Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2022.030794
Article

Tech Science Press

Employing Lexicalized Dependency Paths for Active Learning of Relation Extraction

Huiyu Sun* **and Ralph Grishman**

New York University, New York, 10012, USA
*Corresponding Author: Huiyu Sun. Email: hs2879@nyu.edu
Received: 26 February 2022; Accepted: 20 April 2022

Abstract: Active learning methods which present selected examples from the *corpus* for annotation provide more efficient learning of supervised relation extraction models, but they leave the developer in the unenviable role of a passive informant. To restore the developer's proper role as a partner with the system, we must give the developer an ability to inspect the extraction model during development. We propose to make this possible through a representation based on lexicalized dependency paths (LDPs) coupled with an active learner for LDPs. We apply LDPs to both simulated and real active learning with ACE as evaluation and a year's newswire for training and show that simulated active learning greatly reduces annotation cost while maintaining similar performance level of supervised learning, while real active learning yields comparable performance to the state-of-the-art using a small number of annotations.

Keywords: Relation extraction; simulated active learning; real active learning; rule-based learning; lexicalized dependency paths

1 Introduction

Relation extraction has seen a lot of development in recent years, with supervised learning methods using neural network models making the most progress ([1–16]). This, however, requires lots of training data, which limits the portability to new relations and domains.

Active learning methods which present selected examples from the *corpus* to the user for acceptance or rejection (as instances of the relation) provide more efficient learning of these models ([17–24]). However, they can be frustrating for developers because they do not afford any insight into or direct control of the model under development; the developer becomes a passive informant. This makes for good papers but not for useable systems.

This has led in practice to the continued use of manually-curated rule-based systems, which are more inspectable and thus give the developer more control [25]. Our goal is to find a rule-based representation that is transparent, intuitive and efficient, so that rules can be easily understood and a small set of rules yields satisfactory performance.

We utilize lexicalized dependency paths as the representation for our active learning system. Lexicalized dependency path (LDP) was proposed in [26] and was shown to be a good fit for supervised relation extraction while being more efficient and transparent than previous models. This makes active learning of LDPs feasible. This approach, a type of active learning for relation extractors, is being implemented in ICE [27].

2 Related Work

Reference [28] presented the Snowball system for bootstrapping relations from large text corpora, presenting strategies for generating and evaluating extraction patterns. Reference [19] presented an active learning model for pipelines including segmentation, entity recognition and relation extraction. Reference [20] explored multi-task Active learning adding various constraints on the output. Reference [29] combined semi-supervised learning with active learning for extraction in the biomedical domain. Reference [30] used word clustering for semi-supervised extraction of relations. Reference [21] presented an active learning system for relation extraction looking at both local and global views of relation instances. Reference [31] used similarity of tree-kernels for domain adaptation of relation extraction. Reference [22] presented an efficient active learning system that observed a big gain for small numbers of annotations. They used a good balance of positive and negative samples and boosts efficiency by interleaving self-training and co-testing. Reference [32] used partial and distant supervision for selecting uncertain and informative examples. Reference [24] presented an active learning system for extracting rules from log-linear and neural network models. References [33,34] presented active learning systems in entity recognition for aiding relation extraction.

3 Active Learning

3.1 Lexicalized Dependency Path

LDP [26] is based on the dependency paths between two entities in a dependency tree and it specifies a path between two entities (nodes) through a dependency tree; it has the form:

$$arg1type - deptype : lex : deptype : lex : \ldots : deptype - arg2type.$$

The *arg1type* and *arg2type* are entity types of the two arguments of a relation (e.g., PERSON, GPE), *deptype* specifies a type of dependency relation (e.g., *nsubj*, *dobj*), and *lex* is the base form of a lexical item (e.g., *work*, *from*). If an arc of type *X* in the dependency tree is traversed from governor to dependent, the path includes *'X'*; if the arc is traversed from dependent to governor, the path includes *'X-1'*, e.g., *nsubj-1* or *dobj-1*.

Extending dependency path this way has several advantages over predecessors. First, it creates a rule-based system that is unified and easily understood and it allows a nature extension to other forms of learning such as active learning. Second, this representation also means we can dissect a sentence into a number of LDPs allowing for a supervised trainer to be used.

To combine coarse and fine semantic classes, we differentiate LDPs formed between pairs of entity type arguments and all of its subtype arguments. Given a sentence with a pair of entity mentions, we can proceed in three ways: we can create a type LDP from the types of the entity mentions; we can create a subtype LDP from the subtypes of the entity mentions; or we can create both type and subtype LDPs. In the last case, a test LDP may have two matches against the model: one matching types, one matching subtypes. In that case the subtype match takes precedence. This makes it possible to have general rules and more specific exceptions. Supervised learning of LDPs is shown to be effective in [26] and we describe the extension to active learning next.

3.2 Setup

Active learning (AL) involves a tight coupling of developer and machine. At each cycle of the learning process, the machine selects from an unlabeled training set one or more items and the developer labels them. The selection criteria vary, but basically try to maximize the expected value of the improvement of the accuracy of the model being trained. In our case the items are LDPs and the labels are the relation types or "no relation".

We perform both simulated and real active learning. Real active learning involves a real developer as an integral part of the customization process, an expensive process but an essential one to ensure that we are not overlooking some potential problem with the learning process. In simulated active learning, we simulate the judgment of the developer by consulting an annotated training *corpus*. This makes it possible to make many runs with different parameters at reasonable cost, but is limited by the availability of annotated corpora.

Experiments in active learning require a small set of seeds, a large unlabeled *corpus* for training (the 'source *corpus*'), and a small labeled *corpus* for evaluation (the 'target *corpus*'). In real active learning, ACE [35] is used as the target domain and we used the Los Angeles Times from 2006 to serve as the source domain. This *corpus*, part of the Linguistic Data Consortium's Gigaword collection, has approximately 41,000 documents with 30 million words. Approximately 800,000 LDPs are generated.

ICE [27] converts the seed phrases to LDPs and uses a shared argument pairs metric to identify the LDPs most similar to the seeds, which are returned as a ranked list to the user. A shared argument pairs metric counts the number of pairs of arguments which two LDPs have in common. This is a simple selection strategy based only on positive evidence, but relying on positive evidence has proven effective in active learning of extraction engines when there are few labeled examples [23].

Apart from the shared argument ranker, we also tried the maximum entropy and smallest margin rankers. In terms of entropy, given unlabeled example x and output class Y, the uncertainty can be calculated via the following:

$$S(x) = \sum_{y \in Y}^{n} P(y|x) log P(y|x)$$

The smallest margin can be calculated as follows given the two most likely classes y_1, y_2:

$$S(x) = |P(y_1|x) - P(y_2|x)|$$

The user can then accept some of these LDPs and reject others. This process is repeated for several iterations. The selected LDPs are then scored on the target domain. Simulated active learning splits the ACE *corpus* the same way as supervised learning (bn+nw as training, and each of bc, cts, un and wl as test). The ACE training data is used to provide relation labels for a LDP, and the resulting LDPs are scored on the test.

4 Experimental Results

The ACE 2005 *corpus* [35] is used to evaluate our active learning model. ACE has approximately 600 documents with a total of 300,000 words, annotated for entities, relations, and events. 6 relation classes are annotated: ORG-AFF (Organization-Affiliation), PART-WHOLE, PHYS (Physical), GEN-AFF (General-Affiliation), PER-SOC (Person-Social), and ART (Artifact). We also use the NR class to indicate 'no relation'. 7 entity types are annotated in ACE 2005: PERSON, GPE (Geo-Political Entity), ORGANIZATION, LOCATION, FACILITY, WEA (Weapon), and VEH (Vehicle).

4.1 Simulated Active Learning

The source domain of LA Times contains over 800 k LDPs to sample from, and a portion of selected LDPs won't have a match in ACE, therefore we choose a relatively large batch size of 100. 90% of ACE is used as the oracle, and the remaining 10% as the target domain for active learning (AL), where we picked a train/test split on ACE that produced a F1 close to the supervised mean of 59.14% after 20 runs to reduce the random variations. We use several starting seeds for each relation class, some of which shared with other classes (e.g., the seeds 'Person of GPE' and 'GPE Person' are used for ORG-AFF, GEN-AFF and PHYS).

We ran experiments using type only LDPs first and then added subtypes. The AL curves of using type only LDPs and type + subtype LDPs along with a direct comparison of F1 between the two are shown in Fig. 1. Using the seeds, we get a F1 of 33.66% for type LDPs and 34.21% for type + subtype. Type + subtype clearly outperforms type LDPs, beating it by over 3% after 10 iterations. This is the result of smaller precision drops at the later iterations and a steady increase in recall across all iterations. In the initial iterations, a much bigger F1 increase is observed for both graphs, where for type + subtype the F1 increase in the first 2 iterations is similar to the increase for the latter 8 iterations combined, showing active learning using LDPs is able to achieve good performance using only a small number of labels.

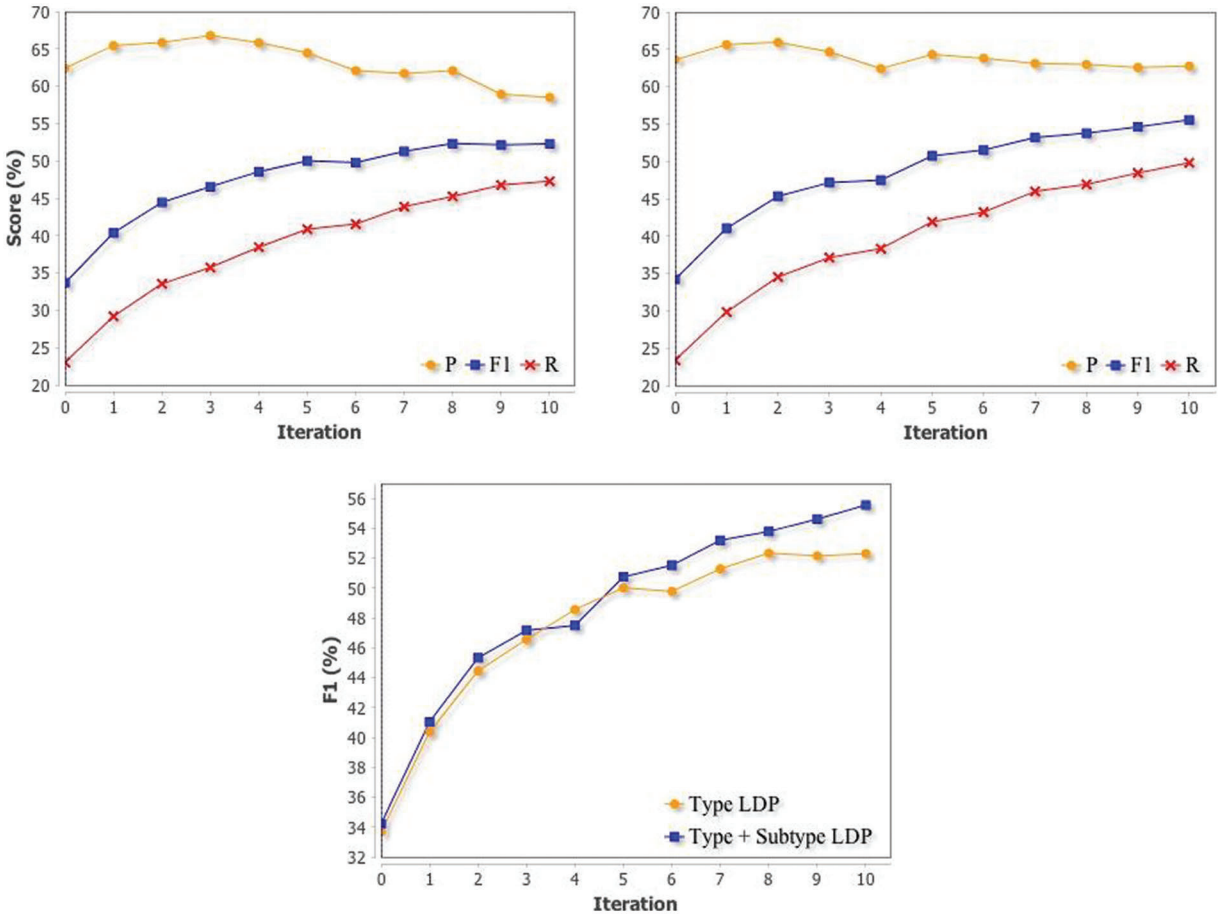

Figure 1: Simulated active learning performances for type LDP (*top left*) and type + subtype LDP (*top right*) on all relation classes. Direct comparison of F1 scores is also shown (*bottom*)

In terms of labeling reduction, after 10 iterations, the LDPs selected from ICE matched a total of 439 LDPs from ACE (274 type and 165 subtype). The training split of ACE contains a total of 5782 LDPs (2524 type and 3258 subtype). Therefore, for type LDP, we only labeled 10.86% of the training LDPs and is able to achieve 91.01% of supervised scores, reducing labeling effort on the new domain by 89.1%. For type + subtype LDPs, we used 7.59% of training LDPs and achieved 93.95% of supervised scores, reducing labeling effort on the new domain by 92.4%.

4.2 Real Active Learning

The source domain for real active learning (AL) is the same as simulated AL but the target domain is now the entire ACE *corpus*. For LDPs selected by ICE, we ask the developer to provide relation labels. In the case of relation label conflicts, the developer is asked to resolve it and select just one label. We ran experiments on both type and type + subtype LDPs; scores for 10 iterations are shown in Fig. 2. Several seeds for each relation class are used and iterations have a batch size of 100. Using the seeds, type and type + subtype LDPs respectively gets a F1 of 34.38% and 34.98% on ACE, and after 10 iterations, the F1 are respectively 53.02% and 57.55%. Type + subtype still outperforms type LDPs by a large margin (4.5%), consistent with results from supervised training and simulated AL, which further shows the advantage of combining fine and coarse semantic classes.

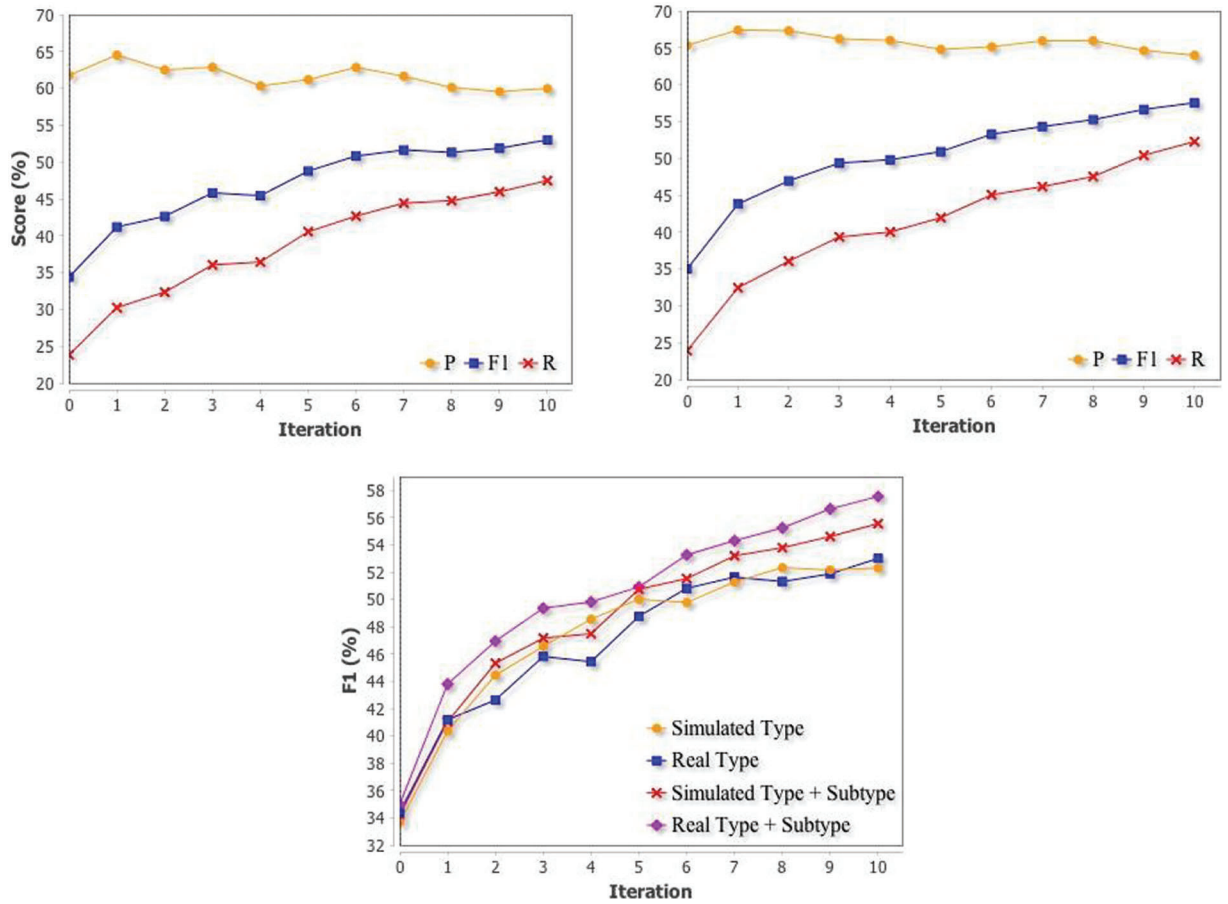

Figure 2: Real Active learning performances for type LDP (*top left*) and type + subtype LDP (*top right*) on all relation classes. Direct comparisons of F1 scores for simulated and real AL are also shown (*bottom*)

Comparing simulated AL with real AL in Fig. 2, it shows that real AL outperforms simulated AL for both type and type + subtype LDPs (2% for type + subtype). This might seem surprising at first as real AL introduces human labeling error. But the result is also expected because real AL doesn't rely on the training split of ACE to provide labels for selected examples: it has an unlabeled pool of over 800 k LDPs to choose from. If the ranking of unlabeled LDPs on the source domain is further optimized, real AL could potentially outperform supervised LDP scores. Considering human error in labeling and relation classification left to the developer, the increase in performance of real AL against simulated AL shows that by giving the developer the ability to inspect the extraction model during development, as we have using the LDP representation, we are able to achieve a better learning performance.

We compare our active learning results to other state-of-the-arts methods in Tab. 1. Precision (P), Recall (R), and F1-score (F) are shown. The F1-score is the standard metric for comparing the overall performance of models. They can be calculated as follows:

$$P = \frac{|\{relevat\ relations\} \cap \{retrieved\ relations\}|}{|\{retrieved\ relations\}|}$$

$$R = \frac{|\{relevat\ relations\} \cap \{retrieved\ relations\}|}{|\{relevant\ relations\}|}$$

$$F = \frac{2PR}{P+R}$$

Table 1: Active learning performance of our system using type + subtype LDPs on ACE 2005 after 10 iterations compared with state-of-the-arts on ACE 2004 using a similar number of labeled instances

Relation	Type + Subtype LDP			B + N + S + G	LGCo-Testing
	P	R	F	F	F
ORG-AFF	74.25	57.93	65.08	**71.52**	66.20
GEN-AFF	55.74	46.17	**50.51**	43.01	44.50
PART-WHOLE	62.12	50.78	55.88	N/A	N/A
PHYS	52.49	42.61	**47.04**	41.16	39.90
PER-SOC	71.32	66.18	68.65	68.87	**70.80**
ART	73.26	46.82	57.13	43.33	**65.40**
ALL	63.98	52.29	**57.55**	52.98	N/A

In terms of other methods that we compare with in Tab. 1, [22] randomly selected 80% of ACE 2004 as source domain for sampling and the rest 20% as target domain for testing. Reference [21] also uses ACE 2004 as both the source and target domain. Our source domain is approximately 100 times larger than ACE 2004 and 2005, therefore in order to directly compare our number of labeled instances to theirs, we use the number of instances from the source that are also in ACE 2005: after 10 iterations, the LDPs selected from the source domain matched 206 LDPs from ACE 2005 so the performance using the closest number of labels, 150 for [22] and 200 for [21], are compared. Relation types are also different between ACE 2004 and 2005 and for direct comparison, EMP-ORG relation of ACE 2004 is compared with ORG-AFF of ACE 2005, GPE-AFF of 2004 is compared with GEN-AFF of 2005, and

PART-WHOLE of 2005 is not compared as the relation class did not exist before. Results of our type + subtype LDP model with the best systems B+N+S+G of [22] after 150 instances and LGCo-Testing of [21] after 200 labels are shown in Tab. 1. Overall, our real active learning system using type + subtype LDPs outperforms theirs despite having to adapt from a vastly different source, which further shows the advantage of having an inspectable development model.

4.3 Discussion

Our LDP model is inspectable and so can provide insights to understand system failures and errors much more easily than feature-based and neural network based models. The supervised LDP model on average misses about 40% of ACE relations. Many of these can be recovered by a kNN classifier or soft match in place of exact match for LDPs and readily explained.

Some errors reflect choices made regarding the dependency structure. For example, some parsers generate rather flat noun groups, thus missing some 'true' dependency relations. The noun phrase 'GM chief Rick…' has the LDP ORGANIZATION–nn-1–PERSON which doesn't contain the crucial word chief on its path that can easily identify the ORG-AFF relation. These prenominal modifiers that serve to identify the relation class would not be on the path. Another example is 'William, brother Harry…' which has the LDP PERSON–appos–PERSON and doesn't contain the word brother which identifies a PER-SOC relation. Reference [36] addressed this problem by introducing 'syntactico-semantic structures' for analyzing relations within noun groups; we aim to adopt a similar strategy in future works.

5 Future Work

We aim to improve the ranking of unlabeled LDPs. In addition to the shared argument ranker used in these experiments, we have implemented a ranker based on word embeddings and plan to try rankers which more directly capture uncertainty. We plan to test on additional entity and relation types, such as SemEval, and to provide a more general hierarchy extending the type-subtype pairs of ACE.

6 Conclusion

We proposed an active learner utilizing lexicalized dependency path (LDP) as the representation to provide the developer with more control over the extraction model. We applied our LDP model to simulated and real active learning and found that combining fine and coarse semantic classes in our use of type + subtype LDPs achieved F1 improvement of 4.5% over just coarse classes. Simulated active learning experiments using LDPs reduced the labeling effort on ACE by over 92% and maintained close to supervised scores. Real active learning experiments of LDPs outperformed simulated by 2% due to having a much bigger pool of unlabeled examples to select from and achieved a better overall performance to the state-of-the-art, showing the advantage of collaborative involvement of the developer in the extraction model.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Tian, G. Chen, Y. Song and X. Wan, "Dependency-driven relation extraction with attentive graph convolutional networks," in *Proc. ACL*, Bangkok, Thailand, pp. 4458–4471, 2021.

[2] Z. Hu, Y. Cao, L. Huang and T. S. Chua, "How knowledge graph and attention help? A quantitative analysis into bag-level relation extraction," in *Proc. ACL*, Bangkok, Thailand, pp. 4662–4671, 2021.

[3] X. Hu, C. Zhang, Y. Yang, X. Li, L. Lin *et al.,* "Gradient imitation reinforcement learning for low resource relation extraction," in *Proc. EMNLP*, Punta Cana, Dominican Republic, pp. 2737–2746, 2021.

[4] H. Wu and X. Shi, "Synchronous dual network with cross-type attention for joint entity and relation extraction," in *Proc. EMNLP*, Punta Cana, Dominican Republic, pp. 2769–2779, 2021.

[5] A. Veyseh, F. Dernoncourt, D. Dou and T. Nguyen, "Exploiting the syntax-model consistency for neural relation extraction," in *Proc. ACL*, Seattle, Washington, USA, pp. 8021–8032, 2020.

[6] I. Beltagy, K. Lo and W. Ammar, "Combining distant and direct supervision for neural relation extraction," in *Proc. NAACL*, Minneapolis, Minnesota, pp. 1858–1867, 2019.

[7] N. Zhang, S. Deng, Z. Sun, X. Chen, W. Zhang *et al.,* "Attention-based capsule networks with dynamic routing for relation extraction," in *Proc. EMNLP*, Brussels, Belgium, pp. 986–992, 2018.

[8] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou *et al.,* "A dependency-based neural network for relation classification," in *Proc. ACL-IJCNLP*, Beijing, China, pp. 285–290, 2015.

[9] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proc. 1st Workshop on Vector Space Modeling for Natural Language Processing at NAACL*, Denver, Colorado, USA, pp. 39–48, 2015.

[10] X. Lin, T. Liu, W. Jia and Z. Gong, "Distantly supervised relation extraction using multi-layer revision network and confidence-based multi-instance learning," in *Proc. EMNLP*, Punta Cana, Dominican Republic, pp. 165–174, 2021.

[11] T. H. Nguyen, B. Plank and R. Grishman, "Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation extraction," in *Proc. ACL-IJCNLP*, Beijing, China, pp. 635–644, 2015.

[12] C. D. Santos, B. Xiang and B. Zhou, "Classifying relations by ranking with convolutional neural networks," in *Proc. ACL-IJCNLP*, Beijing, China, pp. 626–634, 2015.

[13] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng *et al.,* "Classifying relations via long short-term memory networks along shortest dependency paths," in *Proc. EMNLP*, Lisbon, Portugal, pp. 1785–1794, 2015.

[14] D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. COLING*, Dublin, Ireland, pp. 2335–2344, 2014.

[15] S. Zheng, Y. Hao, D. Lu, H. Bao, J. Xu *et al.,* "Joint entity and relation extraction based on a hybrid neural network," *Neurocomputing*, vol. 257, pp. 59–66, 2017.

[16] S. Vashishth *et al.*, "Reside: Improving distantly-supervised neural relation extraction using side information," in *Proc. EMNLP*, Brussels, Belgium, pp. 1257–1266, 2018.

[17] P. Radmard, Y. Fathullah and A. Lipani, "Subsequence based deep active learning for named entity recognition," in *Proc. ACL*, Bangkok, Thailand, pp. 4310–4321, 2021.

[18] Z. Jiang, Z. Gao, Y. Duan, Y. Kang, C. Sun *et al.,* "Camouflaged chinese spam content detection with semi-supervised generative active learning," in *Proc. ACL*, Seattle, Washington, USA, pp. 3080–3085, 2020.

[19] D. Roth and K. Small, "Active learning for pipeline models," in *Proc. AAAI*, Chicago, Illinois, USA, pp. 683–688, 2008.

[20] Y. Zhang, "Multi-Task active learning with output constraints," in *Proc. AAAI*, Atlanta, Georgia, USA, pp. 667–672, 2010.

[21] A. Sun and R. Grishman, "Active learning for relation type extension with local and global data views," in *Proc. CIKM*, Maui, HI, USA, pp. 1105–1112, 2012.

[22] L. Fu and R. Grishman, "An efficient active learning framework for new relation types," in *Proc. IJCNLP*, Nagoya, Japan, pp. 692–698, 2013.

[23] C. Cardellino, S. Villata, L. A. Alemany and E. Cabrio, "Information extraction with active learning: A case study in legal text," in *Proc. CICLing*, Cairo, Egypt, pp. 483–494, 2015.

[24] E. J. de Fortuny and D. Martens, "Active learning-based pedagogical rule extraction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2664–2677, 2015.

[25] L. Chiticariu, Y. Li and F. Reiss, "Rule-Based information extraction is dead! long live rule-based information extraction systems!," in *Proc. EMNLP*, Seattle, Washington, USA, pp. 827–832, 2013.

[26] H. Sun and R. Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science & Engineering*, vol. 43, no. 3, pp. 861–870, 2022.

[27] Y. He and R. Grishman, "ICE: Rapid Information extraction customization for NLP novices," in *Proc. NAACL*, Denver, Colorado, USA, pp. 31–35, 2015.

[28] E. Agichtein and L. Gravano, "Snowball: Extracting relations from large plain-text collections," in *Proc. 5th ACM Int. Conf. on Digital Libraries*, New York, NY, USA, pp. 85–94, 2000.

[29] M. Song, H. Yu and W. Han, "Combining active learning and semi-supervised learning techniques to extract protein interaction sentences," *BMC Bioinformatics*, vol. 12, no. S12, pp. 139, 2011.

[30] A. Sun, R. Grishman and S. Sekine, "Semi-supervised relation extraction with large-scale word clustering," in *Proc. ACL*, Portland, Oregon, USA, pp. 521–529, 2011.

[31] B. Plank and A. Moschitti, "Embedding semantic similarity in tree kernels for domain adaptation of relation extraction," in *Proc. ACL*, Sofia, Bulgaria, pp. 1498–1507, 2013.

[32] G. Angeli, J. Tibshirani, J. Wu and C. D. Manning, "Combining distant and partial supervision for relation extraction," in *Proc. EMNLP*, Doha, Qatar, pp. 1556–1567, 2014.

[33] H. Sun, R. Grishman and Y. Wang, "Domain adaptation with active learning for named entity recognition," *Lecture Notes in Computer Science (LNCS)*, vol. 10040, pp. 611–622, 2016.

[34] H. Sun, R. Grishman and Y. Wang, "Active learning based named entity recognition and its application in natural language coverless information hiding," *Journal of Internet Technology*, vol. 18, no. 2, pp. 443–451, 2017.

[35] C. Walker, S. Strassel, J. Medero and K. Maeda, "ACE 2005 multilingual training corpus," *Linguistic Data Consortium*, Web download: https://catalog.ldc.upenn.edu/LDC2006T06, 2006.

[36] Y. S. Chan and D. Roth, "Exploiting background knowledge for relation extraction," in *Proc. COLING*, Beijing, China, pp. 152–160, 2010.