF network

3.2 Character Embedding

Similar to the bag of words assumption, we can consider a token is composed by a bag of characters. In this layer, we convert each token as a sequence of characters, then feed the sequence into a bidirectional LSTM network to get a fixed length representation of the token. After learning the bidirectional LSTM network, we can solve the out-of-vocabulary problem for pre-trained word embeddings.

3.3 Word Embedding

The embedding is the first layer of our network and it is responsible for mapping the word from string into vectors of numbers as the input for other layers on top. For a given sentence S, we first convert it into a sequence consisting of n tokens, $S = \{c_1, c_2, \cdots, c_n, \}$. For each token $c_i$we lookup the embedding vector x_i from a word embedding matrix $M^{tkn} \in \mathbb{R}^{d|V|}$, where the d is the dimension of the embedding vector and the V is the Vocabulary of the tokens. In this paper, the matrix M^{tkn} is initialized by a pre-trained GloVe vectors(Pennington et al., 2014), but will be updated by learning from our corpus.

3.4 LSTM

Recurrent neural network (RNN) is a powerful tool to capture features from sequential data, such as temporal series, and text. RNN could capture long-distance dependency in theory but it suffers from the gradient exploding/vanishing problems (Pascanu et al., 2013). The Long short-term memory (LSTM) architecture was proposed by Hochreiter and Schmidhuber (1997) and it is a variant of RNN which copes with the gradient problem. LSTM introduces several gates to control the proportion of information to forget from previous time steps and to pass to the next time step. Formally, LSTM could be described by the following equations:

$$i_t = \sigma(W_i x_t + W_i h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W_f x_t + W_f h_{t-1} + b_f) \tag{2}$$

$$g_t = tanh(W_g x_t + W_g h_{t-1} + b_g) \tag{3}$$

$$o_t = \sigma(W_o x_t + W_o h_{t-1} + b_o) \tag{4}$$

$$c_t = f_t \bigotimes c_{t-1} + i_t \bigotimes g_t \tag{5}$$

$$h_t = o_t \bigotimes tanh(c_t) \tag{6}$$

where the σ is the sigmoid function, $\bigotimes$ denotes the dot product, b is the bias, W is the parameters, x_t is the input at time t, c_t is the LSTM cell state at time t and h_t is hidden state at time t. The i_t, f_t, o_t and g_t are named as input, forget, output and cell gates respectively, they control the information to keep in its state and pass to next step.

LSTM get information from the previous steps, that is left context in our task. However, it is important to consider the information in the right context. A solution of this information need is bidirectional LSTM (Graves et al., 2013). The idea of Bi-LSTM is using two LSTM layers and feed in each layer with sequence forwards and backwards separately, and then concatenate the hidden states of the two LSTM to modeling both the left and right contexts

$$h_t = [\overrightarrow{h_t} \oplus \overleftarrow{h_t}] \tag{7}$$

Finally, the outcomes of the states are taken by a Conditional Random Field (CRF) layer that takes into account the transition nature of the beginning, intermediate, and ends of mentions. For a reference of CRF, refer to (Lafferty et al., 2001)

4 Results

In this work, we wanted to propose a model for the Rich Context Competition challenge. We propose a relatively standard architecture based on a BiLSTM-

CRF recurrent neural network. We now describe the results of this network on the dataset provided by the competition.

For all of our results, we use F_1 as the measure of choice. This measure is the harmonic average of the precision and recall and it is the standard measure used in sequence labeling tasks. This metric varies from 0 to 1, and the unit is the highest possible value. Our method achieved a relatively high F_1 of 0.885 for detecting mentions, in line with previous studies.

Table 1. Performance of proposed network

Models	GloVe size	Dropout rate	Precision	Recall	F_1
m1	50	0.0	0.884	0.873	0.878
m2	50	0.5	0.877	0.888	0.882
m3	100	0.0	0.882	0.871	0.876
m4	100	0.5	0.885	0.885	**0.885**
m5	200	0.0	0.882	0.884	0.883
m6	200	0.5	0.885	0.880	0.882
m7	300	0.0	0.868	0.886	0.877
m8	300	0.5	0.876	0.878	0.877

We train models using the training data, monitor the performance using the validation data (we stop training if the performance doesn't improve for the last 10 epochs). We're using the Adam optimizer with learning rate of 0.001 and the batch size equal to 64. The hidden size of LSTM for character and word embedding is 80 and 300, respectively. For the regularization methods to avoid over-fitting, we use L2 regularization with alpha set to 0.01, we also use dropout rate equal to 0.5, and early stop strategy with the patience set to 10. We trained 8 models with the combination of different GloVe vector size (50, 100, 300 and 300) and dropout rate (0.0, 0.5). The performances are reported on the test dataset in Table 1. The best model is trained by word vector size 100 and dropout rate 0.5 with F_1 score 0.885.

We also found some limitations to the dataset. Firstly, we found that mentions are nested (e.g. HRS, RAND HRS, RAND HRS DATA are linked to the same dataset). The second issue most of the mentions have ambiguous relationships to datasets. In particular, only 17,267 (16.99%) mentions are linked to one dataset, 15,292 (15.04%) mentions are listed to two datasets, and 12,624 (12.42%) are linked to three datasets. If these difficulties are not overcome, then the predictions from the linkage process will be noisy and therefore impossible to tell apart.

5 Conclusion

In this work, we report a high accuracy model for the problem of detecting dataset mentions. Because our method is based on a standard BiLSTM-CRF architecture, we expect that updating our model with recent developments in neural networks would only benefit our results. We also provide some evidence of how difficult we believe the linkage step of the challenge could be if the dataset noise are not lowered.

One of the shortcomings of our approach is that the architecture is lacking some modern features of RNN networks. In particular, recent work has shown that attention mechanisms are important especially when the task requires spatially distant information, such as this one. These benefits could also translate to better linkage. We are exploring new architectures using self-attention and multiple-head attention. We hope to explore these approaches in the near future.

Our proposal, however, is surprisingly effective. Because we have barely modified a general RNN architecture, we expect that our results will generalize relatively well either to the second phase of the challenge or even to other disciplines. We would emphasize, however, that the quality of the dataset has a great deal of room for improvement. Given how important this task is for the whole of science, we should try to strive to improve on the quality of these datasets so that techniques like this one can be more broadly applied. The importance of dataset mention and linkage therefore could be fully appreciated by the community.

Acknowledgements

Tong Zeng was funded by the China Scholarship Council #201706190067. Daniel E. Acuna was funded by the National Science Foundation awards #1646763 and #1800956.

Bibliography

Coleridge Initiative (2019). Rich context competition.

Ghavimi, B., Mayr, P., Lange, C., Vahdati, S., and Auer, S. (2017). A semi-automatic approach for detecting dataset references in social science texts. *Information Services & Use*, 36(3-4):171–187.

Ghavimi, B., Mayr, P., Vahdati, S., and Lange, C. (2016). Identifying and Improving Dataset References in Social Sciences Full Texts. *arXiv:1603.01774 [cs]*.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lu, M., Bangalore, S., Cormode, G., Hadjieleftheriou, M., and Srivastava, D. (2012). A Dataset Search Engine for the Research Document Corpus. In *2012 IEEE 28th International Conference on Data Engineering*, pages 1237–1240, Arlington, VA, USA. IEEE.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Singhal, A. and Srivastava, J. (2013). Data Extract: Mining Context from the Web for Dataset Extraction. *International Journal of Machine Learning and Computing*, pages 219–223.

Tjong Kim Sang, E. F. and De Meulder, F. (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.