A Experimental Setup

A.1 Evaluation

For the NYT dataset, we adopt the standard micro F1 score, recall (R) and precision (P) as the metrics for the NER and RE subtasks. For the final result, a correct prediction is that the extracted triple matches the ground truth including two entities, relation direction and relation type. For the NER subtask, we consider the entity type, length, and position in sentences. For the SemEval-2018 dataset, we use the official script which calculates the macro-average F1 score.

A.2 Training and Hyperparameters

The input word is projected to a 200-D pre-trained GloVe (Pennington et al., 2014) word embedding. The hidden state of the encoder is 300-D. The dimension of GRNN units (including the LSTM, GRU and RCNN units) is 200-D. We set $\lambda = 2.0$. For regularization we apply dropout with $p = 0.5$. To select better models, we divide the NYT training data into 100 pieces and the training data of SemEval-2018 into four pieces. We set $k = 1$ and $k = 6$ on SemEval-2018 and NYT dataset respectively. We first use single-sample training to get a good model and then adopt the batch (64) training to fine tune the model. We use the Adam optimization algorithm to update model parameters. The initial learning rate is 0.001 and the decay rate is 0.9. We ran the experiments on an Intel(R) Xeon(R) CPU E7-4830 v3 @ 2.10GHz (Mem: 976G) and one GPU Tesla K40c. The training time for NYT and SemEval-2018 datasets are 8h/epoch and 0.1h/epoch respectively.

B Training Algorithm

We describe the inference and feedback details of Algorithm 1 as follows:

(1) In the inference step, this model first predicts/infers the label sequence, as shown in line 1, and uses the combination of the predicted entities as triple candidates, as shown in line 2-3. Then, the model dynamically extracts subgraphs of multiple triples to construct the forest on the top layer, as shown in line 5-9. This model uses the indices of s and o to find the LCA index as p_w. Then, the model will map the above indices into the GRNN to get the vector representations for a triple as shown in line 8. In line 10-16, this model also memorizes the unexpected entity pairs that are not predicted, using the same operations of line 5-9.

(2) In the feedback step, this model generates the relation type assumption for each predicted triple by using a single layer neural network, as shown in line 17. Then, it compares the assumptions with the ground truth to get the feedback signal to update the model state through backpropagation. The feedback signal is generated by the loss function to calculate the gradients.

This algorithm contains two insights. We incorporate inference and interpretability into the process of model learning. First, we propose to use inference operations to make it a lifelong learning model. This model is an organism that also deals with multi-task learning. During model training, relation candidates are generated to localize relation positions. Second, we present a scheme for modeling graph data to learn the vertex representations that the sequence model cannot learn. GRNN encodes long-range syntactic relations by encoding the dependency structure. We can use the distribution of vertex embedding to explain the relation classification process in a deep learning model. Multi-task learning is a convenient way to integrate two subtasks.

C Ablation Study

Table 1 shows an ablation study of multi-task training and pipeline training. Two systems denote a fine-tuned NER system and a GRNN system. Here we use a BERT model as the NER system. We first compare the multi-task training with different settings. A first observation is that a fine-tuned NER system can slightly improve the final results by 0.76% F1. Because it reduces the unknowns in the testing stage by explicitly indicate the entities. The draw back is that the training process is a multi-stage pipeline and relation candidates are determined by the NER system. We first fine tune the NER system and then predict the label sequence $\hat{Y}_{ner}$ which are also written to disk. Then we train the joint entity and relation extraction model that can use the $\hat{Y}_{ner}$ information. Although the NER system is enhanced, some entities still hurt the final precision. This is because extracting more entities does not always help the relation extraction subtask and some entities may increase the risk of predicting false positives. When we replace $\hat{Y}_{ner}$ with the ground truth NER labels, the result is significantly improved

(5.7% F1). This means that a high quality label sequence can help improve the final results.

Removing the multi-task training degrades the performance by 4.4% F1. This means that multi-task training benefits the RE subtask from the NER subtask. When we remove the NER system, performance drops slightly (0.76% F1). Then, we remove the NER system and multi-task training. We first train the NER subtask, then freeze the parameters of shared layer, and then train the RE subtask. The result is reduced by 3.24% F1 because the RE subtask cannot be encoded at the lower layer to interact with the NER subtask. This means that multi-task joint training is critical for subtask interaction. When we do not freeze the lower layer, the result is much lower. Because when we train the RE subtask, the memory for NER subtask is weakening.

We also apply our training pipeline to BERT, a pre-trained bidirectional transformer that has proven very effective in many NLP tasks. BERT has hundreds of millions parameters and has been trained on large-scale corpus. We fine tune a simple BERT model for TE. Table 2 shows the result. When only using multi-task training, BERT achieves slightly higher results than our GRNN model. The downside is the cost of computation. When we remove the multi-task, NER system and Frozen NER, this model almost forgets all the NER memory, so the RE subtask fails. Model fine-tuning can achieve different functions, which also means that BERT is sensitive to parameter changes.

Figure 1 shows the class-aware prediction result. Figure 1a and 1b refer to the model with or without the ground truth NER label when predicting the result. We observe "*place_lived*", "*nationality*", "*contains*" and "*company*" relations are easier to extract. Other relations in the test set are not extracted. This might be caused by the class imbalance problem in the training data. Some sparse relations are not fully learned. Comparing Figure 1a and 1b, we observe that providing NER labels enhances the extraction of well-trained relation, while other relations are not obviously improved.

D Case Study

As a case study for evaluating how the GRNN classify the relations, we pick some samples from NYT dataset and analyze the result of Bi-GRNN-LSTM model.

Figure 2 shows the results. In the first sample, our model extracts the triple (*Hurley, company, YouTube*) since the *Hurley* and *YouTube* share the relation vertex "*co-founder*" in the dependency graph. This relation vertex implies the *"/business/person/company"* relation.

In the second sample, our model extracts three triples. For (*New York City, neighborhood of, Lower Manhattan*), in the dependency graph, the vertex "*began*" appears in the path of *New York City* to *Lower Manhattan*. "*began*" might generate the relation type "*neighborhood of*" when two entities are locations. One mistake is (*New York, contains, New York City*) which means *"New York"* contains its own. Two entities share the same relation vertex "*grow*" in the dependency path *New York* ← *Cities* ← *grow* → *tends* → *urgently* → *Manhattan* → *began* → *New York City*. This relation vertex leads to the relation "*contains*". For the left triple, the situation is similar to the above description. This means literally considering the relationship might cause a misleading. Understanding complex spatial relationships needs to be further improved.

For the third sentence *"George Eastman House in Rochester"* implies that (*Rochester, contains, George Eastman House*). *" Grant Romer ... at George Eastman House in Rochester"* only indicates a state but does not mean the relation type *"place lived"*. Understanding time state of relationships needs to be further improved.

For the forth sentence, it detects the entity mentions but not the relation type. Two entities share the same relation vertex "*opposition*" so the predicted (*Syria, contains, Iran*) relation might be a coincidence. Because the training data is generated by distant supervision technique, some samples may produce noise. This also indicate that we need to reduce the noise of training data. For the last sentence, "*Prime, Minister, Israel*" are children of "*Olmert*", which decides the "*nationality*" relation.

E Vertex Representation Analysis

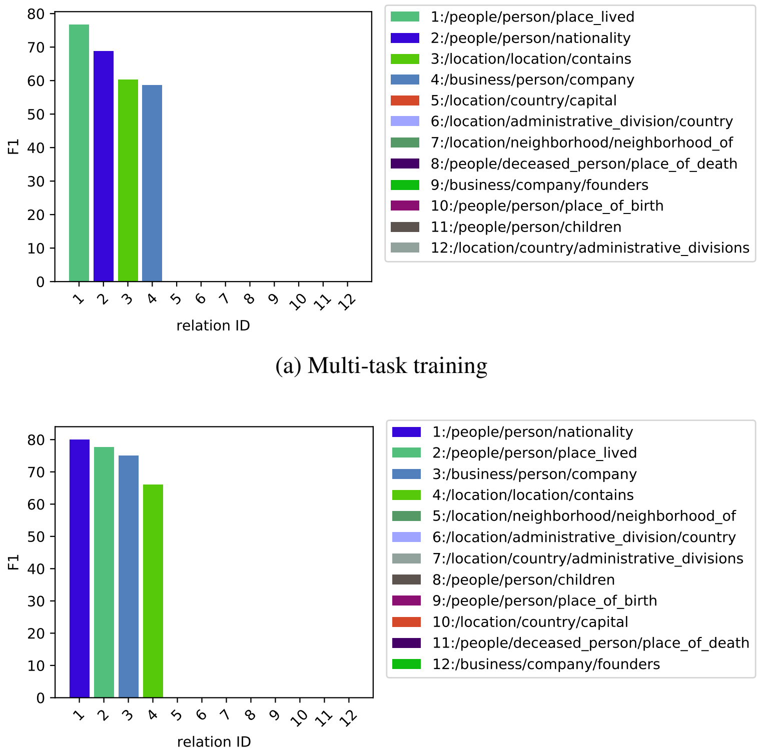
To analyze the vertex embeddings learned for relation classification, Figure 3 shows the visualization of vertex embeddings distribution. We pick some samples from the NYT dataset and use PCA and t-SNE for dimensionality reduction. As shown in Figure 3a, dominant vertices for

Table 1: Bi-GRNN-LSTM model setting ablation

Model	Precision	Recall	F1
Two systems + multi-task	54.64	52.15	53.36
+ NER label	60.26	57.97	59.06
–multi-task	47.39	50.63	48.95
–NER system	54.9	50.5	52.6
–multi-task, NER system	49.61	49.11	49.36
–multi-task, NER system, Frozen NER	38.87	33.16	35.79

Table 2: BERT model setting ablation

Model	Precision	Recall	F1
Two systems + multi-task	51.30	54.93	53.05
+ NER label	54.89	61.01	57.79
–multi-task	50.46	54.43	52.37
–NER system	50.68	56.20	53.30
–multi-task, NER system	46.46	54.93	50.34
–multi-task, NER system, Frozen NER	–	–	–

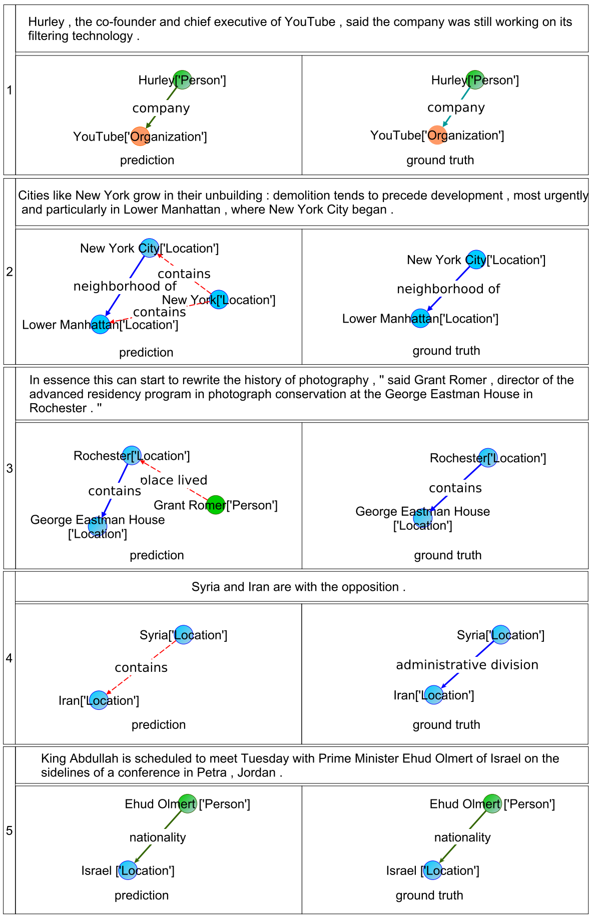


(a) Multi-task training

(b) Multi-task training + NER label

Figure 1: Class-aware result

relation classification are represented as colored points placed according to their coordinates. Different colors denote different relation types. PCA tends to reveal the semantic relationship between relation types. We first observe that there are two main clusters, where the cluster of "*contains*" and "*neighbor_of*" relations describes the relationship between locations, while the other cluster describes the relations about person and organization/location. The lower part of the diagram shows the relations of "*place_lived*", "*company*" and "*nationality*" are closely related, but the relations are mixed. t-SNE tends to show a more accurate relationship between relations. As shown in Fig-



Hurley , the co-founder and chief executive of YouTube , said the company was still working on its filtering technology .

Cities like New York grow in their unbuilding : demolition tends to precede development , most urgently and particularly in Lower Manhattan , where New York City began .

2
New York City['Location']
contains
neighborhood of
New York['Location']
contains
Lower Manhattan['Location']
prediction
New York City['Location']
neighborhood of
Lower Manhattan['Location']
ground truth

In essence this can start to rewrite the history of photography , " said Grant Romer , director of the advanced residency program in photograph conservation at the George Eastman House in Rochester . "

Syria and Iran are with the opposition .

King Abdullah is scheduled to meet Tuesday with Prime Minister Ehud Olmert of Israel on the sidelines of a conference in Petra , Jordan .

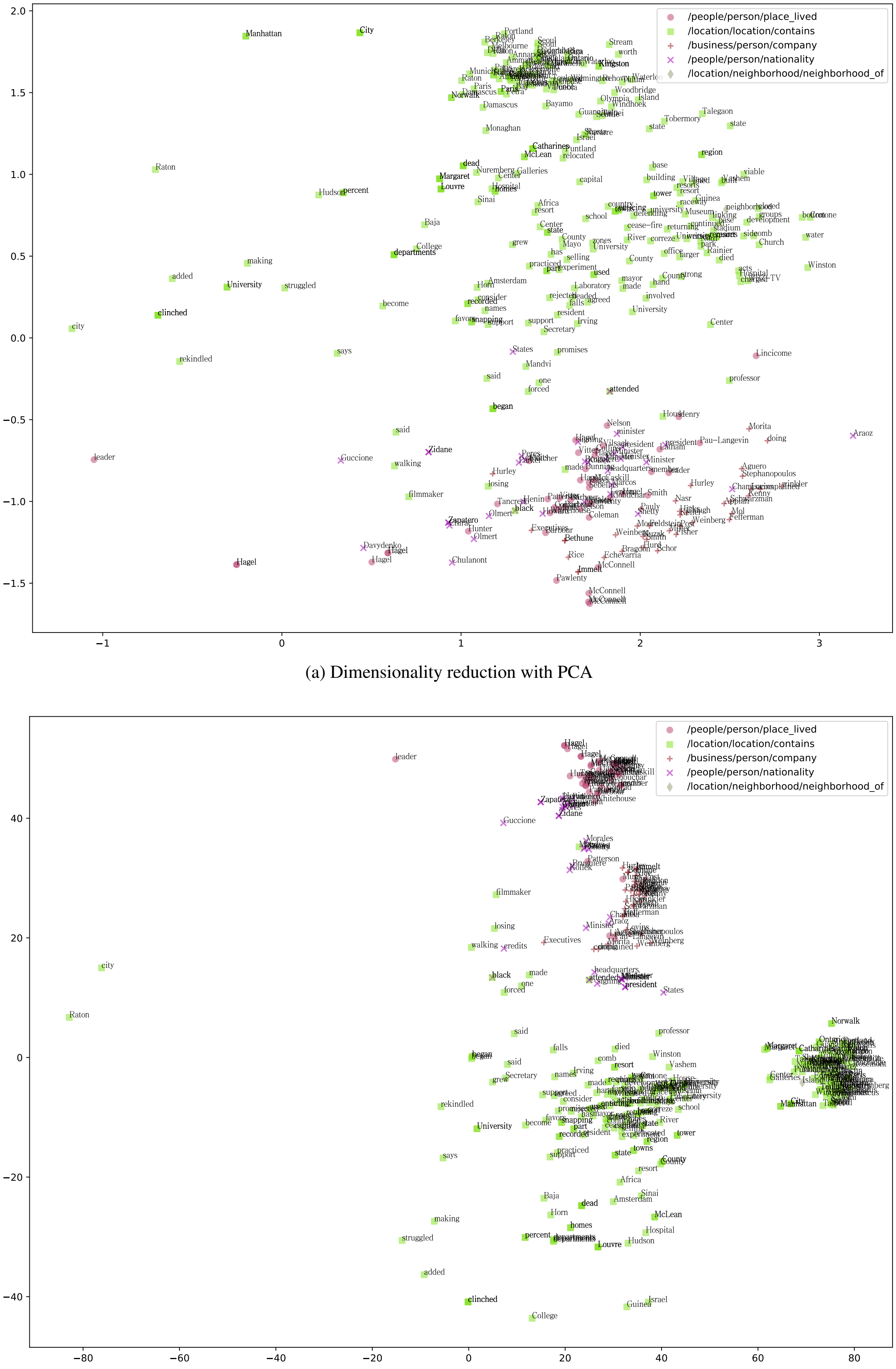
Figure 2: Some extracted triples where the left side is the prediction results and the right side is the ground truth, and the red edge denotes the inconsistent predictions, and the other colors are the consistent predictions

ure 3b, the "*placed_lived*" is closer to "*nationality*". "*nationality*" is closely related to "*company*", while the "*place_lived*" is not always close to "*company*". This phenomenon is also in line with the actual situation. This GRNN, by leveraging vertex embeddings, provides better understanding about relations. This means the vertex embeddings encodes the semantic relationship between relation types.

F Dataset Overview

Figure 4a shows the relation distribution of the NYT dataset. The x- and y-axes are the relation ID and relation number respectively. We first observe there are more than 50,000 "*contains*" relations while others are less than 9,000. Each relations from 11 to 24 occurs less than 1,000 times. Figure 4b shows that "*contains*" relation occupies the main part (48.4%), while the relations from 9 to 24 account for 7.7%. This dataset has class imbalance problem, which poses a challenge to model performance. Figure 4c shows the average distance of each relation type between subject and object. The x- and y-axes are relation ID and token number respectively. We observe that "*industry*", "*geographic_distribution*", "*place_of_death*" relations are often described in a long context, and "*profession*", "*neighborhood_of*" and "*founders*" are more likely described in a short context. Figure 4d shows the token number of samples. The x- and y- axes denote the token number and sample number respectively. We observe the average token number is 37.8.

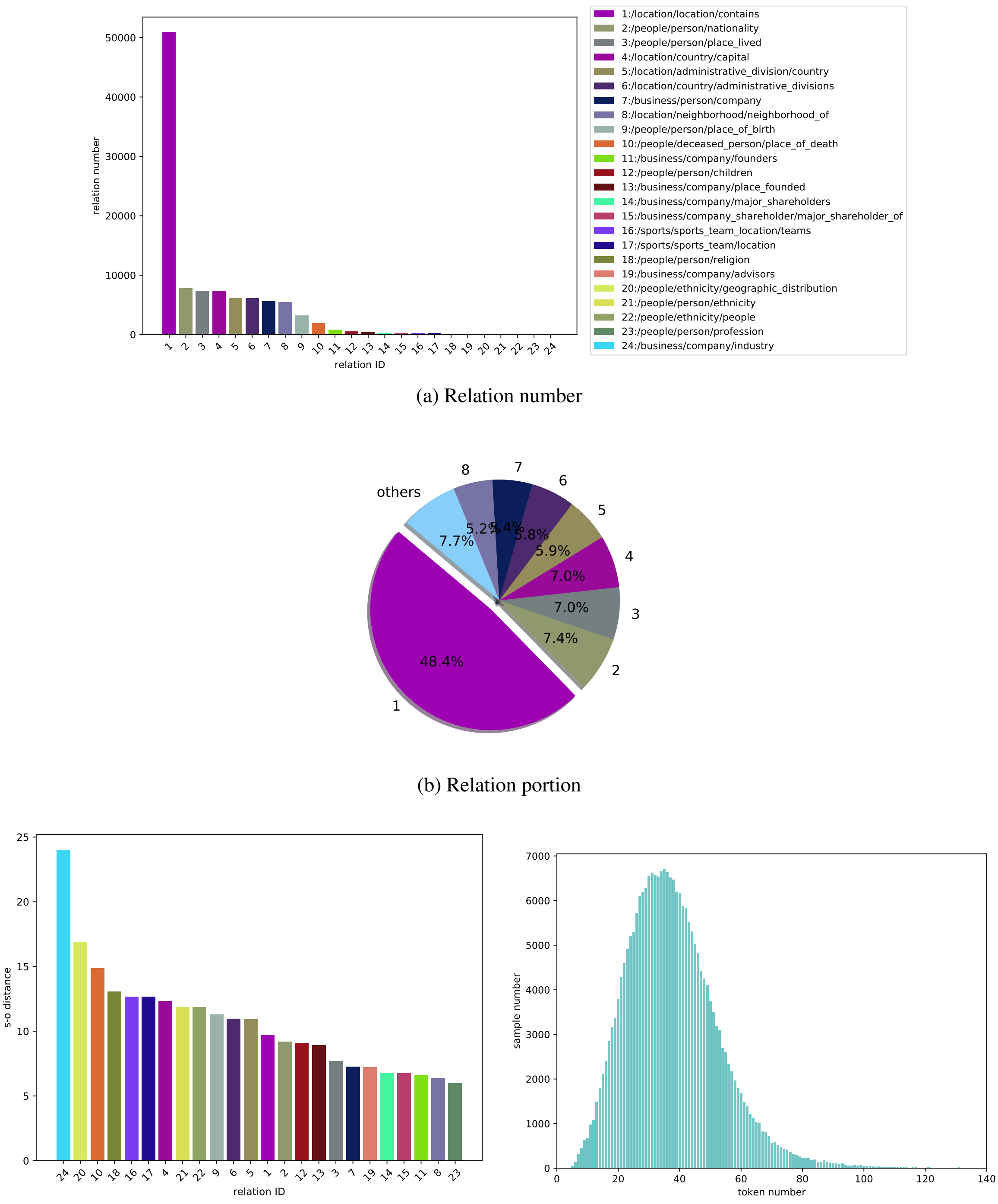
Figure 5a shows the relation distribution of the SemEval-2018 dataset. "*USAGE*" relation occupies the main part (39.3%). "*TOPIC*" relation is sparse so it has more impact for the final results. Because the evaluation process uses a macro-averaged F1 score. Although this dataset has less types of relations than NYT dataset, the data sparsity problem poses a challenge to the model. Figure 5c shows the "*COMPARE*" relationship is more likely described with more words, while the "*MODEL-FEATURE*" and "*PART_WHOLE*" relations are more likely expressed with less words. We observe the average distance between subject and object entities is also closer than NYT dataset. Figure 5d shows the token number of samples. We observe the average token number is 25.8.



(a) Dimensionality reduction with PCA

(b) Dimensionality reduction with t-SNE

Figure 3: Vertex embeddings visualization



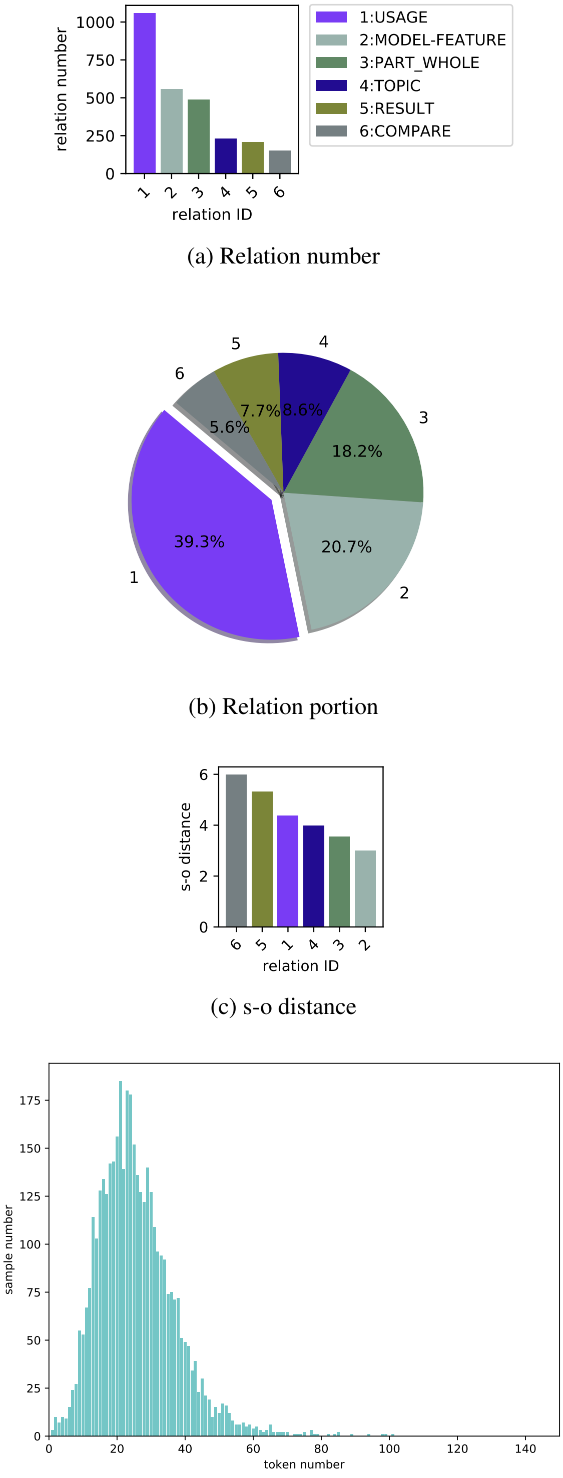
(a) Relation number

(b) Relation portion

(c) s-o distance

(d) Token number

Figure 4: NYT Dataset visualization



(a) Relation number

(b) Relation portion

(c) s-o distance

(d) Token number

Figure 5: SemEval-2018 dataset visualization