

# A KG-based Enhancement Framework for Fact Checking Using Category Information

Shuai Wang<sup>1,2</sup>    Lei Wang<sup>1\*</sup>    Wenji Mao<sup>1,2</sup>

<sup>1</sup>State Key Laboratory of Management and Control for Complex Systems  
Institute of Automation, Chinese Academy of Sciences

<sup>2</sup>School of Artificial Intelligence, University of Chinese Academy of Sciences  
{wangshuai2017, l.wang, wenji.mao}@ia.ac.cn

**Abstract**—The massive spread of false information has brought about severe security-related problems to individuals and society. To debunk misinformation automatically, fact checking has become an important task that aims at retrieving evidence from external sources to verify the truthfulness of a given claim. As knowledge graph (KG) is a classic external source for retrieving relevant evidence. Previous methods typically check a claim by making inferences over it. Entity category information can be utilized to strengthen both the learning and verification process. However, this information was largely ignored in previous research. To make better use of the category information, in this paper, we propose a category-based framework for improving the performance of fact checking with KGs. We first learn prototypes for each category as their representatives, and then propose a prototype-based learning technique for effectively modeling the entity dependency in KG. We further develop a prototype matching technique to explore the category-level relations between head and tail entities for more robust verification. Experimental results on two benchmark datasets and a real-world dataset show that our framework can significantly improve the reasoning abilities of KG reasoning methods on Fact Checking task.

**Index Terms**—fact checking, category-based learning enhancement and verification, knowledge graph

## I. INTRODUCTION

With the rapid development of online contents, massive misinformation disseminates on the web instantly and globally. As a large portion of the misinformation contains false knowledge, knowledge-based rumor information causes severe negative impacts on individuals and society. False knowledge affects science and societal information, undermines trust in science and the capacity of individuals to make evidence-informed choices, and consequently brings about serious security-related problems.

To debunk misinformation automatically, a variety of fact checking methods have been developed [1–3], aiming at retrieving evidence from external sources to verify the truthfulness of a given claim. Since knowledge graph is a structured knowledge base which contains rich high-quality facts, it is commonly used as an external resource to detect the claim typically represented in the triple form (*head entity, relation, tail entity*).

Existing methods on automatic fact checking using KGs falls into two main groups, rule-based methods and

embedding-based methods. Rule-based methods mine the patterns (i.e. paths) between a head entity  $h$  and a tail entity  $t$  to predict whether there is a relation  $h \xrightarrow{r} t$  in KGs [4–7], which regard fact checking as a link prediction task. Due to the intrinsic incompleteness of KGs, rule-based methods cannot always find effective paths to support fact checking [8]. Embedding-based methods embed entities and relations into continuous vector spaces, and then measure the plausibility of facts by matching the latent semantics of entities and relations in the vector spaces [9, 10]. It seems that embedding-based method can overcome the issue of missing paths, however, robust embeddings cannot always be learned for every entity and relation in KGs due to the long tail, which may cause the overfitting problem and the fragileness of the model for uncommon entities or relations.

To learn robust representations of entities and relations for efficient fact checking, a feasible way is to make full use of entity categories. Intuitively, entities of the same category have relatively closer semantic representations and properties compared to those belonging to different categories. Thus, category information can be utilized as regularizer to enhance the robustness of embeddings. In addition, this information can also facilitate the verification of claims. For example, given the fact (*apple, improve, digestive system*) in KG, in which *apple* belongs to *fruit* category and *digestive system* belongs to *digestion* category, many other entities in *fruit* category have similar tails belonging to *digestion*. To verify the veracity of a claim triple (*beef, improve, digestive system*), in which *beef* belongs to *meat* category, because few entities in *meat* category have similar tails belonging to *digestion*, the mismatching in *improving digestive system* with *beef* should be much greater than that with an entity in *fruit* category.

To use category information of entity for enhancing the performance of fact checking, we consider to develop prototype-based framework for robust fact checking. Although prototype learning was previously proposed to find the nearest class prototypes for pattern classification [11–14], our focus is on developing prototype-based framework to augment learning and verification for fact checking, realized by an enhanced graph attention aggregator and a prototype-based matching technique.

In this paper, we propose a Category-Based Learning

\* Corresponding Author

Enhancement and Verification framework (CBLEV) for fact checking with knowledge graphs. To leverage entity category information, we first acquire prototypes for each category, which are the representatives of their corresponding categories. To enrich the semantic representations of entities, we propose a prototype-based learning technique to aggregate category-relevant neighboring information. We further develop a prototype-based matching technique to capture relations of head and tail entity categories for more robust fact verification. In our CBLEV, the prototype-based learning enhancement and verification modules work jointly for effective fact checking.

Our work has made the following contributions:

- 1) We propose a category-based fact checking framework, which can effectively utilize category information to improve the performance.
- 2) We develop a prototype-based learning technique and a prototype matching mechanism to enrich entity representation and better support fact verification in a claim triple.
- 3) Base on two benchmark datasets and a real-world dataset, we conduct experiments to verify the effectiveness of our framework.

## II. RELATED WORK

Fact checking using KGs aims to verify the truthfulness of claim triples by extracting evidence from the KGs. Existing fact checking methods mainly include rule-based methods and embedding-based methods. Rule-based methods leverage paths between head and tail entities as evidence in some cases [4–7]. Ciampaglia et al. [4] proposed the first work to computationally gauge the support for claims by mining the path linking between head and tail entities in a KG. To consider multiple paths, Shiralkar et al. [5] extended the work in [4] by employing a flow network, and they showed that fact checking amounted to finding a “knowledge stream” connecting the head and tail entities. To effectively discover discriminative paths, Shi et al. [6] made use of generalized notion of entities by replacing the specific entities by their type-labels, and then defined some mined rules to extract features among paths. The work performed the best on several real-world datasets. Fionda et al. [7] built a schema graph to generate candidate evidence patterns, together with developed various optimizations and RDFS inference rules.

As rule-based methods cannot always find effective paths to support fact checking due to the intrinsic incompleteness of KGs, embedding-based methods are proposed to alleviate the incompleteness issue. Embedding-based methods map entities and relations into continuous vector space and compute the correctness of an unseen triple in the vector space. Dong et al. [10] proposed a region-based embedding approach to solve the triple verification from a geometric view. Their approach used fine-grained type chains and verified triples according to whether the tail entity was located in the head entity’s subspace. However, their approach could not be extend to larger KGs with more relations. Pan et al. [9] proposed Dual TransE that extended TransE by perform it on two KGs (a true

KG and a false one) to get two bias score, and then evaluate the claim triples by comparing these two bias. Dual TransE achieves the best due to the strength of the embedding based methods.

In addition, for KG reasoning task, there have been a number of KG embedding methods developed to rank candidate entities, for the purpose of knowledge graph completion. The most representative methods include TransE [8], TKRL [15], DistMult [16], A2N [17] and ConvKB [18]. Among them, A2N achieves the state-of-the-art results on KG completion task. To make use of the achievement of KG embedding methods for enhancing fact checking task, in this paper, we focus on designing a learning enhancement and checking framework on top of KG embeddings. Our proposed framework also utilizes category information to enhance the semantic representation of embedding learning and support verification. Below we present our framework in detail.

## III. PROBLEM FORMULATION

Given a claim in the triple form  $(h, r, t)$  and a relevant knowledge graph  $\mathcal{G}$ , a fact checking model  $f(\cdot)$  aims at verifying the truthfulness of the given claim triple by reasoning over the KG, which contains the relevant evidences for fact verification.

## IV. PROPOSED MODEL

We propose an end-to-end category-based learning enhancement and verification framework CBLEV for fact checking. Figure 1 gives an overview of our CBLEV framework, which consist two main modules: (1) a *Category-based Learning Enhancement* module to augment entity embeddings by composing the information from adjacent entities in KG; (2) a *Category-based Checking* module for scoring the truth values of triples via prototype enhanced semantic matching. In order to apply our framework for fact checking task, we first introduce KG embedding and prototype learning, and then present the two above modules in the following sections.

### A. Embedding KG components and Learning Prototypes

Our CBLEV is the category based framework to enhance learning and verification. Originally, we use KG embedding methods to embedded the KG component (entities and relations) in vector space. Thus one advantage of our framework is that it can be easily incorporated with different KG embedding methods to facilitate fact checking, by adding their loss functions to our final loss. We get the KG embeddings by using other KG embedding methods, such as, TransE, DistMult and so on. The embeddings of entities and relations in KG are optimized jointly with all the parameters in our framework.

To leverage the rich information contained in entity category, we use prototypes to represent each category. To achieve this, we build learnable *prototype vectors* as the representatives to encode the category information. The prototype vectors are learnable parameters and they are optimized together with other parameters in the training process.

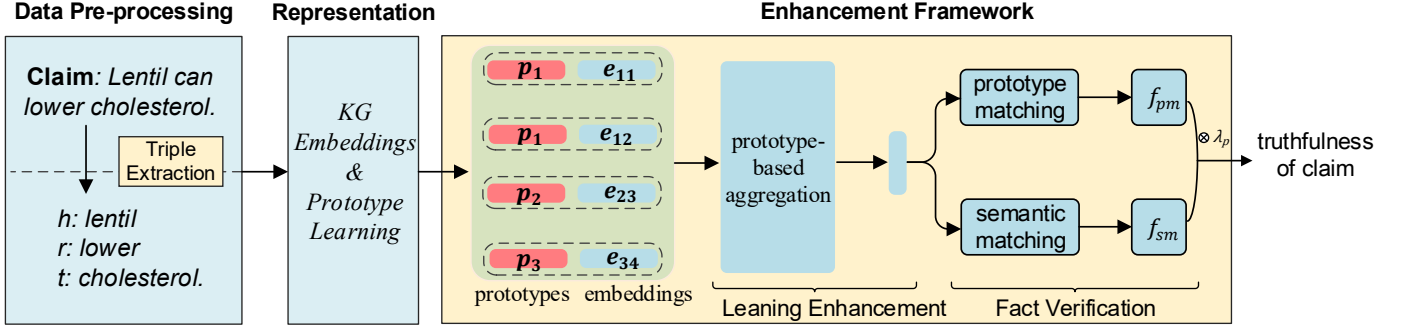


Fig. 1: Overview of our proposed end-to-end category-based learning enhancement and verification framework.

Specifically, let  $\mathcal{C}$  denote the set of category in a KG  $\mathcal{G}$ , entity  $e_{ij}$  belongs to the a category  $c_i \in \mathcal{C}$  with a corresponding prototype vector  $\mathbf{p}_i$ . We pull the entity embeddings  $\mathbf{e}_{ij}$  and its prototypes vector  $\mathbf{p}_i$  closer to each other by minimize the 2-norm distance between them:

$$\mathcal{L}_{pl} = \sum_{e_{ij} \in \mathcal{E}} \|\mathbf{p}_i - \mathbf{e}_{ij}\|_2 \quad (1)$$

### B. Category-based Learning Enhancement

The semantic information contained in the entity’s relational neighbors can be used to enrich entity representation and support the fact checking. Thus we employ the associated entities and their prototype information to extend entity embeddings.

Specifically, given a head entity embedding  $\mathbf{h}_i$  and one tail entity embedding  $\mathbf{t}_j$  with the its relation  $\mathbf{r}_j$  (i.e.,  $(h_i, r_j, t_j)$  is a triple fact in  $\mathcal{G}$ ), we design a prototype-based aggregation technique(called *PLe*), which uses attention mechanism to aggregate embeddings and prototypes from category-relevant adjacent entities. We extend vanilla graph attention [19] to incorporate prototype information and develop two independent attention mechanisms to effectively compose the embeddings and prototypes simultaneously:

$$\tilde{\mathbf{h}}_i = \mathbf{W}_a \sum_{j \in \mathcal{N}(h_i)} (\alpha_{ij} \mathbf{t}_j \parallel \zeta_{ij} \mathbf{p}_j) + b \quad (2)$$

where  $\alpha_{ij}$  and  $\zeta_{ij}$  are the attention scores for computing tails’ embeddings and prototypes respectively,  $\parallel$  is the concatenating operation,  $\mathcal{N}(h_i)$  denotes the adjacent triple set of  $h_i$  and  $\mathbf{W}_a$  is the projection matrix. We develop the attention mechanism to aggregate the representations of adjacent nodes:

$$\alpha_{ij} = \text{Softmax}(\mathbf{W}_t \text{ReLU}(\mathbf{W} \mathbf{h}_i \parallel \mathbf{W}(\mathbf{t}_j \odot \mathbf{r}_j))) \quad (3)$$

where  $\odot$  is the element-wise multiplication to model the interaction between relation and the corresponding tail entity, and  $\mathbf{W}$  is a shared linear transformation matrix to improve the expressiveness. The projection matrix  $\mathbf{W}_t$  along with a ReLU nonlinearity is used for better modeling the relationship between head entity and its relations.

For aggregating prototypes, we then use a two-layer feedforward network  $\text{FeedForward}(\cdot)$  to better model the relationship between head entity prototype and tail entity prototype:

$$\zeta_{ij} = \text{Softmax}(\text{FeedForward}(\mathbf{p}_i \parallel \mathbf{p}_j)) \quad (4)$$

Meanwhile, to avoid weakening the semantic embedding  $\mathbf{h}_i$  of an original entity, we concatenate it with the synthetic representations  $\tilde{\mathbf{h}}_i$  and use a projecting matrix to obtain the final representation:

$$\hat{\mathbf{h}}_i = \text{ReLU}(\mathbf{W}_h(\tilde{\mathbf{h}}_i \parallel \mathbf{h}_i)) \quad (5)$$

Similarly, to learn the generalized embeddings for all the entities in KG, we also augment the embeddings of tail entities using head entities’ information in the same way.

### C. Category-based Fact Verification

Our model combines the detailed semantic matching with more general prototype matching for checking the truthfulness of the claim triple. As the CBLEV framework can be effectively incorporated with other KG embedding methods, these methods typically design various scoring functions to measure the plausibility of a claim triple  $f_{SM}(\mathbf{h}, \mathbf{r}, \mathbf{t})$ .

In addition, the dependency relationships of head and tail entities in category level can be severed as the more general relations, which are the discriminative information to support verification. Thus we develop a prototype matching technique to model this relationship to facilitate the fact checking. As each dimension of embeddings represents the relation-specific attributes of entities [20], inspired by this, we construct the prototype matching by modeling entries at the same dimension of the prototypes:

$$f_{PM} = \mathbf{p}_h^\top \text{diag}(\mathbf{p}_r) \mathbf{p}_t \quad (6)$$

where  $\mathbf{p}_h$  and  $\mathbf{p}_t$  are the corresponding prototypes of  $h$  and  $t$ ,  $\mathbf{p}_r$  is the constructed relation between two prototypes, and  $\text{diag}(\mathbf{p}_r)$  is a diagonal matrix formed by the elements in  $\mathbf{p}_r$ . We add the scores produced by the above two matching techniques as the final score, which can be represented as:

$$f(h, r, t) = f_{SM} + \lambda_p f_{PM} \quad (7)$$

where  $\lambda_p$  is the trade-off parameter.

TABLE I: Categories of tail entities in *FOOD*

lung, hair, spleen, stomach, kidney, heart, liver, bone, brain, eye, skin, cancer, urine, cardiovascular, weight loss, immunity, muscle, fiber, sleep, digestion
肺, 发, 脾, 胃, 肾, 心, 肝, 骨, 脑, 眼, 肤, 癌, 尿, 血管, 减 肥, 免疫, 肌, 纤维, 睡眠, 消化

TABLE II: Statistics of the three KG datasets.

Dataset	#entity	#relation	#train	#valid	#test
FOOD	4,192	86	26,767	1,986	1,080
FB15K-237	14,505	237	272,115	20,000	20,000
FB15K	14,951	1,345	483,142	50,000	59,071

The objective of the verification module is to minimize the regularized logistic loss, which encourages  $f(\cdot)$  to produce a higher score for a true triple than that for a false one:

$$\mathcal{L}_{fc} = \sum_{(h,r,t) \in \Delta} \log(1 + \exp(-f(h,r,t))) + \sum_{(h',r',t') \in \Delta'} \log(1 + \exp(f(h',r',t'))) \quad (8)$$

where  $\Delta$  and  $\Delta'$  are the set of correct triples and that of incorrect triples, respectively.

The final loss to optimize the category-based enhancement framework can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{fc} + \lambda \mathcal{L}_{pl} \quad (9)$$

where  $\lambda$  is the trade-off parameter.

## V. EXPERIMENTS

In this section, we incorporate CBLEV with the representative KG embedding methods to validate our framework by comparing with the previous fact checking methods.

### A. Experimental Setup

1) *Datasets*: We conduct experiments on three datasets, a dataset in food (*FOOD*) and two benchmark KG datasets *FB15K* [8] and *FB15K-237*. The *FB15K* and *FB15K-237* are two commonly used KG datasets. *FB15K* is a dense graph extracted from *Freebase*, and *FB15K-237* is constructed by deleting all the inverse triples in *FB15K* to form a sparser graph. For *FB15K* and *FB15K-237*, we use the entity categories collected by [15]. For the *FOOD* dataset we developed, we assign a category in Table I to each tail entity that contains the effect of food, according to whether the tail entity contains the category word (in Chinese). For example, *digestive system* (消化系统) belongs to *digestion* (消化) category.

Following [15], we construct the negative triples for all the datasets. The statistics of the three KGs are shown in Table II.

2) *Hyperparameters*: We use 100-dimensional KG embeddings on *FB15K* and *FB15K-237* and 30-dimensional embeddings on *FOOD*. The trade-off parameter  $\lambda$  and  $\lambda_p$  are set to 0.1 and 1 respectively. We train our model with 0.01 learning rate and 100 batch size using AdaGrad. We set dropout rate to 0.5 and adopt L2-norm for regularization.

3) *Comparative Methods*: Although KG embedding methods are designed to rank candidate entities for knowledge graph completion, we incorporate our framework CBLEV with the representative KG embedding methods as the starting point, and compare them with the existing KG based fact checking methods. The five representative KG embedding methods we incorporated with include:

- 1) TransE [8] is the most representative translational distance model, which defines its scoring function as the distance between  $h + r$  and  $t$ .
- 2) DistMult [16] is a representative semantic matching method that predict a truth score for a triple by using a bi-linear function.
- 3) TKRL [15] takes advantage of entity categories by considering categories as projection matrices for entities to improve their representations.
- 4) ConvKB [18] employs a convolutional neural network to effectively capture the global relationships between entities and relation.
- 5) A2N [17] selectively composes relevant graph neighbors with attention weights computed by semantic matching model to enhance the entity representations.

The existing fact checking methods we compared with include:

- 1) KnowStream [5] regards the fact checking as a network-flow problem, which utilizes all the paths between head and tail entities and computes the truth score using Dijkstra's algorithm.
- 2) PredPath [6] defines some mined rules to extract features from paths, and predicts whether there is a link between head and tail entities.
- 3) Dual TransE [9] is a TransE based model, which predicts the plausibility score of the claim triple by using the bias calculated by TransE.

For KnowStream and PredPath, we use publicly available codes in [5] and tune the hyperparameters based on what are reported in [5] and [6] respectively. We reimplement the Dual TransE [9] method and tune the hyperparameters in same way. We select accuracy and  $F_1$  as metrics to evaluate our framework.

### B. Experimental Results

1) *Main Results*: Table III gives the experimental results on fact checking. It can be seen from the table that our CBLEV framework with the five representative KG embedding methods outperform all the fact checking methods. TKRL+CBLEV and A2N+CBLEV gets the highest accuracy and  $F_1$  values respectively on *FB15K*. ConvKB+CBLEV and A2N+CBLEV are the best performing models on *FB15K-237* and *FOOD* respectively, and they perform generally well

TABLE III: Experimental results on fact checking by different methods.

Method	FB15K		FB15K-237		FOOD	
	Acc.	$F_1$	Acc.	$F_1$	Acc.	$F_1$
KnowStream [5]	0.835	0.829	0.801	0.729	0.725	0.734
PredPath [6]	0.813	0.806	0.802	0.806	0.728	0.746
Dual TransE [9]	0.912	0.903	0.874	0.875	0.764	0.726
TransE+CBLEV	0.923	0.911	0.892	0.881	0.788	0.780
DistMult+CBLEV	0.944	0.934	0.914	0.907	0.820	0.811
TKRL+CBLEV	0.949	0.935	0.897	0.885	0.814	0.801
ConvKB+CBLEV	0.944	0.933	0.930	0.924	0.831	0.826
A2N+CBLEV	0.946	0.937	0.922	0.918	0.833	0.829

on all the datasets. This demonstrates that the deep learning model ConvKB can effectively model the interrelations between entities and relation, and A2N is beneficial to learn robust representations via dynamically composing information from adjacent nodes. TransE+CBLEV and TKRL+CBLEV are TransE based models. They cannot perform well on *FOOD*, as there are many 1-to-N relations in *FOOD* and TransE based models have some limitations in this case [21].

We can also see that all the methods get relatively good results on *FB15K*. The reason of this is that *FB15K* is a dense knowledge graph and thus the methods can learn generalized embeddings or find effective paths on it. The performances of all the methods decline on *FB15K-237*, as reasoning over this sparser graph is harder. Since the real-world dataset *FOOD* is the sparsest, most methods cannot perform well on this dataset. Nonetheless, A2N+CBLEV and ConvKB+CBLEV can obtain reasonable results and they outperform the three fact checking methods by a large margin. This indicates that our framework is more applicable to the sparse KG compared to other methods. The experimental results demonstrate the effectiveness of our framework for improving the performance.

### C. Ablation Study

We conduct ablation studies to verify the effectiveness of each component in our framework. We incorporate three representative KG embedding methods with CBLEV, including the classical translational distance method TransE, representative semantic matching model DistMult and the deep learning model ConvKB. We construct three variants of CBLEV:

- *−PM*: excluding prototype matching module in fact verification process.
- *−PLE*: excluding prototype-based learning enhancement module from CBLEV.
- *−PLE −PM*: excluding both learning enhancement and prototype matching modules from CBLEV.

The experimental results of the ablation study are given in Table IV. We can see from the table that excluding *PM* or *PLE* from CBLEV will cause performance drop. It can be also seen that the two modules improve the performance more on the sparser KG textitFB15K-237 and *FOOD*. Specifically,

TABLE IV: Experimental results of the ablation study (Here “\*” denotes that the performance significantly drops without the corresponding module(s)).

Method	FB15K		FB15K-237		FOOD	
	Acc.	$F_1$	Acc.	$F_1$	Acc.	$F_1$
TransE+CBLEV	0.923	0.911	0.892	0.881	0.788	0.780
−PM	0.918	0.907	0.883	0.873	0.774	0.767
−PLE	0.914	0.902	0.885	0.877	0.764	0.756
−PLE −PM	0.911	0.901	0.872	0.874	0.757*	0.746*
DistMult+CBLEV	0.945	0.934	0.916	0.909	0.821	0.813
−PM	0.927	0.916	0.902	0.893	0.801	0.794
−PLE	0.914	0.902	0.889	0.884	0.771*	0.765*
−PLE −PM	0.895	0.884	0.878*	0.879*	0.750*	0.753*
ConvKB+CBLEV	0.944	0.933	0.930	0.924	0.831	0.826
−PM	0.940	0.930	0.926	0.919	0.821	0.816
−PLE	0.936	0.931	0.924	0.916	0.811	0.809
−PLE −PM	0.934	0.931	0.917	0.911	0.804	0.803

TABLE V: Illustrative categories of head entities and their most associative categories of tail entities (Top 5 tail categories based on the prototype matching scores).

head categories	tail categories
meat 肉类	(strengthen) immunity, muscle, brain, skin, eye 免疫, 肌, 脑, 肤, 眼
berry 浆果	(prevent) cancer, heart, digestion, urine, fiber 癌, 心, 消化, 尿, 纤维
dairy 奶制品	bone, digestion, heart, cardiovascular, immunity 骨, 消化, 心, 血管, 免疫
seafood 海产品	heart, eye, sleep, liver, brain 心, 眼, 睡眠, 肝, 脑
leaves vegetables 叶菜类	weight loss, fiber, cardiovascular, stomach, skin 减肥, 纤维, 血管, 胃, 肤

the module *PLE* can significantly enhance the verification on *FOOD*. The reason of this is that the adjacent entities can be seen as attribute nodes, which are beneficial for enriching entity representations. The performance of ConvKB+CBLEV drops slightly without the modules, demonstrating the effective expressive power of deep learning model. The experimental results on the ablation study further verify the usefulness of each component in our framework.

### D. Case Study: Effect of Prototype Matching

To show the effect of prototype matching technique, Table V illustrates several head entity categories and the corresponding tail entity categories acquired by prototype matching in Function 6. We select the 5 entity categories with the highest scores. We can see from the table that the head categories are quite compatible with tail categories, which is in line with our common knowledge to effectively support fact checking.

## VI. CONCLUSION

In this paper, we propose a category-based learning enhancement and verification framework for fact checking with knowledge graphs, which can effectively utilize entity category information to improve the performance of fact checking. Our framework learns prototypes for each entity category as their representatives, and then further develops a prototype-based aggregation technique for enriching entity representations and a prototype-based matching technique to facilitate fact verification. Experimental results on two benchmark datasets and a real-world dataset demonstrate the effectiveness of our framework.

## ACKNOWLEDGMENTS

This work is supported in part by the Ministry of Science and Technology of China under Grants #2016QY02D0305, and NSFC under Grants #71621002, #71702181 and #71603253.

## REFERENCES

- [1] A. Vlachos and S. Riedel, “Fact checking: Task definition and dataset construction,” in *Proceedings of Association for Computational Linguistics*, 2014, pp. 18–22.
- [2] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, “Truth of varying shades: Analyzing language in fake news and political fact-checking,” in *Proceedings of Empirical Methods in Natural Language Processing*, 2017, pp. 2931–2937.
- [3] N. Hassan, F. Arslan, C. Li, and M. Tremayne, “Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster,” in *Proceedings of Knowledge Discovery and Data Mining*, 2017, pp. 1803–1812.
- [4] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, “Computational fact checking from knowledge networks,” *PLoS One*, vol. 10, pp. 1–13, 06 2015.
- [5] P. Shiralkar, A. Flammini, F. Menczer, and G. L. Ciampaglia, “Finding streams in knowledge graphs to support fact checking,” in *Proceedings of International Conference on Data Mining*, 2017, pp. 859–864.
- [6] B. Shi and T. Wenginger, “Discriminative predicate path mining for fact checking in knowledge graphs,” *Knowledge-Based Systems*, vol. 104, pp. 123–133, 2016.
- [7] V. Fionda and G. Pirrò, “Fact checking via evidence patterns,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2018, pp. 3755–3761.
- [8] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Proceedings of Neural Information Processing Systems*, 2013, pp. 2787–2795.
- [9] J. Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li, and J. Liu, “Content based fake news detection using knowledge graphs,” in *Proceedings of International Semantic Web Conference*, 2018, pp. 669–683.
- [10] T. Dong, Z. Wang, J. Li, C. Bauckhage, and A. B. Cremers, “Triple classification using regions and fine-grained entity typing,” in *Proceedings of Association for the Advancement of Artificial Intelligence*, vol. 33, 2019, pp. 77–85.
- [11] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *Proceedings of Neural Information Processing Systems*, 2017, pp. 4077–4087.
- [12] Y. Cao, L. Huang, H. Ji, X. Chen, and J. Li, “Bridge text and knowledge by learning multi-prototype entity mention embedding,” in *Proceedings of Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1623–1633.
- [13] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, “Robust classification with convolutional prototype learning,” in *Proceedings of Computer Vision and Pattern Recognition*, 2018, pp. 3474–3482.
- [14] K. Allen, E. Shelhamer, H. Shin, and J. Tenenbaum, “Infinite mixture prototypes for few-shot learning,” in *Proceedings of International Conference on Machine Learning*, 2019, pp. 232–241.
- [15] R. Xie, Z. Liu, and M. Sun, “Representation learning of knowledge graphs with hierarchical types,” in *Proceedings of International Joint Conferences on Artificial Intelligence*, 2016, pp. 2965–2971.
- [16] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, “Embedding entities and relations for learning and inference in knowledge bases,” in *Proceedings of International Conference on Learning Representations*, May 2015.
- [17] T. Bansal, D.-C. Juan, S. Ravi, and A. McCallum, “A2n: Attending to neighbors for knowledge graph inference,” in *Proceedings of Association for Computational Linguistics*, 2019, pp. 4387–4392.
- [18] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, “A novel embedding model for knowledge base completion based on convolutional neural network,” in *Proceedings of North American Chapter of the Association for Computational Linguistics*, Jun. 2018, pp. 327–333.
- [19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” *Proceedings of International Conference on Learning Representations*, 2018.
- [20] D. Q. Nguyen, T. Vu, T. D. Nguyen, D. Q. Nguyen, and D. Phung, “A capsule network-based embedding model for knowledge graph completion and search personalization,” in *Proceedings of Association for Computational Linguistics*, Jun. 2019, pp. 2180–2189.
- [21] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” in *Association for the Advancement of Artificial Intelligence*, 2014.