Uncertain Knowledge Graph Embedding: a Natural and Effective Approach

Shihan Yang\textsuperscript{1}, Rui Tang\textsuperscript{1}, Zhiwei Zhang\textsuperscript{2}, and Guozhong Li\textsuperscript{1}

\textsuperscript{1}Faculty of Management and Economics, Kunming University of Science and Technology, Kunming, 650093, China
\textsuperscript{2}College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China
E-mail: dr.yangsh@kust.edu.cn

Abstract. Uncertain Knowledge Graph (UKG) models uncertainty of the knowledge, which usually models the inherent uncertainty of relation facts in a knowledge base with a confidence value. Embedding such uncertain knowledge represents is still an unresolved challenge, although deterministic Knowledge Graph embedding has been extensively studied recently, which aims at representing entities and relations as vectors in a continuous vector space and preserving semantic information as much as possible. For capturing both structural and uncertainty information of relation facts in the continuous vector space, we propose a simple but effective two-steps approach to Uncertain Knowledge Graph Embedding (UKGE) based on the skip-gram/CBOW model and learning confidence score by Long Short-Term Memories (LSTM) neural network. We show that the embedding techniques achieve results comparing with state-of-the-art approaches on uncertain knowledge graph embedding. The approach achieves the best tradeoff between efficiency and accuracy of UKGE. Because of the simplicity, the method can also handle large size graphs in lower time consumption.

1. Introduction

There are two kinds of Knowledge Graph (KG), graph-structured knowledge bases: (i) Deterministic KG, such as FreeBase [1] and YAGO [2], in which relation facts are deterministic (implied confidence score is 1); (ii) Uncertain KG, such as NELL [3] and ConceptNet [4], in which every relation fact associates with a confidence score between 0 and 1 that represents the trust degree of the relation fact to be true. KG embedding models, which encode entities and relations as low-dimension continuous vectors, are necessary tools for introducing the structured knowledge representations into machine learning field. They try to accurately capture the similarity of entities and preserve the structured semantic information of knowledge bases in this continuous embedding space, and have shown promising results on downstream tasks such as link prediction, question answering, recommendation, and triplet classification. While most of them only focus on capturing deterministic knowledge, it is also critical to obtain and preserve uncertainty semantic information for some reasons. First, uncertainty is the essential part of knowledge and uncertain knowledge representation has great benefits for various applications, such as auto-answering system, and uncertain decision-making. Second, uncertainty enhances inference in knowledge-driven tasks, such as short text understanding in Natural Language Processing (NLP) filed, which often entails interpreting real world concepts that are always
ambiguous or intrinsically vague. However capturing the uncertainty semantic information of Uncertain Knowledge Graph Embedding (UKGE) remains an unresolved problem [5]. There are several challenges on UKGE:

- Need to encode additional confidence information to preserve knowledge uncertainty in the embedding space;
- Properly to estimate the uncertainty of unseen relation facts, instead of regarding them as zero when they do not occur in the KG just like deterministic KG embedding models doing.

To address these issues, we regard every triple (relation fact) as a short sentence with only three words and proposed a simple and effective uncertain knowledge graph embedding model UKG\(_E\) (Uncertain Knowledge Graph simple-effective Embedding), which consists of two steps: words embedding and confidence learning. When treating a relation fact as a sentence, the knowledge base expressed in KG/UKG is a document including all same length triple-sentences, and all entities and relations is combined into a corpus. First step is to learn words embedding from the knowledge base. Each word embedding is a entity or relation embedding, which is the pre-training parts of the approach. The pre-training can lead to high quality embedding as the [6] argued. Second step is to learn confidence scores by LSTM neural network, which can memory the hidden triple-sentence structure information for better confidence predictions. We achieve a better results in a downstream task, link prediction, and better evaluating on the mean squared error of the testing datasets than state of the art. Furthermore, we obtain that with lower training time and less computing power, which means the approach can be used to address very large-size uncertain knowledge graph embedding requirements.

2. Related work

There are several works on handling uncertain semantical information of knowledge graphs. [7] has proposed a probabilistic embedding method KG2E for explicitly modelling the (un)certainties of entities and relations in KGs based on Bayesian framework. This model considers uncertainty of knowledge in a word-level, not in a sentence-level when treating a triple as a sentence. [8] has proposed a embedding uncertain networks method based on matrix factorization, whose model, so-called URGE (UncerRtain Graph Embdedding), generates the embedding for a given graph. Its embedding vectors can carry the proximity information of nodes in the graph. However the model is not specific to uncertain KGs, although it can handle uncertainty of knowledge presented by graphs, so this model only considers the node proximity of the graph, and does not consider explicit relations. [9] has focused on partial order or lattice structure embedding over concept space, and extended the Order Embedding (OE) by probability theory to address some uncertainty of knowledge. It emphasizes the asymmetric structure embedding and probabilistic reasoning over partial order graphs, but the relations in KG is not always partial order. Most of the time, we often consider the general relations in the knowledge graph. [10] argues that there is a lack of methods for quantifying predictive uncertainty in a knowledge graph embedding representation, and has proposed a highly scalability probabilistic model, LIM (Latent Information Model) and LFM (Latent Fact Model) based on neural variational inference, to evaluate predictive uncertainty during link prediction tasks. This model has roughly the same goal as confidence learning, the second step of our model, but our model is more intuitive, concise and faster to execute.

The recent work [5] has argued that there has been no previous work on learning embedding for uncertain KGs before them. They proposed the uncertain knowledge graph embedding problem: Given an uncertain knowledge graph, the embedding model aims to encode entities and relations in a low-dimensional real space in which structure information and confidence scores of relation facts are preserved as much as possible. We follow this problem definition,
furthermore we employ a natural but effective method and get some better results in learning KGs’ confidence than that they reported.

3. Problem definition
Uncertain Knowledge Graph (UKG) associates each relation fact with a confidence score. Formally, let \( l = (s, p, o, c) \) be a relation fact (a triple with confidence) containing a subject (entity) \( s \), a predicate (relation) \( p \), an object (entity) \( o \), and a confidence score \( c \) in a UKG \( K \). For any triple \( (s, p, o, c) \in E \times R \times E \times [0, 1] \), where \( E \) is the set of all entities, \( R \) is the set of all relations, and the confidence score is between 0 and 1, interpreted as a probability of this relation fact to be true. UKG Embedding (UKGE) aims to encode each entity and relation in a low-dimensional vector space in which structure information and confidence scores of facts are preserved. A representation function \( F \) defined as

\[
F : E \cup R \rightarrow \mathbb{R}^d
\]

assigns a vector dimensionality \( d \) to each entity and relation. And a confidence function

\[
C : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, 1]
\]

assigns each relation fact in the embedding space \( l_v = (s_v, p_v, o_v) \) to a real value between 0 and 1, where \( s_v, p_v, o_v \in \mathbb{R}^d \). So, this study focus on discovering simple but fast, effective functions \( F \) and \( C \) in order to preserve many structure information and confidence in the embedding space enough to perform valuable approximate reasoning. we consider following problems:

- Trade-off between generating embedding at a high rate and preserving uncertain information as much as possible;
- Test and valid distributional hypothesis of confidences of relation facts of UKGE;
- Approximately but effectively reason in the embedding space for some domain knowledge bases.

4. Modelling
We propose two-steps approach. First, treating each relation fact without confidence \( l = (s, p, o) \) as a short sentence with three words, we train the entity and relation embedding by means of the skip-gram model or CBOW model [11]; Second, it learns the confidence score of each uncertain relation fact \( c = C(l_v) \) by means of LSTM (long short-term memories) neural network [12, 13] trained with uncertain knowledge bases, i.e., the function \( C \) actually means LSTM.

4.1. The skip-gram model and the CBOW model
In natural language processing (NLP), the continue bag of words model (CBOW) uses continuous distributed representation of the context to predict the current word based on these context. Otherwise, the continue skip-gram model use each current word as an input to predict words within a certain range (window) before and after the current word.

In the skip-gram model [14], combing all entities and relations into a corpus \( E \cup R \), we consider a triple without confidence as a sentence of three words over the corpus, \( T = \{u_s, u_p, u_o\} \), and aim to maximize the average log probability

\[
\frac{1}{3} \sum_{u \in T} \sum_{u' \in T \setminus u} \log p(u' | u)
\]
where the context window is 2 for the words sequence size is always $|T| = 3$. The probability above is defined by the soft maximum function

$$p(u'|u) = \frac{\exp(v^O_x \cdot v^I_{u'})}{\sum_{x \in E \cup R} \exp(v^O_x \cdot v^I_{u})}$$

(4)

where $v^O_x$ and $v^I_x$ are respectively output and input vectors representation of a word $x$. In the actual model training, $x$ selects all members from a batch samples set, and 5 negative samples sampled by corrupting the object strategy.

CBOW, the mirror of the skip-gram model, predicts the current word from context, where the context window is also 2. The optimization aims to minimize the following loss function

$$L = - \sum_{u' \in T \setminus u} \log(p(u|u'))$$

(5)

where $u'$ is in the context of $u \in T$. In the experiment, we found that there was a slight difference between using these two different models. For one dataset, adopting CBOW in the first step, the prediction of confidence has better results; for another dataset, the skip-gram leads to better prediction. We discuss this later in section VI.

4.2. Confidence learning by neural network

Employing a LSTM neural network to evaluate the confidence scoring function (2), we input a sequence of embedding vectors $(s_v, p_v, o_v)$ for each triple $(s, p, o, c) \in K$ into the neural network, and the model outputs a confidence scores $c$ after a dense hidden layer connected to a single output neuron with sigmoid activation function. Adam optimizer [15] is used in the model training and the early-stopping strategy is adopted.

5. Metrics and baseline

In the communities, KGE is always evaluated on link prediction (or knowledge base completion), while achieving high performances on link prediction does not always mean that the embeddings are good, since the reasoning task is often highly dependent on external algorithms [16]. Furthermore, UKGE is KGE with a confidence distribution of all relation facts, in which the inference is far beyond the link prediction. So the mean square error (MSE) and the mean absolute error (MAE) are the simple evaluation criteria between original confidence values and confidence scores computed by the model in test data set. The smaller the value of MSE and MAE are, the better the model is. We treat the relational facts that appear in the knowledge base and the relational facts that do not appear in the same way. In the first step of our model we also evaluated on link prediction, and in the second step only need to evaluate their confidence scoring by means of MSE and MAE. For both steps we consider the runtime of performance, which evaluates on handling total number of triples per second. For inference tasks in UKGE, we evaluate relation facts by the confidence function that outputs scores with $(s_v, p_v, o_v)$ inputs, $c \approx LSTM(s_v, p_v, o_v)$, which is very different from link predictions that evaluated by adding embedding vectors $s_v + p_v \approx o_v$ in KGE, or by a neural network $NN(s_v, p_v) \approx o_v$. For comparing, we select baselines [5] which is state of the art result on uncertain KG embedding and [16] which is state of the art on quickly generating Knowledge Graph embedding.

6. Experiment

$UKGsE$ is implemented in Python 3.6 with the Gensim and PyTorch libraries. Sources, datasets, and execution logging are available online 1. All experiments were carried on the laptop computer with Win10, 8G RAM, and Intel 4-cores CPU @ 2.30GHz.

1 http://github.com/ShihanYang/UKGsE
6.1. Datasets

CN15k is extracted from ConceptNet, and matches the number of nodes with FB15k [17], the widely used benchmark dataset for deterministic Knowledge Graph Embeddings. CN15k is actually a subgraph of the commonsense KG ConceptNet, contains 15,000 entities, 36 relations, and 234,675 uncertain relation facts (delete the duplicate 6,483 triples from total 241,158). The original confidence scores vary from 0.1 to 22, and [5] normalized them into $[0, 1]$ by min-max normalization after bound the confidence values to $[0, 3.0]$. We get the dataset from the website and cut the duplicate samples off.

PPI5k is also from the literature [5]. The Protein-Protein Interaction Knowledge Base STRING [18] labels the interactions between proteins with occurrence probabilities. PPI5k is a subset of STRING, a denser graph with fewer entities but more relation facts than CN15k, contains 271,666 uncertain relation facts for 4,999 entities and 7 relations.

Table 1 shows the overview of these two datasets.

### Table 1. The overview of datasets *

| Dataset | #Ent. | #Rel. | #Facts | Avg(c) | Std(c) |
|---------|-------|-------|--------|--------|--------|
| CN15k   | 15,000 | 36    | 234,675| 0.627  | 0.234  |
| PPI5k   | 4,999  | 7     | 271,666| 0.415  | 0.213  |

* Here #Ent. denotes the number of entities, so do #Rel. and #Facts relations and relation facts. Avg(c) and Std(c) are the average and standard deviation of the confidence values.

6.2. Setup

For the first step, words encoding, regarding each entity and relation as a word and each fact as a sentence, we implement a CBOW/skip-gram model trained by Word2Vec of the natural language processing library Gensim [19] with windows 2 (maximum distance between the current and predicted word within a sentence), dimensionality of the word vectors 64, ignores words with total frequency lower than 1, negative sampling number 5, initial learning rate 0.025, iteration epochs 10, all entities and relations as vocabulary, and all facts as corpus.

For the second step, confidence learning, inputting word vectors of each fact encoded in the first step into the LSTM model with original confidence as labels, outputting a confidence score between 0 and 1, we train LSTM with epochs 100, batch size 64, mean squared error loss function, Adam optimizer, for which we set the exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and the negative triple generated by corrupting the object of each fact.

Both datasets have been cut into three parts at the confidence learning step, 80% for training, 10% for validation, 10% for testing. For evaluating our model’s ability of handling negative triples, negative triples are also generated by corrupting the object of each fact in test dataset.

For hyper-parameters, we select among the following sets of values: embedding dimensionality $ed \in \{64, 100, 128, 200\}$, batch size $bs \in \{64, 128, 256, 512\}$, word2vec strategy $ws \in \{skip-gram, CBOW\}$. LSTM training was stopped using early stopping strategy on the validation set. The best hyper-parameter combinations on PPI5k are $\{ed = 100, bs = 64, ws = CBOW\}$. On CN15K they are $\{ed = 128, bs = 128, ws = skip-gram\}$. We report results based on their best hyper-parameter settings.

Here we explain that different word2vec strategies lead to different confidence prediction effectiveness as that distinct datasets have different ratios of nodes (entities and relations) to facts (samples or triples). As for PPI5k, the ratio equals to about 54.3, which gets more samples to train vectors from their context. For CN15k, the ratio is just 15.6, which is more suitable for

https://github.com/stasl0217/UKGE
Table 2. Runtime and accuracy for embedding relation facts without confidence in KG *

| Dataset | CN15k | PPI5k |
|---------|-------|-------|
| Dimen.  | 64    | 100   | 128   |
| Runtime(s) | 14.75 | 15.18 | 16.35 |
| Triples/s | 14,611 | 14,197 | 13,180 |
| hits@1  | 0.61  | 0.56  | 0.63  |
| hits@3  | 1.03  | 1.01  | 1.1   |
| hits@10 | 2.20  | 2.23  | 2.31  |

* The hits@X rate value is (%), Dimen. dimensionality, and Triples/s the processing rate.

Table 3. Filtered hits@X values (%) for the query* on PPI5k using several different strategies

| Approaches       | hits@1 | hits@3 | hits@10 |
|------------------|--------|--------|---------|
| LSTM+corrupted   | 3.84   | 9.79   | 19.23   |
| LSTM+random      | 1.39   | 4.89   | 10.49   |
| Step_1 of UKGE  | **2.767** | **5.34** | **12.61** |

* The query is like (subject, predicate, ?object).

forecasting context from the node. Moreover we assume that the critical value of this ratio is 50.0, but this needs further experimental verification.

6.3. Runtime

In the model, wording embedding performs very fast but effective enough. Same as the majority of existing literatures done, hits@1, hits@3 and hits@10 of relation facts without confidence in test dataset are shown in table 2. Hits@X is calculated by means of substituting the object of each fact in the testing dataset for all different entities in the corpus. The first step of the approach shows us the results is plausible that the embedding vectors already contains some structural semantical information. The hits@X is not so bad compared to state of the art (results are compared in table 3 with [16]), however the time cost of performance is much lower than others. For dimension 100 embedding vectors on PPI5k dataset, the model just takes 13.00 seconds on a laptop computer without GPU, which trained just 10 epochs with 249,946 facts and corpus size 5,006. The handling rate 17,911 triples per second is also very impressive, and the best rate reported in [16] is 10,887 triples per second. The results means that the approach achieves a better tradeoff between higher efficiency and more accuracy of KG embedding, and that can be used to deal with large-size knowledge bases expressed in KG or UKG.

As for confidence scoring of relation facts on the second phrase of the approach, we learn them by means of LSTM neural network, which is the main time-consuming part of the approach. Table 4 shows the performance details of LSTM trained by means of early stopping strategy with monitor ‘validation loss’, minimal delta 0.0001 and patience 10. We can see that good results are obtained in a short time. On CN15k dataset, the average loss 7.72, which is comparable to the best result (8.61 of UKGErect in table 5), only needs 19 training epochs. On PPI5k dataset, we get the average loss 1.239 after 29 training epochs. It is also know from the table 4 that with the increase of training times, better results can be obtained. The performance rate (triples / s) also reported. As the dimension increases, the processing rate decreases, but the precision will also be improved. So the trade-off between efficiency and accuracy is always a problem worth considering in practical KG-based application. [16] has handled with deterministic KG embedding without confidence scoring, so we compare with the reported results in [5] in the next subsection.
6.4. Confidence prediction

Evaluating confidence prediction by mean squared error error (MSE) and mean absolute error (MAE), we perform it on testing dataset with unseen relation facts. UKG\_s\_E gets the best MSE value 7.71 over 19,293 testing samples on CN15k dataset after 19 epochs training with 128 dimensionality embedding vectors, and gets the third best MSE (with 0.03 difference to the best) 0.98 over 21,720 unseen test samples on PPI5k dataset after 80 epochs training with 100 dimensionality embedding vectors. The results are shown in table 5 compared with URGE, UKGE\_rect and UKGE\_logi reported in [5].

On CN15k dataset, the model gets better results both in MSE and MAE metrics. To get the best results of UKGE\_rect and UKGE\_logi, we train the model more than 100 epochs by running programs they provided, while UKG\_s\_E get smaller MSE values on stopping training at epochs 19 and 80 for two datasets respectively. The UKG\_s\_E get more trade-off between accuracy and time-consumed. We explain why this approach works effectively as following:

- When treating triples as sentences, then the embedding vector trained by Word2vec can include the hidden structural semantic information. The simpler sentence structure is, the easier and the more features of structure are preserved. Here the short triple-sentence is three words sentence (subject, predicate, object), which is simple enough.

- Pre-training is very important and effective to UKGE. Better pre-training vectors get better confidence prediction. Moreover, High quality pre-training can greatly improve the convergence speed of confidence learning.

- It is better to consider a relation fact as a whole (a sentence) than to train with head + relation ≈ tail as the objective function.

7. Conclusion and future work

A two-steps approach can effectively and quickly address the uncertain knowledge graph embedding, which includes link prediction and confidence score prediction. The method achieve
the best tradeoff between better accuracy and cheaper computing cost. In the near future, we need more experiments to validate on very large-size knowledge bases. In a knowledge graph, reasoning rules are the key parts for discovering and applying knowledge, but they are gone in knowledge graph embedding space. Embedding rules, more complex structural semantic information, is still a big challenge and also our research interesting. We think of combining formal reasoning technologies into the uncertain knowledge graph embedding space to perform complex reasoning tasks, such as non-analogical reasoning, abductive reasoning.

Acknowledgments
This research work is partly supported by Talents Introduction Project of Kunming University of Science and Technology (No.20180025).

References
[1] Bolacke, K. D., Evans, C., Paritosh, P., Sturge, T., and Taylor, J., "Freebase: a collaboratively created graph database for structuring human knowledge," In Proceedings of SIGMOD ’08, ACM, New York, NY, USA, pp. 1247–1250.
[2] Rebele T., Suchanek F., Hoffart J., Biega J., Kuzey E., Weikum G., "YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames," In The Semantic Web – ISWC 2016, Lecture Notes in Computer Science, Vol.9982. Springer, Cham.
[3] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Plataniotis, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling, "Never-ending learning," Communications of the ACM, Vol.61, May 2018, pp. 103-115.
[4] Robyn Speer, Joshua Chin, and Catherine Havasi, "ConceptNet 5.5: an open multilingual graph of general knowledge," In Proceedings of the AAAI Conference on Artificial Intelligence Vol.31, 2017, pp. 4444–4451.
[5] Chen, X., Chen, M., Shi, W., Sun, Y., and Zaniolo, C., "Embedding uncertain knowledge graphs," In Proceedings of the AAAI Conference on Artificial Intelligence, Vol.33, 2019, pp. 3363-3370.
[6] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, 2018.
[7] S. He, K. Liu, G. Ji, and J. Zhao, "Learning to represent knowledge graphs with gaussian embedding," In Proceedings of CIKM ’15, ACM, New York, NY, USA, 2015, pp. 623–632.
[8] Jiafeng Hu, Reynolds Cheng, Zhipeng Huang, Yixiang Fang, and Sijiang Luo, "On Embedding Uncertain Graphs," In Proceedings CIKM ’17, ACM New York, NY, USA, 2017, pp. 157–166.
[9] Vilnis, L., Xiang Li, Shikhar Murty and A. McCallum, "Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures," ACL (2018), 2018. (arXiv:1805.06627)
[10] Cowen-Rivers, Alexander Imani, Pasquale Minervini, Tim Rocktaschel, Matko Bosnjak, S. Riedel and J. Wang, "Neural Variational Inference For Estimating Uncertainty in Knowledge Graph Embeddings." ArXiv abs/1906.04985, 2019.
[11] Mikolov, T., Chen, K., Corrado, G., and Dean, J., "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3787, 2013.
[12] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," Neural Computation, Vol.9(8), Nov. 1997, pp. 1735–1780.
[13] Gers, Felix A., Schmidhuber, Jürgen, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM." Neural Computation, Vol.12(10), 2000, pp. 2451-2471.
[14] Mikolov, T., Sutskever I., Chen K., Corrado G.S., Dean J., "Distributed Representations of Words and Phrases and their Compositionality," Advances in Neural Information Processing Systems Vol.26, 2013, pp. 3111-3119.
[15] Kingma, D. P. and Ba, J., “Adam: A Method for Stochastic Optimization,” arXiv:1412.6980, 2014.
[16] Soro Tommaso, Stefano Ruberto, Diego Moussallem, Edgard Marx, D. Esteves and Axel-Cyrille Ngonga Ngomo, “Expeditious Generation of Knowledge Graph Embeddings,” ArXiv:abs/1803.07828, 2018.
[17] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko, "Translating embeddings for modeling multi-relational data," In Advances in Neural Information Processing Systems, 2013, pp. 2787-2795.
[18] Szkłarczyk, D., Morris, J. H., Cook, H., Kuhn, M., et al, "The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible," Nucleic acids research, 2016.
[19] Radim Rehurek, and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks 2010.