Clustering-based Unsupervised Generative Relation Extraction

Chenhan Yuan
The University of Manchester
chenhan.yuan@postgrad.manchester.ac.uk

Ryan A. Rossi
Adobe Research
ryrossi@adobe.com

Andrew Katz
Virginia Tech
akatz4@vt.edu

Hoda Eldardiry
Virginia Tech
hdardiry@vt.edu

***Abstract*—Existing unsupervised relation extraction methods work by extracting sentence features and using these features as inputs to train a generative model. This model is then used to cluster similar relations. However, these methods do not consider correlations between sentences with the same entity pair during training, which can negatively impact model performance. To address this issue, we propose a Clustering-based Unsupervised generative Relation Extraction (CURE) framework that leverages an Encoder-Decoder architecture to train a relation extractor as the encoder. Given multiple sentences with the same entity pair as inputs, CURE is deployed by predicting the shortest path between entity pairs on the dependency graph of one of the sentences. After that, we extract the relation information using the encoder. Then, entity pairs that share the same relation are clustered based on their corresponding relation information. Each cluster is labeled based on the words in the shortest paths corresponding to the entity pairs in each cluster. Experimental results demonstrate the effectiveness of CURE compared to state-of-the-art models across all benchmark datasets.**

***Index Terms*—relation extraction, generative models, deep learning, knowledge graphs**

I. Introduction

Relation extraction has been deployed in many important AI tasks, such as search engines, recommender systems, and question answering [1]–[3]. Relation Extraction (RE) focuses on how to extract a relation given an entity pair in the sentence. This was initially explored in rule-based and supervised ways. However, supervised relation extraction methods require some prior knowledge about the text, such as marking the correct triplets in each sentence. This limits the use of supervised relation extraction. Lately, unsupervised and distant supervised learning approaches have been introduced to the Relation Extraction problem [4]–[8]. These approaches address the problem of a lack of labeled training text data. In the distant-supervised method, researchers assumed that if the same entity pair appeared in different sentences, these sentences might describe the same relation and are marked as the same relation as in the seed example [4], [7]–[9]. As to the unsupervised learning approaches, based on selected features, clustering techniques were used in some work to find similar concept pairs and relations. After that, different groups were assigned different labels which can be achieved by manually labeling or selecting common words [5], [6], [10], [11].

Nevertheless, using seed examples to expand the training dataset causes error propagation problems [12]. Unlike the distant-supervised learning-based approach, unsupervised relation extraction models do not consider the correlation between sentences with the same entity pair, which can negatively impact model performance. Meanwhile, predefined feature selections, such as trigger words [11] and keywords [13], may introduce biases and influence the final result of the models [14].

To alleviate the issues discussed above, we propose a novel unsupervised approach to train a generative model that can extract relation information accurately. Our model does not require labeling new data or pre-defining sentence features. Concretely, we first extract the shortest path of the entity pair in this graph. After that, we train an encoder and a decoder simultaneously, the decoder reconstructs the input of encoder, i.e., the shortest path. After training this model, a well-trained encoder, also known as relation extractor, is obtained to extract relation information. Subsequently, a cluster-based method is used to cluster entity pairs based on their relation information. Finally, we label each cluster automatically by analyzing attributes of words that appear in the shortest path, such that the label of each cluster is exactly the relation words. These attributes include word frequency and word vector distance.

The contributions of this paper are three-fold: First, we propose a Clustering-based Unsupervised generative Relation Extraction (CURE) framework for (1) relation extractor training and (2) triplets clustering. Both approaches outperform the state-of-the-art approaches on the relation extraction task. Second, we develop a novel method for automatically training a relation information extractor based on the shortest path prediction. This method does not require labeling text or pre-specifying sentence features. Finally, the proposed relation cluster labeling approach selects relation words based on word frequency and word vector distance, enabling a more accurate description of the relation.

II. Related Work

Hasegawa et al. first proposed the concept of the context of entity pairs, which can be deemed as extracted features from sentences. After that, they clustered different relations based on feature similarity and selected common words in the context of all entity pairs to describe each relation [5]. An extra unsupervised feature selection process was proposed to reduce the impact of noisy words in context [15].

Some works also considered unsupervised relation extraction as a probabilistic generation task. Latent Dirichlet Allocation

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

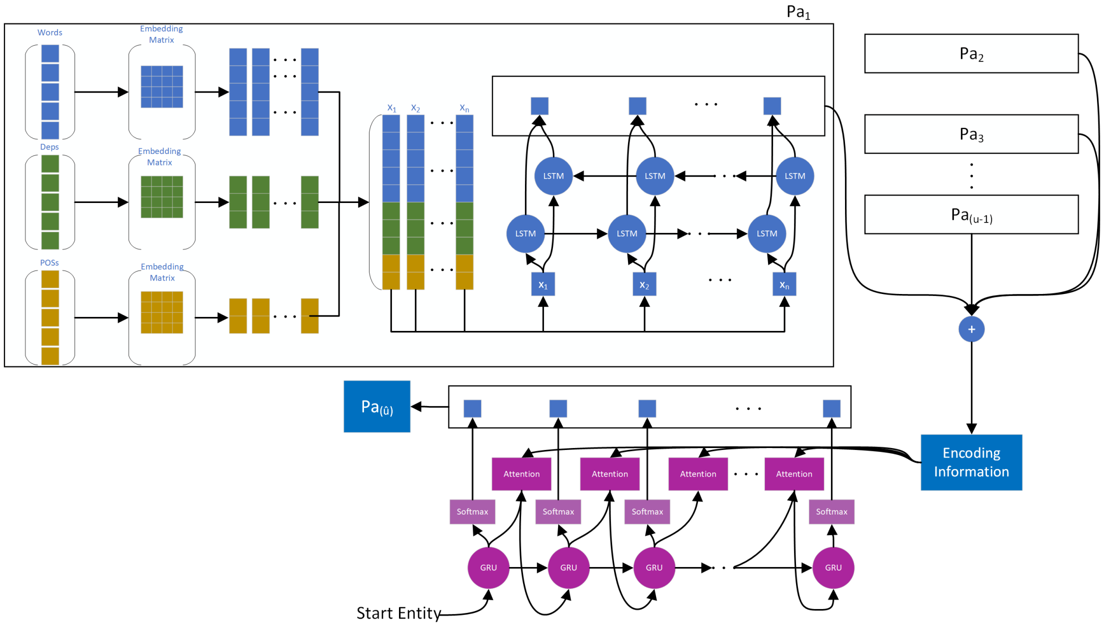

Fig. 1. The architecture of relation extractor training stage of CURE

(LDA) was applied in unsupervised relation extraction [11], [16]. Researchers replaced the topic distributions with triplets distributions and implemented Expectation Maximization algorithm to cluster similar relations. Marcheggiani et al. argued that previous generative models make too many independence assumptions about extracted features. As a variant of an autoencoder [17], they introduced a variational autoencoder (VAE) to a relation extraction model [18]. They first predicted semantic relation given entity pairs and reconstructed entities based on the prediction, respectively. Then they jointly trained the model to minimize error in entity recovery. In unsupervised open domain relation extraction [10], the authors used corresponding sentences of entity pairs as features and then vectorized the features to evaluate the similarity of relations.

TABLE I
AN EXAMPLE OF PATH SEARCH

Original Sentence: Ronald Reagan served as the 40th president of the United States	
Ent	(Ronald Reagan, the United States)
Dep	[‘nsubj’, ‘ROOT’, ‘prep’, ‘pobj’, ‘prep’, ‘pobj’]
POS	[‘PROPN’, ‘VERB’, ‘ADP’, ‘NOUN’, ‘ADP’, ‘PROPN’]
W	[‘Reagan’, ‘served’, ‘as’, ‘president’, ‘of’, ‘States’]

However, to the best of our knowledge, the correlation between sentences with the same entity pair has not been explicitly used to create a probabilistic generative relation extraction model. Multiple sentences with the same entity pair often occur in large-scale corpora, which can be used to let the relation extraction model learn how to extract features from sentences and convert them into relation information.

III. Framework

A. Model Overview

The proposed Clustering-based Unsupervised Generative Relation Extraction (CURE) model includes two stages. The first is the relation extractor training stage. We train a relation extraction model, which takes text and (e_i, e_j) as input and outputs vectorized relation representations. The second is the triplets clustering stage. The relation extractor model is used to extract relation representations then the relations are clustered. For a given sentence, the model then selects the closest centroid from cluster centroid set.

We begin by introducing the Encoder-Decoder model that is used to train the relation extractor. This proposed model captures the relation information given (e_i, e_j) and text. The model architecture is shown in Figure 1. This training model first encodes the semantic shortest paths of one entity pair in various sentences. The encoding information generated by the encoder reflects the relation information of the input (e_i, e_j). The decoder uses the summation of this information to generate the predicted semantic shortest path of that entity pair. More formally, our model optimizes the decoder ($\mathcal{D}$) and encoder ($\mathcal{E}$), s.t.

$$\operatorname*{argmax}_{\mathcal{D}_\theta,\mathcal{E}_\gamma} \mathbb{P}(Pa_u|Pa_1, Pa_2, \cdots, Pa_{u-1}) \tag{1}$$

where Pa_i is the i-th semantic shortest path of (e_i, e_j).

The formal definition of semantic shortest path is explained in section III-B. Here, we briefly explain why the task of this stage is to predict $\hat{Pa}_u$ given other semantic shortest paths. Note that it is necessary to build a well-trained encoder that can extract relation information from given semantic shortest paths.

In our scenario, since all the semantic shortest paths of one entity pair possibly share similar relation information, we treat one of them as the "correct expected result", and the remaining semantic shortest paths are provided as input to the encoder-decoder training model. This "correct expected result" will be generated as output by that model. This proposed semantic shortest path prediction approach provides a unsupervised mechanism to train the encoder-decoder model.

In the triplets clustering stage of CURE, the well-trained encoder is used as the relation extractor. The procedure of using the relation extractor model is shown in Fig. 2. This procedure first generates encoding information of input entity pairs (e_i, e_j) using the pre-trained relation extractor. Then entity pairs are clustered based on their corresponding encoding information. After labeling each cluster centroid, each entity pair (e_i, e_j) is assigned a relation r_k, which is the cluster label. The details are discussed in Section III-E.

B. Semantic Shortest Paths

Given a dependency tree of one sentence, the semantic shortest path (SSP) of two entities is defined as the shortest path from one entity (node) to the other entity (node) in the dependency tree. Razvan et al. mentioned that the semantic shortest path can capture the relation information of entity pairs [19]. Table I shows an example in which, given an entity pair and a sentence, the semantic shortest path is the path from the start entity "Ronald Reagan" to the end entity "the United States". Since only words on this path may not be sufficient to capture the relation information, we save the dependency tags D, Part-Of-Speech (POS) tags P and words W to represent this path. Note that since some entities are compound words, which can be divided into different nodes by the dependency parser, we choose the word that has a "subjective", "objective" or "modifier" dependency relation as a representative.

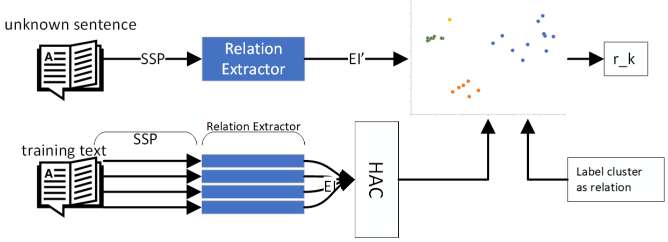

Fig. 2. The triplets clustering stage of CURE

C. Encoder

For each semantic shortest path of a given entity pair (e_i, e_j), the D, P and W sequences are embedded into vectors with different dimensions. After the embedding process, the vector representations of W, P and D are concatenated in order.

We use the Bi-directional LSTM (Bi-LSTM) [20] to encode this sequential data. After all nodes on the shortest path are encoded, the encoder concatenates each hidden state in order. The encoding information is the summation of encoding results of all shortest paths. The formal description is defined in Equation 2:

$$ei = h''_1 \oplus h''_2 \oplus \cdots \oplus h''_n \quad EI = \sum_{j=1}^{u-1} ei_j \tag{2}$$

where n is the length of each shortest path and ei_j is the encoding result of j-th shortest path. EI is the encoding information of one entity pair and h''_i is the hidden state of Bi-LSTM.

D. Decoder

In the decoder part, the words on the semantic shortest path must be generated correctly. We use a Gated Recurrent Units (GRU) neural network [21] as the basic unit of our proposed decoder. We introduce the attention mechanism to the decoder that can make the model notice only the information related to the current generation task [22]. In general, as shown in Equation 3, the attention mechanism is achieved by using attention weights to incorporate encoding information.

$$\begin{aligned} \overline{h}_i &= gru(\overline{h}_{i-1}, \overline{q}_{i-1}) \\ \overline{q}_{i-1} &= attn_\beta\left(\left(attn_\alpha(\overline{h}_{i-1}) \otimes EI\right) \oplus \overline{q}_{i-2}\right) \\ &= W_\beta\left(\left((W_\alpha \otimes \overline{h}_{i-1} + b_\alpha\right) \otimes EI) \oplus \overline{q}_{i-2}\right) \end{aligned} \tag{3}$$

where $\overline{h}_i$ is the output of the i-th GRU unit, which is the predicted probability distribution of the word at that position. $\overline{q}_{i-1}$ is the input of the GRU and the weighted information of the previous state and the encoding information. gru is the GRU function. $attn_\beta$ and $attn_\alpha$ are two different attention matrices that will be learned.

We design the loss function as the average cross entropy value of each predicted word and correct word.

E. Triplets Clustering

When training the encoder-decoder model is complete, a well-trained relation extractor is obtained. The relation extractor can use a vector to represent relation r_k. Therefore, according to the method introduced in Fig. 2, we use Hierarchical Agglomerative Clustering (HAC) to cluster similar vectors together using Euclidean distance. The result of the HAC clustering is the same as the clustering result of the entity pairs that share similar relations.

Then we extract the W corresponding to the entity pairs in each cluster, thus a candidate relation word set $\mathcal{R}$ is obtained. Based on set R, the relation word of each cluster (i.e., cluster label) can be selected using the following equation:

$$\hat{r_k} = w \ s.t. \ \underset{w}{\operatorname{argmax}} \frac{word2vec(w) \cdot v}{||word2vec(w)|| \cdot ||v||}$$

$$\text{where} \quad v = \sum_{r_i \in \mathcal{R}} N\left(\sum_{r_j \in \mathcal{R}, j \neq i}\left(1 - \frac{r_i \cdot r_j}{||r_i|| \cdot ||r_j||}\right) C(r_i)\right) r_i$$

where w is the selected relation word, r_i is the vector representation of the i-th word in $\mathcal{R}$ and $C(r_i)$ is the number

TABLE II
EXPERIMENTAL RESULTS ON NYT

Relation	Model	Rec.	Prec.	F1
company	**CURE**	**48.2**	**60.4**	**53.6**
	Open-RE	46.8	54.9	50.5
	Rel-LDA	39.4	50.7	44.3
	VAE	47.3	51.6	49.4
placeBirth	**CURE**	**47.5**	**38.2**	**42.3**
	Open-RE	38.4	31.3	34.5
	Rel-LDA	31.7	25.7	28.4
	VAE	43.2	32.9	37.4
capital	**CURE**	54.2	65.5	**59.3**
	Open-RE	53.2	**66.1**	59.0
	Rel-LDA	48.4	63.9	55.1
	VAE	**56.3**	59.8	58.0
contains	**CURE**	**56.7**	53.4	**55.0**
	Open-RE	51.6	**56.9**	54.1
	Rel-LDA	43.3	49.8	46.3
	VAE	49.1	49.0	49.0
nationality	**CURE**	39.8	**75.4**	**52.1**
	Open-RE	36.4	62.8	46.1
	Rel-LDA	31.3	64.6	42.2
	VAE	**41.3**	65.1	50.5
neighborOf	**CURE**	**43.9**	**45.1**	**44.5**
	Open-RE	42.5	43.4	42.9
	Rel-LDA	33.8	38.6	36.0
	VAE	37.1	44.0	40.3
placeLived	**CURE**	**38.7**	**33.1**	**35.7**
	Open-RE	37.4	27.6	31.8
	Rel-LDA	32.4	24.5	27.9
	VAE	35.3	32.9	34.0
children	**CURE**	52.8	**47.0**	**49.7**
	Open-RE	48.0	45.7	46.8
	Rel-LDA	44.3	42.3	43.3
	VAE	**53.1**	39.7	45.4

TABLE III
EXPERIMENTAL RESULTS ON UNPC

Relation	Model	Rec.	Prec.	F1
hasCapital	**CURE**	**62.9**	**60.2**	**61.5**
	Open-RE	60.5	58.1	59.3
	Rel-LDA	56.7	56.5	56.8
	VAE	61.6	58.3	59.9
hasNeighbor	**CURE**	**68.5**	**56.7**	**62.0**
	Open-RE	62.3	53.8	57.7
	Rel-LDA	61.4	52.6	56.6
	VAE	67.3	54.6	61.8
isCitizenOf	**CURE**	**57.6**	40.1	**47.3**
	Open-RE	55.2	39.5	46.0
	Rel-LDA	52.5	36.9	41.2
	VAE	53.1	**41.0**	46.3
isLocatedIn	**CURE**	**71.9**	**46.7**	**56.6**
	Open-RE	68.7	42.1	52.2
	Rel-LDA	66.0	39.4	49.3
	VAE	68.3	44.9	54.2
isPoliticianOf	**CURE**	**47.5**	**41.1**	**44.1**
	Open-RE	44.7	38.8	41.5
	Rel-LDA	39.2	35.7	37.2
	VAE	45.2	38.0	41.3

of occurrences of the i-th word in $\mathcal{R}$. $N(\cdot)$ is the min-max normalization function. We first project the words into a high-dimension space using a pre-trained Word2Vec model [23]. Then the vector summation of these words obtains the vector of the relation word.

The direct summation of each word vector may result in some information loss. However, intuitively, the more occurrences of a word in $\mathcal{R}$, the weight should be greater in the summation process. On the other hand, words with more occurrences in $\mathcal{R}$ may also be common words or stop words. Therefore, we add another factor, which measures the cosine similarity between the current word vector and other word vectors in $\mathcal{R}$. If the sum of the cosine similarity is higher, then the word is more similar to other words, so we lower the value of this factor.

IV. EXPERIMENTS

A. Baseline Models

We compare CURE to three state-of-the-art unsupervised relation extraction models. **Rel-LDA**: the topic distribution in LDA is replaced with triplets distribution, and similar relations are clustered using Expectation Maximization [11]. **VAE**: the variational autoencoder first predicts semantic relation given entity pairs then reconstructs entities based on the prediction. The model is jointly trained to minimize error in entity recovering [18]. **Open-RE**: corresponding sentences of entity pairs are used as features and then the features are vectorized to evaluate relation similarity [10].

B. Datasets

We use a New York Times (NYT) dataset [24] and the United Nations Parallel Corpus (UNPC) dataset [25] to train and test our model and other baseline methods.

NYT dataset. In the NYT dataset, following the preprocessing in Rel-LDA, only entity pairs that appear in at least two sentences were included in the training set, so the number of entity pairs in training set is 60K. Furthermore, all entity pairs in the testing set have been matched to Freebase [26].

UNPC dataset. The UNPC dataset is a multilingual corpus that has been manually curated. The number of entity pairs in training set is 200k and 2.6k sentences are selected to use as the testing set. Each sentence also contains at least one entity pair. The number of unique entity pairs is 1.5k in the testing set (previous work used a testing set with 1k unique entity pairs [6]). Similarly, all entity pairs in the testing set have been matched to YAGO [27].

We chose to additionally use this corpus for further evaluation for two reasons: (1) The scale of this dataset is far greater than that of NYT dataset, so the model is more likely to learn methods for extracting relation patterns. (2) To ensure that a model that achieves excellent results on NYT is not over fitting to the dataset.

C. Results on NYT

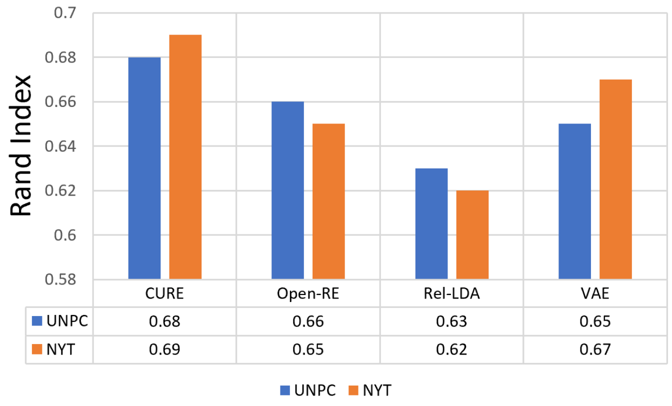

Fig. 3. Clustering Performance Results (Rand Index)

Table II shows the performance of each model on assigning relations to entity pairs, which involves relation extraction followed by clustering. We compare the models on selected relations, which appear most frequently in the testing dataset. We report recall, precision and F1 scores for each method in Table II. Since the original Rel-LDA and VAE methods did not investigate automatic cluster labeling, we compare against a variant of these methods, where we use the most frequent trigger word in each cluster as the label. Trigger words are defined by the non-stop words on semantic shortest paths. A cluster (and each entity pair in that cluster) is labeled by the relation (in Freebase) that is similar to the most frequent trigger word in that cluster. Notably, CURE achieves the highest accuracy assigning relations to entity pairs as shown in Table II.

D. Results on UNPC

Similarly, Table III reports recall, precision and F1 scores and shows that our model achieved the best performance in most relations. Although, overall, CURE outperforms all the baselines, we note that it did not perform well on some relations. In these cases, we notice that CURE performs more detailed clustering than needed. For example, given the relation "isPoliticianOf", CURE divides entity pairs in this category into finer grain subsets, such as "president" or "ambassador". Experiments on UNPC show that CURE outperforms state-of-the-art approaches on datasets of different genres or sizes and not overfit to a particular dataset to obtain positive results.

TABLE IV
CLUSTERING LABEL COMPARISON BETWEEN SELECTING RELATION WORDS BASED ON WORD VECTOR SIMILARITY (WVS) AND SELECTING RELATION WORDS BASED ON COMMON WORDS (CW)

	LABEL WORDS	RELATION
WVS CW	**metro government city** city states help	capital
WVS CW	**live stay york** york live play	placeLived
WVS CW	**born rise country** country city live	placeBirth
WVS CW	**near neighbor close** include like york	neighborOf
WVS CW	**business executive group** group expert executive	company
WVS CW	**locate include states** states country city	contains

TABLE V
ABLATION STUDY VARYING CLUSTERING METHOD AND DIMENSION RATIO (F1 SCORE)

		Ratio of encode dim. to input emb dim.				
		0.50	**0.75**	**1**	**1.50**	**1.75**
M-TO-M	**HAC**	54.9	55.0	54.1	53.7	53.8
	K-means	54.3	54.6	53.6	53.4	53.2
	GMM	54.7	55.0	54.8	54.1	53.5
1-TO-1	**HAC**	20.2	20.3	19.9	18.3	18.4
	K-means	20.4	20.7	20.1	20.0	19.2
	GMM	19.8	20.6	18.7	19.1	18.5

E. Clustering Performance

We evaluate clustering performance of each model using rand index. We implement the evaluation as follows: 1) We pair n entity pairs in the testing set together. Therefore, we obtain $\binom{n}{2}$ pairs of entity pairs. 2) We partition the testing set into m subsets using Freebase or YAGO, and into k subsets using CURE and the baseline methods. Following the definition of rand index, we then compare the m and k subsets to measure the similarity of the results of the two partitioning methods.

The rand index evaluation result is shown in Figure 3. CURE performs slightly better on NYT than on UNPC. One possible reason is that most sentences of the UNPC dataset do not directly explain the relation between two entities, so some entity pairs are assigned to more general relations.

F. Label Words Selection Evaluation

In this section, we compare the results of two approaches for **selecting relation words**: (1) based on word vector similarity (denoted as **WVS** and used by CURE), and (2) based on common words (denoted as **CW** and used by previous work [5]). We implement this evaluation as follows: (1) WVS and CW are used to generate the label of the selected cluster. (2) We compare the top three generated cluster labels with the given relation as shown in Table IV.

The relation words selected by WVS can capture the relations better than CW. For example, for the relation "contains", WVS finds words that describe the relation between two geographic locations, such as "locate" and "include". However, CW can only find that "contains" is related to each geographical division, such as "State" and "country". Moreover, the candidate word lists generated by WVS and CW have different orders. For example, for the relation "company", CW regards "group" as the best word to describe the relation and puts "executive" in the last place. This arrangement is not consistent with facts, because "company" in Freebase emphasizes the relation between the company's leader or owner and the company. WVS arranges its candidate words list differently and more accurately, putting "business" in the first place and "executive" in the second place.

G. Ablation Study

In Table V, we investigate different clustering methods used during the test stage while varying the encoding and input embedding dimension. In particular, the columns of Table V represent the ratio of encoding information dimension to input embedding dimension. For this experiment, we report F1 score of the "contains" relation, which is the most popular relation in the NYT dataset. Note we used multi-path, HAC, and 0.75 embedding ratio in our previous experiments, which is the default settings of CURE. We also provide results for the one-to-one setting, that is, one path is used to predict one semantic shortest path. Overall, using multiple paths to predict one semantic shortest path significantly outperforms the one-to-one setting across all clustering methods as shown in Table V. In terms of clustering, HAC and GMM perform best for different encoding dimension to input embedding dimension ratios.

V. Conclusion

In this paper, we proposed a Clustering-based Unsupervised Generative Relation Extraction (CURE) framework to extract relations from text. The CURE relation extractor is trained using the correlations between sentences with the same entity pair. The CURE clustering approach then uses the relation information identified by the relation extractor to cluster entity pairs that share similar relations. Our experiments demonstrate that including sentence correlation improves unsupervised generative clustering performance by comparing our approach to three state-of-the-art baselines on two datasets.

References

[1] C. Xiong, R. Power, and J. Callan, "Explicit semantic ranking for academic search via knowledge graph embedding," in *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1271–1279.

[2] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5329–5336.

[3] Y. Zhang, H. Dai, Z. Kozareva, A. J. Smola, and L. Song, "Variational reasoning for question answering with knowledge graph," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] G. Angeli, M. J. J. Premkumar, and C. D. Manning, "Leveraging linguistic structure for open domain information extraction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 344–354.

[5] T. Hasegawa, S. Sekine, and R. Grishman, "Discovering relations among named entities from large corpora," in *Proceedings of the 42nd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2004, p. 415.

[6] Y. Yan, N. Okazaki, Y. Matsuo, Z. Yang, and M. Ishizuka, "Unsupervised relation extraction by mining wikipedia texts using information from the web," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1021–1029.

[7] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld, "Open information extraction from the web," *Communications of the ACM*, vol. 51, no. 12, pp. 68–74, 2008.

[8] N. Nakashole, G. Weikum, and F. Suchanek, "Patty: a taxonomy of relational patterns with semantic types," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1135–1145.

[9] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2011, pp. 1535–1545.

[10] H. Elsahar, E. Demidova, S. Gottschalk, C. Gravier, and F. Laforest, "Unsupervised open relation extraction," in *European Semantic Web Conference*. Springer, 2017, pp. 12–16.

[11] L. Yao, A. Haghighi, S. Riedel, and A. McCallum, "Structured relation discovery using generative models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1456–1466.

[12] N. Konstantinova, "Review of relation extraction methods: What is new out there?" in *Analysis of Images, Social Networks and Texts*, D. I. Ignatov, M. Y. Khachay, A. Panchenko, N. Konstantinova, and R. E. Yavorsky, Eds. Cham: Springer International Publishing, 2014, pp. 15–28.

[13] D. P. Nguyen, Y. Matsuo, and M. Ishizuka, "Relation extraction from wikipedia using subtree mining," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 22, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, p. 1414.

[14] B. Rozenfeld and R. Feldman, "High-performance unsupervised relation extraction from large corpora," in *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 2006, pp. 1032–1037.

[15] J. Chen, D. Ji, C. L. Tan, and Z.-Y. Niu, "Unsupervised feature selection for relation extraction," in *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*, 2005.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[17] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1442–1451.

[18] D. Marcheggiani and I. Titov, "Discrete-state variational autoencoders for joint discovery and factorization of relations," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 231–244, 2016.

[19] R. C. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction," in *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, 2005, pp. 724–731.

[20] D. Zhang and D. Wang, "Relation classification via recurrent neural network," *arXiv preprint arXiv:1508.01006*, 2015.

[21] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[24] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 148–163.

[25] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen, "The united nations parallel corpus v1. 0," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 3530–3534.

[26] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[27] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.