Hybrid Attention-Based Prototypical Networks for Noisy Few-Shot Relation Classification

AAAI 2019

Motivation

- Relation Classification:
  – Finding the semantic relation between entity mentions in text
- Supervised Learning:
  – Manual labeling is time-consuming
- Distant Supervision:
  – Use the relation of two entities in a Knowledge Base as the semantic relation of two entity mentions in text
  – Introduce noise to labeling
  – Few instances for rare relations

Motivation

- Model RC as Few-shot Learning (FSL)
  - Few examples per relation
- Approaches for FSL:
  - Transfer learning: Use information of abundant labels for rare labels
  - Metric Learning: Learn distance distributions among labels
  - Meta Learning: Learn to learn
    - Prototypical networks

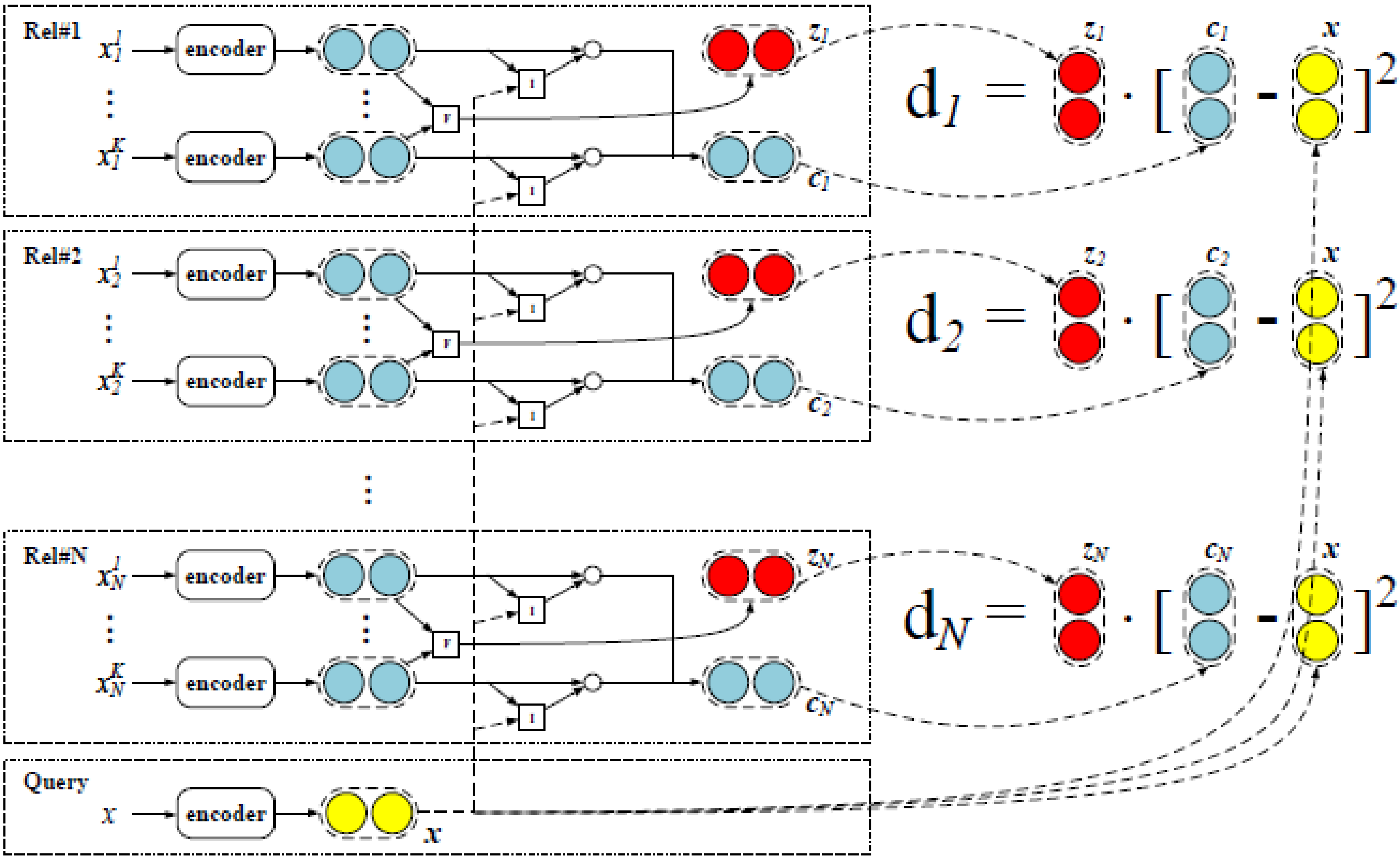
Motivation

- Prototypical networks are mainly used for CV
- Challenges for NLP:
  - More diversity
  - More noise
- This paper address:
  - RC with rare instances per class and noisy labels
  - Use prototypical network as a technique to model RC as FSL addressing diversity and noise in prototypical networks

Contributions

- Introducing Two levels of attention:
  - Feature level: Select most useful features for computing prototypes
  - Instance level: Selects most useful instances in support set based on the given query
- Analyzing robustness to noise:
  - Compared to vanilla prototypical network their approach is more robust to noise in labels

Model



Rel#1
encoder
z_1
c_1
Rel#2
z_2
c_2
Rel#N
z_N
c_N
Query
x
$d_1 = z_1 \cdot [c_1 - x]^2$
$d_2 = z_2 \cdot [c_2 - x]^2$
$d_N = z_N \cdot [c_N - x]^2$

Model

- Inputs to model:

$$\begin{aligned}\mathcal{S} = \{&(x_1^1, h_1^1, t_1^1, r_1), \ldots, (x_1^{n_1}, h_1^{n_1}, t_1^{n_1}, r_1),\\ &\ldots\\ &(x_m^1, h_m^1, t_m^1, r_m), \ldots, (x_m^{n_m}, h_m^{n_m}, t_m^{n_m}, r_m)\},\\ &r_1, r_2, \ldots, r_m \in \mathcal{R},\end{aligned}$$

- Use CNN to encode the sentences:
- Inputs to CNN:
  - GloVe embedding
  - Position Embedding

Prototypical Network

- Find prototypes:

$$\mathbf{c}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathrm{x}_i^j,$$

- Classify query:

$$p_\phi(y = r_i | x) = \frac{\exp\big(-d(f_\phi(x), \mathbf{c}_i)\big)}{\sum_{j=1}^{|\mathcal{R}|} \exp\big(-d(f_\phi(x), \mathbf{c}_j)\big)},$$

Instance Level Attention

- Compute attention weight for each instance in support set based on the query instance:

$$\alpha_j = \frac{\exp(e_j)}{\sum_{k=1}^{n_i} \exp(e_k)},$$

$$e_j = \mathrm{sum}\Big\{\sigma\big(g(\mathbf{x}_i^j) \odot g(\mathbf{x})\big)\Big\},$$

- Find Prototypes:

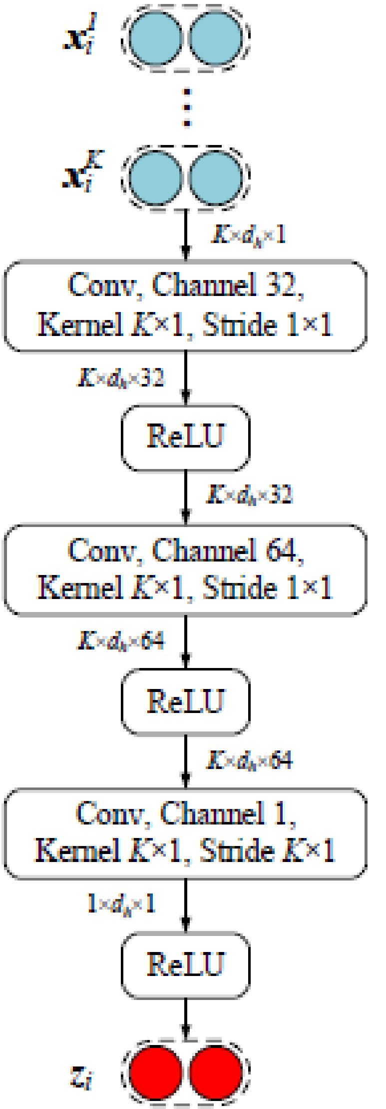
$$\mathbf{c}_i = \sum_{j=1}^{n_i} \alpha_j \mathbf{x}_i^j.$$

Feature Level Attention

- Find attention weight per feature of prototypes:
  - 2D CNN

- Find distance of query to prototypes:

$$d(\mathbf{s}_1, \mathbf{s}_2) = \mathbf{z}_i \cdot (\mathbf{s}_1 - \mathbf{s}_2)^2$$



Experiments

- FewRel:
  - Training: 64 relations
  - Dev: 16 relations
  - Test: 20 relations
  - 700 instances per relation

- Noise Level:
  - Randomly change relation labels to wrong labels

Parameters

Convolutional Window Size m	3
Word Embedding Dimension d_w	50
Position Embedding Dimension d_p	5
Hidden Layer Dimension d_h	230
Batch Size	4
Training Classes for One Batch	20
Initial Learning Rate	0.1
Weight Decay	10^{-5}
Learning Rate Decay γ	0.1

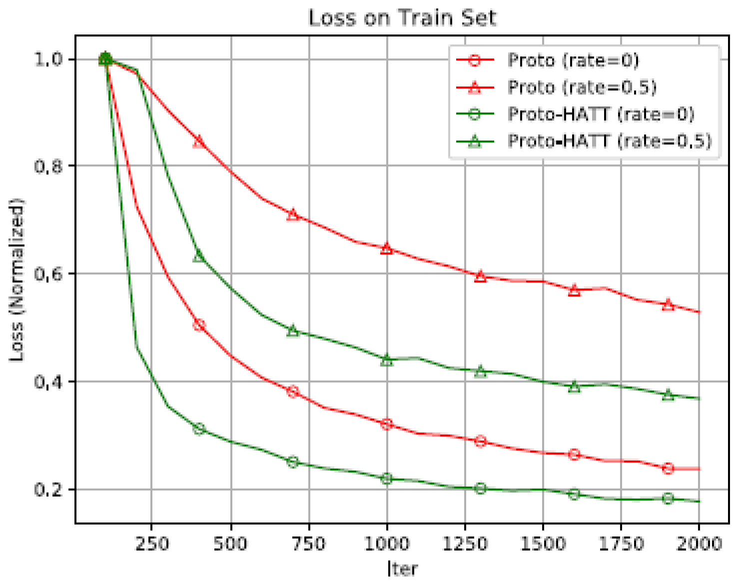
Robustness to noise

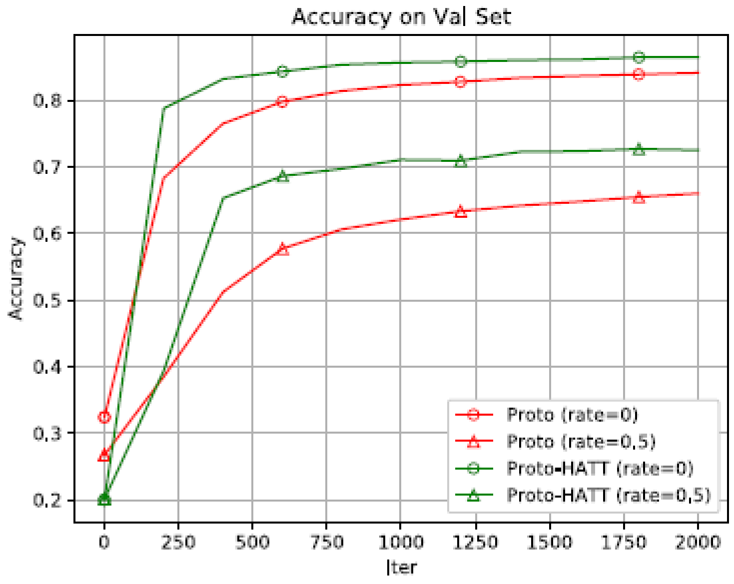
Noise Rate	Model	5 Way 5 Shot	5 Way 10 Shot	10 Way 5 Shot	10 Way 10 Shot
0%	Proto	89.05 ± 0.09	90.79 ± 0.08	81.46 ± 0.13	84.01 ± 0.13
	Proto-HATT	$\mathbf{90.12 \pm 0.04}$	$\mathbf{92.06 \pm 0.06}$	$\mathbf{83.05 \pm 0.05}$	$\mathbf{85.97 \pm 0.08}$
10%	Proto	87.63 ± 0.10	90.15 ± 0.08	79.39 ± 0.14	83.05 ± 0.12
	Proto-HATT	$\mathbf{88.74 \pm 0.06}$	$\mathbf{91.45 \pm 0.05}$	$\mathbf{81.09 \pm 0.08}$	$\mathbf{85.08 \pm 0.07}$
30%	Proto	82.45 ± 0.09	87.64 ± 0.07	72.43 ± 0.12	79.31 ± 0.11
	Proto-HATT	$\mathbf{84.71 \pm 0.07}$	$\mathbf{89.59 \pm 0.05}$	$\mathbf{75.68 \pm 0.11}$	$\mathbf{82.43 \pm 0.07}$
50%	Proto	72.91 ± 0.15	81.71 ± 0.10	61.11 ± 0.17	71.29 ± 0.14
	Proto-HATT	$\mathbf{76.57 \pm 0.07}$	$\mathbf{85.17 \pm 0.09}$	$\mathbf{65.97 \pm 0.11}$	$\mathbf{76.42 \pm 0.13}$

Comparing to Baselines

Model	5 Way 5 Shot	10 Way 5 Shot
Finetune*	68.66 ± 0.41	55.04 ± 0.31
kNN*	68.77 ± 0.41	55.87 ± 0.31
Meta Network*	80.57 ± 0.48	69.23 ± 0.52
GNN*	81.28 ± 0.62	64.02 ± 0.77
SNAIL*	79.40 ± 0.22	68.33 ± 0.25
Proto*	84.79 ± 0.16	75.55 ± 0.19
Proto	89.05 ± 0.09	81.46 ± 0.13
Proto-IATT	89.63 ± 0.08	82.16 ± 0.13
Proto-FATT	89.70 ± 0.03	82.45 ± 0.05
Proto-HATT	$\mathbf{90.12 \pm 0.04}$	$\mathbf{83.05 \pm 0.05}$

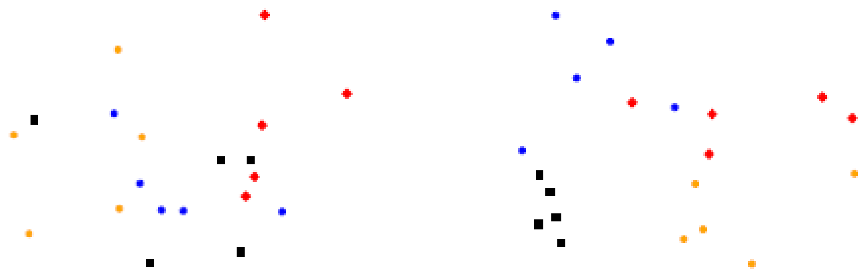
Convergence Speed





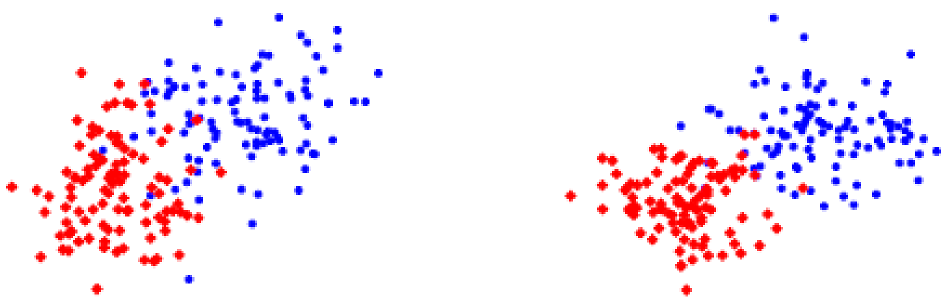
Representation Visualization

- Feature Attention:



(a) Features with lower scores. (b) Features with higher scores.

- Sentence encoding with attention:



(c) Emb trained without HATT. (d) Emb trained with HATT.

Question?