Analysis of Distant Supervision for Relation Extraction Dataset

Kijong Han[1], Sangha Nam[1], YoungGyun Hahm[1], Jiseong Kim[1], Jiho Kim[1],
Jin-Dong Kim[1,2], Key-Sun Choi[1]

[1] Korea Advanced Institute of Science and Technology, South Korea
[2] Database Center for Life Science, Japan
{han0ah, nam.sangha, hahmyg, jiseong, hogajiho}@kaist.ac.kr,
jdkim@dbcls.rois.ac.jp, kschoi@kaist.ac.kr

Abstract. Deep learning techniques have been applied to relation extraction task, and demonstrated remarkable performances. However, the results of these approaches are difficult to interpret and are sometimes counter-intuitive. In this paper, we analyze the ontological and linguistic features of a relation extraction dataset and the pros and cons of existing methods for each feature type. This analysis result could help design an improved method for relation extraction by providing more insights into the dataset and models.

Keywords: information extraction, relation extraction, dataset analysis

1 Introduction

Relation Extraction (RE) is to extract semantic triples consisting of entity pairs and relation between the entity pairs from non-structured natural language text. Supervised learning approaches for RE require a large amount of labelled training data, which requires considerable human effort. To address this problem, a distant supervision (DS) method [4] is widely used these days. Despite its usefulness, there is a problem in DS for RE. The distant supervision method automatically generates labelled data, so there are wrongly labelled data which can cause noise.

Statistical machine learning and deep learning have been applied to solve these problems, and have demonstrated a remarkable performance improvement [4,7]. However, the results of these approaches are difficult to interpret and are sometimes counter-intuitive.

Thus, to interpret the results of existing RE methods, we analyze ontological and linguistic features of a RE dataset and the pros and cons of existing methods for each feature type. This analysis result provides more insights into the datasets and characteristics of existing RE methods, so it could help design an improved RE method. A convolutional neural network (CNN)-based method [7], and a Markov logic network (MLN)-based method [1] are selected for analysis. Our implementations are available at http://github.com/machinereading/re-cnn for CNN, and http://github.com/machinereading/re-mln for MLN.

2 Background

We are inspired by the study that constructs and analyze the dataset for recognizing textual entailment (RTE) task [3]. This study categorized an RTE dataset according to linguistic phenomenon, and analyzed it by applying a previous RTE methods. We conducted a type of analysis suitable for a RE task.

We selected two existing RE methods for analysis. One is a CNN-based method [7], the other is an MLN-based method [1]. We select CNN as a representative of deep learning-based methods, and MLN as a representative of methods that utilize logic rules and hand-crafted features. MLN is a model that combines a Markov random field and weighted logic rules [6]. It represents information as first-order logic predicates and formulas. (e.g. $HasFea(D_i, write) \Rightarrow Label(D_i, author)$. If data D_i has the feature word 'write' then the relation label of D_i is author). Each formula has a weight that represents the confidence, and the weight is trained statistically from the dataset. To utilize this weight, this model can calculate the probability that ground predicate is true. This model has logic rules with weights, and this information is very useful for analyzing the dataset.

3 Analysis Setup

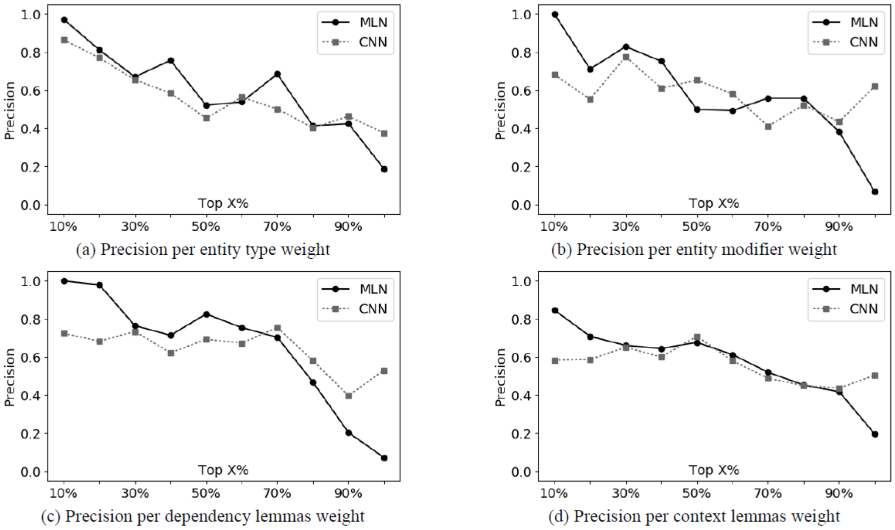
3.1 Dataset

We used the Korean DS for RE dataset [5], which was constructed from the Korean Wikipedia (2017. 07) sentences and K-box triples. K-Box is a knowledge base extended from the Korean DBpedia. We randomly sampled these datasets. A total of 13,489 DS training data instances, 4,096 gold test data instances, and the top 30 most frequent relations were used for this study. Gold test data was constructed by the process of removing wrongly labelled data from DS by 14 part-time students hired by our research team. In this paper, we refer to one sentence having a designated entity pair as ‘a data’or ‘a data instance’.

3.2 Method

First, we analyzed the overall performance of the two methods. Second, we selected and classified four features by analyzing the data manually, and we investigated how important each feature type was for the prediction of a data relation. These features are also used as MLN features. The features are as follows:

(1) Entity type: Fine-grained entity type defined by a K-box, which are originated from DBpedia ontology classes.(e.g. An entity type of the Lionel Messi is *Athelete*.) **(2) Entity modifier**: A modifier is a sentence component modifying another component. For example, in the sentence ‘John who is the author of ...’, ‘author’is the clausal modifier of the entity ‘John’. **(3) Lemmas in a dependency path**: Lemmas in a dependency path between entity pairs is an important feature. Many previous studies also leveraged this feature [1,4]. **(4) Context lemmas**: Context lemmas are lemmas of words that not dependencies or modifiers in sentences.



(a) Precision per entity type weight

(b) Precision per entity modifier weight

(c) Precision per dependency lemmas weight

(d) Precision per context lemmas weight

Fig. 1. Precision per weight of each feature type

4 Analysis Results

Overall Performance. The best F1-score was 0.616 (CNN), 0.611 (MLN), and the accuracy was 0.584 (CNN), 0.594 (MLN). The F1-score was measured in terms of how accurately the method extracts triples, as calculated by other studies [1,4]. Accuracy was measured considering only the best prediction for the data instance. Both models showed similar performance overall.

Importance of each Feature Type. We investigated the importance of each feature type by measuring the precision per weight of the formulas in MLN. In the MLN method, each prediction of a relation label for each data has a list of weighted formulas affecting the prediction as described in Section 2. The higher the weight, the more important the feature in the formula based on the training data statistics. Each graph in Figure 1 is drawn considering only the weight of a specific feature type in the calculation. In the X axis, the X% point represents the portion of data that has top (X-10%,X%) weight for a specific feature type. The Y axis represents the accuracy for that portion of the data. Precision was measured by considering only the best prediction for the data instance. For all graphs in Figure 1, the MLN curve shows a lower performance than CNN for a range with a low weight, and higher performance than CNN for a range with a high weight. The MLN curve shows a stronger weight correlation than the CNN curve. Thus, all four features are meaningful to some degree. The MLN curve in entity type (a), entity modifier (b), and dependency lemmas (c) graph shows close to a 1.0 precision for the top 0-20% highest weight dataset. This means that these three features are crucial for specific RE sentences.

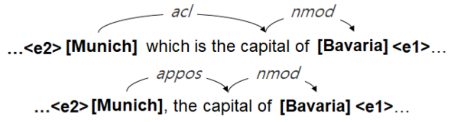

acl
nmod
...<e2> [Munich] which is the capital of [Bavaria] <e1>...
appos
nmod
...<e2>[Munich], the capital of [Bavaria] <e1>...

Fig. 2. Examples of a simple N of N pattern

Method	N-of-N pattern	All data
MLN	0.718	0.594
CNN	0.493	0.584

Fig. 3. Accuracy per data type

Simple N of N Pattern. We found a data pattern that is very intuitive for determining a relation, but does not work well with CNN. This pattern simply consists of entity1 which is part of an 'N of N'phrase, and entity2 modified by an 'N of N'phrase. Examples are shown in Figure 2. There is a total of 71 data instances of this pattern. In this pattern, the clue word (e.g. 'capital'in Figure 2) is strong evidence for inferring a relation. MLN also utilizes this clue word as strong evidence, because this word acts as both an entity modifier and dependency lemma feature. Thus, MLN shows a higher performance for this pattern than its overall performance as shown in Figure 3.

5 Conclusion

We analyzed the ontological and linguistic features of RE datasets, as well as the pros and cons of existing methods for each feature type. We expect that these insights into the RE dataset analyzed in this study could help design an improved RE method. For example, we can use important feature(e.g. entity type) as an additional discrete feature vector for input, or combine high precision rule derived from the pattern(e.g. simple N-of-N pattern in Section 4.) into the neural net architecture by utilizing the model such as [2].

Acknowledgement. This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (2013-0-00109, WiseKB: Big data based self-evolving knowledge base and reasoning platform)

References

1. Han, X., Sun, L.: Global distant supervision for relation extraction. In: AAAI (2016)
2. Hu, Z., Ma, X., Liu, Z., Hovy, E., Xing, E.: Harnessing deep neural networks with logic rules. In: ACL (2016)
3. Kaneko, K., Miyao, Y., Bekki, D.: Building japanese textual entailment specialized data sets for inference of basic sentence relations. In: ACL (2013)
4. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: ACL-IJCNLP (2009)
5. Nam, S., Han, K., Kim, E.k., Choi, K.S.: Distant supervision for relation extraction with multi-sense word embedding. In: GWC workshop on Wordnets and Word Embeddings (2018)
6. Richardson, M., Domingos, P.: Markov logic networks. Machine learning 62 (2006)
7. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: COLING (2014)