Extracting Definitions and Hypernyms with a Two-Phase Framework

Yifang Sun(✉), Shifeng Liu, Yufei Wang, and Wei Wang

The University of New South Wales, Sydney, Australia
{yifangs,weiw}@cse.unsw.edu.au, {shifeng.liu,yufei.wang}@unsw.edu.au

Abstract. Extracting definition sentences and hypernyms is the key step in knowledge graph construction as well as many other NLP applications. In this paper, we propose a novel supervised two-phase machine learning framework to solve both tasks simultaneously. Firstly, a joint neural network is trained to predict both definition sentences and hypernyms. Then a refinement model is utilized to further improve the performance of hypernym extraction. Experiment result shows the effectiveness of our proposed framework on a well-known benchmark.

Keywords: Definition extraction · Hypernym extraction

1 Introduction

Both definition extraction and hypernym extraction are fundamental tasks in knowledge graph construction. For example, the first step of Wikipedia BiTaxonomy Project [4], which produced a taxonomized version of Wikipedia, is to extract definitions and hypernyms. They also play important roles in many other NLP tasks such as relation extraction and question answering.

To solve these problems, traditional lexico-syntactic pattern based methods focus on finding hypernym–hyponym pairs in one sentence and take the sentence as definitional [8]. The patterns are sequences of words such as "*is a*" or "*refers to*", which are either manually crafted or semi-automatically generated. Pattern based methods suffer from both low precision and low recall. On one hand, the patterns are usually noisy, which hurts the precision of the methods. On the other hand, the coverage of the patterns is limited by the highly variable syntactic structures, which affects the recall.

Machine learning technique is another option, as definition extraction can be modeled as a binary classification problem, and hypernym extraction can be modeled as a sequence labeling classification problem. However, there are several drawbacks of the previous machine learning based methods. For example, the two tasks are separately solved as they are modeled as different classification problems. Thus the correlation between them is not well employed.

This research was partially funded by ARC DPs 170103710 and 180103411, and D2DCRC DC25002 and DC25003.

G. Li et al. (Eds.): DASFAA 2019, LNCS 11448, pp. 415–419, 2019.
https://doi.org/10.1007/978-3-030-18590-9_57

In this paper, we propose a novel machine learning framework to extract definitions and hypernyms *simultaneously*. Our framework contains two phases. In phase I, we employ a joint neural network model to predict (a) whether the sentence is definitional, and (b) the best k label sequences for the sentence. In phase II, we train a refinement model to improve the prediction quality of hypernyms in phase I. Unlike most existing machine learning methods, in our framework, the features are effective but easy to obtain.

We demonstrate the effectiveness of the proposed framework by experimenting it on a well-known benchmark of textual definition and hypernym extraction [9]. We show that our proposed framework substantially improves the performance for both tasks, leading to the new state-of-the-art.

2 Proposed Two-Phase Framework

2.1 Problem Definition

Given a sentence, our objectives are (1) classifying the sentence as definitional (labeled as `True`) or non-definitional (labeled as `False`), and (2) labeling each word in the sentence as at the beginning of (e.g., `B-HYP`), inside of (e.g., `I-HYP`), or outside (e.g., `O`) a hypernym. In this paper, we propose to solve the problem with a supervised learning framework. A set of sentences, their labels, and annotated labels for each word in the sentences are given as the training data.

2.2 Phase I: A Joint Neural Network Model

Figure 1 shows the architecture of the neural network in phase I. There are mainly four parts in the neural network.

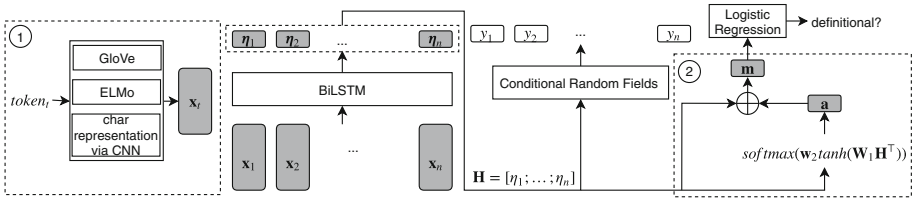

Fig. 1. The architecture of the neural network in phase I

As the first step in the neural network, we take the sequence of tokens from the sentence as input, and for each token, we generate its representation (e.g., the dashed box 1 in Fig. 1). The representation includes the character level representation which is generated by a CNN, the GloVe word embedding, and the ELMo word representation.

The representations for the tokens (e.g., $\mathbf{x}_i$) are then fed into the BiLSTM layer. We use LSTMs instead of Recurrent neural networks (RNNs) to overcome

the gradient vanishing/exploding issue and capture long-distance dependencies. We use the bi-directional LSTM (BiLSTM) to access to both the left and right contexts for each token, which leads to a better performance. The output of the BiLSTM layer will be used for both of the following two parts.

As shown in dashed box 2 in Fig. 1, we utilize a self-attention mechanism [7] to encode a variable length sentence into a fixed size embedding (e.g., an embedding for the sentence). The embedding $\mathbf{m}$ is generated by a linear combination of the BiLSTM output vectors. $\mathbf{m}$ will be used as the input of a multilayer perceptron (MLP) to determine whether the sentence is definitional or not.

We use CRF [5] to predict the hypernyms as it is known to be one of the most effective solutions for sequence labeling tasks. The input of the CRF in our framework is the output vector sequence of the BiLSTM layer, and the output of the CRF is a label sequence with length n.

The loss function of the neural network (e.g., $\mathcal{L}_1$) is a combination of the loss for both tasks:

$$\mathcal{L}_1 = \mathcal{L}_{def} + \mathcal{L}_{hyp} = -\sum_i y \log p(y \mid \mathbf{m}) - \sum_i \log(p(\mathbf{y} \mid \eta)).$$

2.3 Phase II: Refinement Model

We observe that in phase I, the performance on labeling hypernyms is relatively low. However, the true hypernyms usually can be labeled correctly in at least one of the k (e.g., $k = 5$) label sequences with highest scores. This motivates us to use another classifier to refine the result of the best k label sequences of the CRF layer.

We use XGBoost [3] to do the multiclass classification, with softmax to calculate the probabilities. Given token x, the probability of all the possible labels (i.e., `O`, `B-HYP`, and `I-HYP`) are computed using the softmax function and stored in the vector $\mathbf{s}$:

$$\mathbf{s} = softmax(\mathbf{W}^\top f(x)),$$

where $f(x)$ is the feature vector of token x. $\mathbf{W}$ is the weight matrix.

We use cross-entropy with regularization as the loss function:

$$\mathcal{L}_2 = -\sum_i y \log p(y \mid x) + \lambda \|\mathbf{W}\|_2^2 .$$

In order to achieve better performance, in addition to best k labels, we also utilize the similarity to hyponym, POS tag, and dependency parsing information as features in phase II.

We present the detail about the proposed framework in the full version of this paper [10].

3 Experiments

We evaluate our model on a public benchmark of definition extraction and hypernym extraction [9]. The benchmark contains 4,619 sentences (1,908 of them are annotated as *definitional*), 1,908 hyponyms and 2,046 hypernyms.

We evaluate the performance of different models by comparing the predicted results on the test set using *Precision* (P), *Recall* (R), and F_1 score. We also report *Accuracy* (Acc) for definition extraction. Following the previous work [8], hypernyms are evaluated in substring level. All the results are averaged over 10-fold cross validation. The ratio of training samples, develop samples and test samples is 8:1:1. All the experiments are performed on Intel Xenon Xeon(R) CPU E5-2640 (v4) with 256 GB main memory and Nvidia 1080Ti GPU.

Table 1. Evaluation results

Method	Definition extraction				Hypernym extraction		
	P	R	F1	Acc	P	R	F1
WCL-3 [8]	**98.8**	60.7	75.2	83.5	78.6	60.7	68.6
Boella and Di Caro [2]	88.0	76.0	81.6	89.6	83.1	68.6	75.2
Li et al. [6]	90.4	92.0	91.2	–	–	–	–
Espinosa-Anke et al. [1]	–	–	–	–	**84.0**	76.1	79.9
Proposed framework	96.8	**96.5**	**96.6**	**97.3**	83.8	**83.4**	**83.5**

Table 1 concludes the results for both tasks. Our framework significantly outperforms the other methods by at least 5.4 in F_1 score for definition extraction task and at least 3.6 F_1 score for hypernym extraction task. For detailed analysis and more experiment results please refer to the full version of this paper [10].

4 Conclusion

In this paper, we propose a two-phase framework to tackle definition extraction and hypernym extraction tasks simultaneously. A joint neural network is used to predict for both tasks, with the performance further enhanced by a refinement model. The experiment shows the effectiveness of our framework.

References

1. Espinosa-Anke, L., Ronzano, F., Saggion, H.: Hypernym extraction: combining machine-learning and dependency grammar. In: Gelbukh, A. (ed.) CICLing 2015. LNCS, vol. 9041, pp. 372–383. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18111-0_28
2. Boella, G., Di Caro, L.: Extracting definitions and hypernym relations relying on syntactic dependencies and support vector machines. In: ACL, pp. 532–537 (2013)

3. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: KDD, pp. 785–794. ACM (2016)
4. Flati, T., Vannella, D., Pasini, T., Navigli, R.: Two is bigger (and better) than one: the Wikipedia bitaxonomy project. In: ACL, vol. 1, pp. 945–955. ACL (2014)
5. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML (2001)
6. Li, S.L., Xu, B., Chung, T.L.: Definition extraction with LSTM recurrent neural networks. In: Sun, M., Huang, X., Lin, H., Liu, Z., Liu, Y. (eds.) CCL/NLP-NABD -2016. LNCS (LNAI), vol. 10035, pp. 177–189. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47674-2_16
7. Lin, Z., Feng, M., dos Santos, C.N., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. CoRR abs/1703.03130 (2017)
8. Navigli, R., Velardi, P.: Learning word-class lattices for definition and hypernym extraction. In: ACL, pp. 1318–1327. ACL (2010)
9. Navigli, R., Velardi, P., Ruiz-Martínez, J.M.: An annotated dataset for extracting definitions and hypernyms from the web. In: LREC (2010)
10. Sun, Y., Liu, S., Wang, Y., Wang, W.: Extracting definitions and hypernyms with a two-phase framework. arXiv e-prints, January 2019