Parasitic Neural Network for Zero-Shot Relation Extraction

Shengbin Jia*
Tongji University
Shanghai, China
shengbinjia@tongji.edu.cn

Shijia E
Tencent Ins
Shanghai, China
allene@tencent.com

Yang Xiang
Tongji University
Shanghai, China
shxiangyang@tongji.edu.cn

ABSTRACT

Conventional relation extraction methods can only identify limited relation classes and not recognize the unseen relation types that have no pre-labeled training data. In this paper, we explore the zero-shot relation extraction to overcome the challenge. The only requisite information about unseen types is the name of their labels. We propose a Parasitic Neural Network (PNN), and it can learn a mapping between the general feature representation of text samples and the distributions of unseen types in a shared semantic space. Experiments show that our model significantly outperforms others on the unseen relation extraction task and achieves effect improvement more than 20%, when there are no manual annotations or additional resources.

KEYWORDS

relation extraction, zero-shot, neural network, knowledge graph

ACM Reference format:
Shengbin Jia, Shijia E, and Yang Xiang. 2016. Parasitic Neural Network for Zero-Shot Relation Extraction. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17)*, 5 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Relation Extraction (RE) task aims to determine relational facts from the unstructured text and can populate knowledge bases or benefit downstream knowledge-driven applications, for example, information retrieval, question answering.

The conventional methods (including one/few -shot learning) [2, 5] can not meet practical needs of the relation extraction. Generally, there are massive fine-grained types of relations in the real world. However, these methods are often to distinguish the seen relational taxonomy, where the relation types are limited and each type must have a certain number of labeled samples. They are unable to generalize to new (unseen) relations (i.e., they will break down when predicting a type that has no training examples). Collecting sufficient labeled instances for training on all expected categories is almost impossible, in contrast with the limited number of relation types covered by existing datasets.

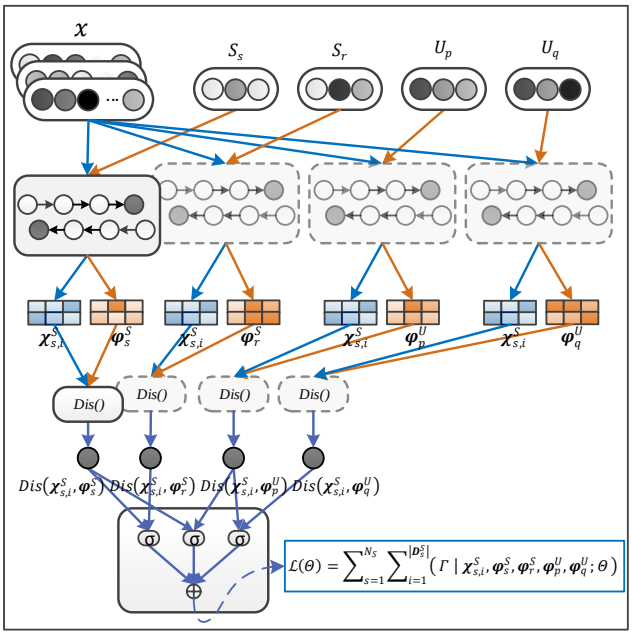

Figure 1: The architecture of the Parasitic Neural Network.

To address the challenge, we develop a Zero-Shot Relation Extraction (ZRE), which is under the restriction that the extractor should identify facts of new relation types after learning from limited labeled instances of seen types. The ZRE is a powerful and promising learning paradigm by reducing annotation costs and improving application efficiency. However, it is immature and has received limited attention.

The existing popular methods address the ZRE task to develop specific transfer learning procedures by reading comprehension [6], textual entailment [10] and so on. We consider those methods to be **indirect-trick**. They need much artificial descriptive information to improve the understandability of relation types, Annotation cost severely decreases their applicability to new types. In this paper, we are committed to the **direct-trick** method. It does not need any manual intervention to pre-describe relation types. Instead, it just uses the names of type labels.

Furthermore, we propose the Parasitic Neural Network (PNN) model. It learns the mapping between the text feature representations and the type embeddings (prototypes) in a shared semantic space. This mapping is invariable and does not depend on a certain type or instance so that it successfully adapts to unseen types. The model requires to solve two key problems: (1) how to learn the distributions of unseen types in the shared space, (2) how to

*This author is the one who did all the really hard work.

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

obtain the general features that can be shared across the relation categories.

In summary, our key contributions are presented as follows. (1) We develop a general zero-shot learning framework for unseen relation extraction by the direct-trick. It emphasizes to use no manual annotations or external knowledge. (2) We propose the PNN that leverages the association of the relation types in a shared semantic space to learn the distributions of unseen types automatically. It can also learn the general features of the text samples by using a special tied network structure. (3) Our experimental results achieve significant improvement than other direct-trick methods and most indirect-trick methods.

2 RELATED WORK

Most of the works of the zero-shot learning were focused on the area of computer vision [12, 17]. In the area of nature language processing, the applications of zero-shot learning have been emerging in recent years, such as entity typing [11], event extraction [3] and knowledge graph completion [1, 13].

As for zero-shot relation extraction [1], it is immature and has received limited attention. By analyzing linguistics, old-fashioned approaches developed unsupervised models (e.g., clustering) based on combinations of manual features, patterns or corpus-level resources [4, 9, 18]. They tended to be inefficient and consumed much manpower. The recent methods were to transfer other tasks to produce relations. Levy et al. [6] formulated relation types as various parametrized natural-language questions, then used a reading comprehension model to process the questions to obtain relation facts. By considering the text and the relation description as the premise and hypothesis respectively, Obamuyide et al. [10] transformed the extraction task to determine the truthfulness of the hypothesis by a textual entailment model. They were expensive to manually formulate reading comprehension questions or entailment rules. In addition, transfer-based methods were constrained by the capability of indirect tasks whose errors or defects could be cascaded into the relation extraction.

Therefore, we explore the ZRE via the direct-trick. Although the concept of the direct-trick has a few similar studies in other fields [12], to our knowledge, it is the first on the RE task. Besides, we perform the zero-shot learning by automatically extending annotations for unseen classes. This idea is simple but novel.

3 METHODOLOGY

3.1 Approach Overview

Let $\boldsymbol{S} = \{S_s \mid s = 1, \cdots, N_S\}$ denotes a set of seen relation types and $\boldsymbol{U} = \{U_u \mid u = 1, \cdots, N_U\}$ unseen types, with $\boldsymbol{S} \cap \boldsymbol{U} = \emptyset$. Suppose that the dataset $\mathbb{D} = \mathbb{D}^S \cap \mathbb{D}^U$ is a collection of text instances. The $\mathbb{D}_s^S = \left\{x_{s,i}^S \mid y_s^S = S_s\right\}$ is as the set of labeled training instances belonging to seen types S_s. The $\mathbb{D}^U = \left\{x_j^U\right\}$ is as the set of testing instances, meanwhile, $y_j^U \in \boldsymbol{U}$ is to be predicted as the corresponding type labels for x_j^U. In semantic embedding space $\mathbb{R}^z$, the instance x will be embedded to $\boldsymbol{\chi}$ and it is assumed to belong to one category. The types will be vectorized as type prototypes $\boldsymbol{\varphi} = \left\{\boldsymbol{\varphi}^S, \boldsymbol{\varphi}^U\right\}$. Overall, the ZRE learning task is defined as: Given $\mathbb{D}^S$, the ZRE system learns the mapping $f(\cdot) : \boldsymbol{\chi} \rightarrow \boldsymbol{\varphi}$, which can classify testing instances $\mathbb{D}^U$ (i.e., to predict y^U).

[1]The ZRE extracts the canonical relations independent of how the original text is phrased. It is different from the Open Relation Extraction that can discover rough and non-standardized relation facts.

Algorithm 1 Parasitic neural network training algorithm.

Require: $\boldsymbol{S}, \mathbb{D}^S, \boldsymbol{\varphi}^S, \boldsymbol{U}, \boldsymbol{\varphi}^U$.

1: Calculate the semantic distances of seen types $\boldsymbol{S}$ to unseen types $\boldsymbol{U}$, as,
$$\boldsymbol{D}(S_s, U_u) = Dis(\boldsymbol{\varphi}_s^S, \boldsymbol{\varphi}_u^U) \mid s = 1, ..., N_S,\ u = 1, ..., N_U.$$
2: Obtain the array $\boldsymbol{R}$ by ranking the $\boldsymbol{D}(\boldsymbol{S}, \boldsymbol{U})$ (from small to large),
$$\forall\{\ s = 1, ..., N_S,\ m = 1, ..., N_U - 1\} \text{ s.t.}$$
$$\boldsymbol{R}[s][m] \in \boldsymbol{U} \wedge \boldsymbol{D}(S_s, \boldsymbol{R}[s][m]) \leqslant \boldsymbol{D}(S_s, \boldsymbol{R}[s][m+1]).$$
3: **for** S_s (as *Host*) in $\boldsymbol{S}$ **do**
4: **for** $x_{s,i}^S$ in $\mathbb{D}_s^S$ **do**
5: Select S_r from $\boldsymbol{S}$ randomly;
6: Select any $U_p = \boldsymbol{R}[s][p]$ from $\boldsymbol{R}[s][1 : N_U - 1]$ as *Parasite*;
7: Select any $U_q = \boldsymbol{R}[s][q]$ from $\boldsymbol{R}[s][p : N_U]$ as *Parasite*.
8: Construct four sets of inputs for PNN sub-networks, as $(x_{s,i}^S,\ S_s), (x_{s,i}^S,\ S_r), (x_{s,i}^S,\ U_p), (x_{s,i}^S,\ U_q)$.
9: Run PNN to
10: obtain the $\boldsymbol{\chi}_{s,i}^S$ of instance $x_{s,i}^S$;
11: obtain the corresponding prototypes $\boldsymbol{\varphi}_s^S, \boldsymbol{\varphi}_r^S, \boldsymbol{\varphi}_p^U, \boldsymbol{\varphi}_q^U$;
12: calculate $Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_s^S)$, $Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_r^S)$, $Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_p^U)$, $Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_q^U)$.
13: Minimize the Joint energy function in Eq. 3.
14: **end for**
15: **end for**

The instances with the same type will cluster around a single prototype in the shared semantic space, whereas they are far away from other type prototypes. Meanwhile, the more similar types are distributed closer in the space [15, 16]. Therefore, we determine the semantic distance $Dis(\cdot)$ between the feature representations $\boldsymbol{\chi}$ and the type prototypes $\boldsymbol{\varphi}$. Here, the semantic distance is a quantification of the mapping $f(\cdot)$. The smaller the distance, the better the mapping fit.

Furthermore, we can establish the following assumptions of **premise**: (1) Given any relation type R_1 and corresponding instance x, it should be sure that the semantic distance between x and R_1 is the smallest (or even 0), compared with the distance between x and any other types. (2) For arbitrarily given type R_2 and type R_3 ($R_1 \neq R_2 \neq R_3$), if the semantic distance between R_2 and R_1 is smaller than that between R_3 and R_1, the semantic distance between R_2 and x should be smaller than that between R_3 and x.

The above premises imply the association of the relation types in the shared semantic space. According to this correlation, we can create annotations for unseen types (*Parasite*) by considering the instances of seen types as *Host*, just like "Parasitism". Algorithm 1 (lines 1 to 8) shows the process of data creation. Then, we train the PNN model to learn the unseen types' distributions.

Once the model is optimized to master the relation extraction task, we determine the possible relation that a test instance x_j^U may represent, if any. The top ranked prediction from the candidate

predicted types $\boldsymbol{U}$, denoted as $C(x_j^U, 1)$, is given by:

$$C(x_j^U, 1) = argmax\ Dis(\boldsymbol{\chi}_j^U, \boldsymbol{\varphi}_u^U),\ \ u = 1, 2, \ldots, N_U \tag{1}$$

Moreover, $C(x_j^U, K)$ denotes the K^{th} most probable relation type predicted for x_j^U.

3.2 Model Architecture

As shown in Figure 1, the PNN consists of four sub-networks that accept distinct inputs but are then joined by a joint energy function.

To fit the unseen relation types, the parameters between the sub-networks are tied, that is, each network computes the same metric on a shared workbench (shared by *Host* and *Parasite*). Weight tying guarantees that (1) two instances of the identical classes cannot be mapped by their respective networks to very different locations in the semantic space, (2) the feature that learned from the sub-network is category shared interest and not the sample private.

Input Representation. The sub-network takes as input one piece of text and a relation type prototype, the text contains the head and tail entities of a candidate relation. We transform the instance x into its distributed representation $\boldsymbol{x}$ by adopting triple embeddings $\{\boldsymbol{x}^w, \boldsymbol{x}^c, \boldsymbol{x}^p\}$. The $\boldsymbol{x}^w$ denotes the word embedding obtained from the pre-trained corpus. To deal with unregistered words, we use a convolutional neural network to encode character-level information of a word into its character embedding $\boldsymbol{x}^c$, as [8] doing. The $\boldsymbol{x}^p$ represents the position embedding to specify entity pairs. Similarly to [7], it is defined as the combination of the relative distances from the current word to head or tail entities.

Relation Type Prototype. We achieve the vector representation of each relation type into the semantic space to serve as the prototype $\boldsymbol{\varphi}$, with the word embeddings of type labels' names. Word vectors capture distributional similarities from a large text corpus. Each prototype is the average of word embeddings of the core words (i.e. nouns, adjectives, etc., except prepositions, conjunctions) in its label name.

Learning Feature Representation from Text. The sample text has latent feature information which is category-invariant. We feed the triple embeddings of each instance into the bidirectional Ordered Neurons Long Short-Term Memory Network (ON-LSTM) [14] to encode the feature representation $\boldsymbol{\chi}$. The ON-LSTM performs tree-like syntactic structure composition operations on a sentence without destroying its sequence form. It can learn temporal semantics, meanwhile, capture potential syntactic information involved in natural language. Notably, the syntax is important for relation extraction to acquire the associations among entities and relational phrases. For example, if the head entity is the subject of a relation phrase, the tail entity is likely to be the object of the phrase.

Joint Energy Function. As described in Algorithm 1, each sub-network produces a semantic distance metric. They interact with each other and then joint together. In detail, we establish a series of Trunks, shaped like $\{Branch_1, Branch_2\}$, including $\{Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_s^S),\ Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_r^S)\}$, $\{Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_s^S), Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_p^U)\}$, and $\{Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_p^U), Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_q^U)\}$. According to the premises mentioned above, to compare the semantic distance between the two in each trunk, we are motivated by the triplet loss [12], where we set the σ function to ensure that the $Branch_1$ is smaller than the $Branch_2$ by at least a margin m, as,

$$Branch_1 + m \le Branch_2, \quad \forall\ Trunk(Branch_1,\ Branch_2). \tag{2}$$

Thus, the Joint Energy Function is defined as,

$$\mathcal{L}(\Theta) = \sum_{s=1}^{N_S} \sum_{i=1}^{|\mathbb{D}_s^S|} (\Gamma \mid \boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_s^S,\ \boldsymbol{\varphi}_r^S,\ \boldsymbol{\varphi}_p^U,\ \boldsymbol{\varphi}_q^U; \Theta), \tag{3}$$

$$\begin{gathered} max(Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_s^S) - Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_r^S) + m_1, 0) + \\ \Gamma = \beta\ max(Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_s^S) - Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_p^U) + m_2, 0) + , \\ \gamma\ max(Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_p^U) - Dis(\boldsymbol{\chi}_{s,i}^S, \boldsymbol{\varphi}_q^U) + m_3, 0) \end{gathered} \tag{4}$$

where we employ the cosine distance (within the range [0, 2]), β and γ are the trade-off parameters.

4 EXPERIMENTS

4.1 Settings

Dataset. We evaluate models using the relation extraction dataset of [6]. It consists of 120 relation types from the knowledge base Wikidata. We use the positive labeled relation instances in this dataset. There are 225,060 samples. By applying a similar process to [6] and [10], we randomly select 24 classes as a testing set, 10 classes as the dev set, and the rest as the training set. The results reported for each experiment are the average taken over five runs with independent random initializations. Given different thresholds regarding distance, we can measure the precision (P), recall (R) and F1 of the results. We report the optimal values.

Hyperparameters. We implement the neural network by using the Keras library. The word embedding is from the GloVe [2]. The character embedding is initialized randomly. Parameter optimization is performed with Adam optimizer. To mitigate overfitting, we apply the dropout and early-stopping methods. Besides, we set m_1=0.1, m_2=0.1, m_3=0.08, β= γ=1.

Comparison Systems. We experiment with several variants of the PNN. We compare the bidirectional ON-LSTM to the bidirectional LSTM. We also verify the choice of distance, including logistic regression probability (*LR*), euclidean distance (*EU*), and cosine distance (*COS*). We compare our PNN-based systems to the external systems. *NaiveMAP* learns the mapping between the samples and seen types, by using the single mapping distance as loss simply (*Single*) [13], or by adopting siamese network with triplet loss (*Triplet*) [3]. Levy et al. [6] extract relation via reading comprehension, by using different descriptions for relation types (i.e., *NL* - the label's name, *SQ* - only a single question template per relation type, *MQ* - multiple questions, and *QE* - an ensemble learning way). Obamuyide et al. [10] formulate the ZRE task as a textual entailment problem, where *TE* transforms external entailment corpus for training, and *MD* represents training with manual annotation.

4.2 Results and Analysis

Performance of Each Method. Table 1 shows the results of several direct- or indirect- trick models. As for the PNN-based models, the *ON-LSTM* and *COS* are the best partners, forming our ultimate model. The choice of distance metric is important, where the cosine distance can greatly improve the effect of a PNN. The *ON-LSTM* is

[2]When these embeddings are used for relation type prototypes, we do not fine-tune them during training.

Table 1: The performance of the PNN-based models and external systems.

Models				P	R	F1
direct -trick	NaiveMAP	Single		8.97	8.97	8.97
		Triplet		48.59	48.55	48.57
	PNN	ON-LSTM	LR	43.67	43.50	43.58
		LSTM	COS	45.87	45.82	45.85
		ON-LSTM	EU	52.80	52.79	52.80
		ON-LSTM	COS	**54.16**	**54.00**	**54.08**
		NL		40.50	28.56	33.40
indirect -trick	Levy et al. [6]	SQ		37.18	31.24	33.90
		MQ		43.61	36.45	39.61
		QE		45.85	37.44	41.11
	Obamuyide et al. [10]	TE		-	-	44.38
		MD		-	-	64.78

Table 2: The effects of our model trained with varying number of seen relation types. The hits@K represents the F1 of correct extractions ranked in the top K in eq. 1.

Proportion	Hit@K			
	K=1	K=2	K=3	K=5
22/86 (25%)	28.80	38.16	43.92	54.76
43/86 (50%)	40.05	50.98	55.61	62.41
65/86 (75%)	48.00	60.12	64.61	69.56
86/86 (100%)	54.08	63.99	67.67	71.94

about 8% better than *LSTM* in F1. It shows that the potential syntactic information captured by the ON-LSTM is very useful for relation extraction. However, the LSTM explicitly imposes a chain structure that can not discern the hierarchical syntactic information.

As for the models based on the direct-trick, our ultimate model greatly outperforms others. As expected, *NaiveMAP+single* is insufficient in a zero-shot setting. Compared with the *NaiveMAP +triplet*, our model improves the effect. The *NaiveMAP* models cannot learn the semantic distributions of unseen relation types, and our model has this advantage. Our model achieves effect improvement more than 20% than the model of Levy et al., when there are no manual annotations or additional resources.

As for the systems based on the indirect-trick, most of them are inferior to our model, although they utilize lots of external resources (including transfer models, manual annotations). The results of Levy et al. and Obamuyide et al. indicate that the more quantity and higher quality of annotation information, the better the models will perform. Thus, these indirect-trick methods are constrained by extra annotation effort.

Analyze the Impact of Training Set Size. Table 2 shows the results of our model after being trained with varying proportions of seen types. As the seen types in the training set increase, the performance of unseen relation extraction will become better. The reason may be that the diversity of training set reduces the tendency of the model to overfit seen types. In addition, most of the correct extractions appear in the front part (i.e. top K=5) of the candidate type ranks. It proves the validity of our premises, where the semantic distance between each sample and its corresponding prototype tends to be minimal.

Table 3: Examples of unseen relation type "father".

father (0)	named_after (0.361)	employer (0.644)	chairperson (0.889)
[Samuel Dirksz van Hoogstraten]$_{entity}$ trained first with his father [Dirk van Hoogstraten]$_{entity}$ and stayed in Dordrecht until about 1640.			
0.403	0.562	0.726	1.119
[Bertrade de Montfort]$_{entity}$ was the daughter of [Simon I de Montfort]$_{entity}$ and Agnes , Countess of Evreux.			
0.362	0.531	0.728	1.122

Case Study. We sample an unseen relation type "father" and its corresponding instances from the test set. The 1^{st} row of Table 3 presents several unseen types and their respective semantic distance from the target type "father". The 3^{rd} and 5^{th} rows of Table 3 show the semantic distances between each text instance and the relation types. The distance between each instance and the target type is minimum. Besides, the smaller the semantic distance between a relation type and the target type is, the smaller the semantic distance between it and the instance corresponding to the target type can be. Therefore, the conclusion of the test results is consistent with the premises mentioned above.

5 CONCLUSION

In this paper, we propose a general zero-shot relation extraction framework via the direct-trick to identify unseen relations. Furthermore, we propose the parasitic neural network. Inspired by parasitism, it owns a tied network structure and expands annotations automatically for unseen relation types to learn their distributions. The experiment performance is conspicuous. We will release the source code when the paper is openly available.

REFERENCES

[1] Orpaz Goldstein. 2018. *Zero-Shot relation extraction from word embeddings.* Ph.D. Dissertation. UCLA.

[2] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *EMNLP*.

[3] Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-Shot Transfer Learning for Event Extraction. In *ACL*. 2160–2170.

[4] Stanley Kok and Pedro Domingos. 2008. Extracting semantic networks from text via relational clustering. In *ECML*. 624–639.

[5] Shantanu Kumar. 2017. A survey of deep learning methods for relation extraction. *arXiv preprint arXiv:1705.03645* (2017).

[6] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-Shot Relation Extraction via Reading Comprehension. In *CoNLL*. 333–342.

[7] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*. 2124–2133.

[8] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *ACL*. 1064–1074.

[9] Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. 2012. Ensemble semantics for large-scale unsupervised relation extraction. In *EMNLP*. 1027–1037.

[10] Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot Relation Classification as Textual Entailment. In *EMNLP Workshop on FEVER*. 72–78.

[11] Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. Description-Based Zero-shot Fine-Grained Entity Typing. In *NAACL*. 807–814.

[12] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*. 815–823.

[13] Haseeb Shah, Johannes Villmow, Adrian Ulges, Ulrich Schwanecke, and Faisal Shafait. 2019. An Open-World Extension to Knowledge Graph Completion Models. In *AAAI*.

[14] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In

ICLR.
[15] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*. 4077–4087.
[16] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *NeurIPS*. 935–943.
[17] Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. 2019. A survey of zero-shot learning: Settings, methods, and applications. *ACM TIST* 10, 2 (2019), 13.
[18] Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. 2009. Unsupervised relation extraction by mining Wikipedia texts using information from the web. In *ACL*. 1021–1029.