Knowledge Graph Extraction

Yueen Ma

May 2020

1 Name Entity Recognition

To extract a knowledge graph, it is better that we know what name entities are there in the sentence first, so that we could build relations between the recognized name entities at a later stage.

To extract name entities, we used a BiLSTM-CNNs-CRF model [5]. Studies have shown that CNN is effective for extracting morphological information [2, 1]. Therefore, this model uses CNN for character-level representation. The output of the CNN layer is concatenated with GloVe word embedding [8] to represent a word.

Next, word-level information is encoded into context information by Bi-directional LSTM or BiLSTM [3, 7]. Because we are using the IOB tagging format [9], label sequence follows certain rules. For instance, I-ORG cannot follow I-PER. Therefore, label sequences are modeled jointly using a conditional random field (CRF) [4].

We retrained the model on a combination of our dataset and a part of the CoNLL-2003 dataset [10] because we have a relatively small dataset and need a part of it for testing. The CoNLL-2003 dataset has two classes which also appears in our dataset: 'Person' and 'Organization'. In addition, we added a list of known organizations to our dataset. Since it is hard to draw a clear line between the class 'Title' and the class 'Role', we decided to collapse them into one class. The performance of the model is shown in Table 1.

2 Relation Extraction

2.1 Nearest Person

This baseline algorithm is based on the simple idea that a non-person name entity is often related to a person entity nearby. For example, one could say 'General Lamidi Adeosun' where 'Lamidi Adeosun' has the rank 'General', as in one of the documents in our dataset. Therefore, the algorithm merely relates a non-person name entity to the person name entity immediately to the right; if there is no person name entity to its right, then the algorithm relates it to the nearest person entity no matter where the person entity is.

2.2 Shortest Dependency Path

Instead of using the distances between name entities in raw text, we can take syntactic information into account. This method relates a non-person name entity to the person name entity to which the dependency path is shortest. This relies largely on how well the dependency parser performs, so we used a state of the art dependency parser [6]. Since one name entity could span multiple tokens, we only use the shortest path among many possible paths between the tokens of two name entities.

In our dataset, if we impose a constraint that the algorithm can only choose between the two persons that appear immediately to the left and right, the performance can be improved.

2.3 Neural Network

We used a neural network to incorporate the edge types along the dependency path and name entity class. We use the phrase "path pattern" to refer to the list of edge types along a dependency path. The path patterns are encoded into one-hot vectors where the less frequent path patterns are treated as a single "unknown" category. Since there are multiple possible persons in a sentence, there will be an one-hot vector for each person.

In addition, the path length is concatenated to the one-hot vector to compensate for the loss of information when we replace the less frequent path patterns. Multiple one-hot + length vectors of different persons are concatenated together along with a small one-hot vector which encodes the type of the non-person name entity, which makes up the input of the neural network.

Class	True Positive	False Positive	False Negative	Precision	Recall	F1 Score
Person	87	13	6	0.87	0.94	0.90
Rank	80	14	11	0.85	0.88	0.86
Organization	103	33	31	0.76	0.77	0.76
Title/Role	85	20	23	0.81	0.79	0.80
All Classes	355	80	71	0.82	0.83	0.82

Table 1: NER model evaluation

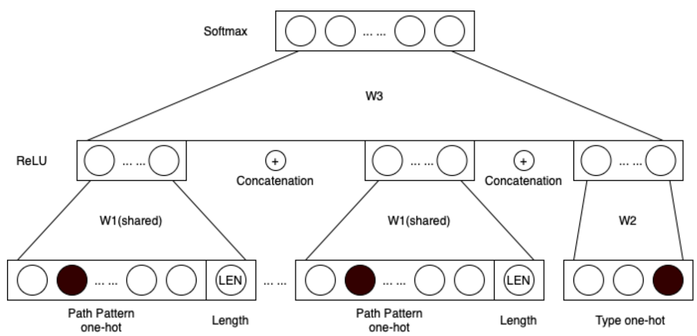

Figure 1: The architecture of the model

Method	Precision	Recall	F1 score
Nearest Person	0.567	0.701	0.627
Shortest Dep. Path (No constraint)	0.625	0.765	0.687
Shortest Dep. Path (With constraint)	0.679	0.833	0.748
Neural Network	0.620	0.767	0.685

Table 2: RE algorithms evaluation

The first layer has a set of shared weights for those one-hot + length vectors and a separate set of weights for the name entity type vector. The second layer is a dense layer whose output is activated by a softmax layer. The softmax layer outputs a vector where the largest element corresponds to the target person. The architecture is shown in Figure 1 and its performance is shown in Table 2.

References

[1] Jason P.C. Chiu and Eric Nichols. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

[2] Cícero Nogueira Dos Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1818–II–1826. JMLR.org, 2014.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[4] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[5] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf, 2016.

[6] Dat Quoc Nguyen and Karin Verspoor. An improved neural network model for joint POS tagging and dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 81–91, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[7] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks, 2012.

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[9] Erik F. Tjong Kim Sang and Jorn Veenstra. Representing text chunks. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, pages 173–179, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics.

[10] Erik F. Tjong Kim Sang and Fien De Meulder. 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*, pages 142–147, 2003.