Learning Inductive Evolving Knowledge Graph Embedding and Construction of Temporally Aware Event Evolutionary Graph

Jiaqi Zeng (517021910882) [1] **Shan He (517021910894)** [1]

Abstract

This paper introduces our group's work on Evolving Knowledge Graph (**Evolving KG**). Evolving KG is a structure of knowledge graph that counts time slot of each record in, showing the evolving nature of actual relationship network. We will discuss some existing works that we referred, and raise our innovative improvements on them. Besides, we construct a temporally aware event evolutionary graph with 609,871 entities and 467,479 quadruples and at last perform experiments and data visualization on it.

1. Introduction

The knowledge graph (KG) represents a collection of interlinked descriptions of entities – real-world objects, events, situations or abstract concepts, where Descriptions have a formal structure that allows both people and computers to process them in an efficient and unambiguous manner. And entity descriptions contribute to one another, forming a network, where each entity represents part of the description of the entities, related to it.

Knowledge graph has been proven as an effective model for characterizing and studying complex multi-relational settings in real world. In recent years databases such as YAGO, Freebase, DBpedia, were established and applied into multiple research fields. Traditional KG is represented as a combination of triples:(source entity(s), relation(r), target entity(t)), that indicates the fact *subjectEntity-relation-objectEntity*, and form the entire relationship network from them.

Many studies is funded on the traditional KG structure. However, traditional knowledge graphs simply provide a snapshot of knowledge structure for specific time period. In actual world, the relationship between entities changes from time to time, and the real KG has its envolving nature.

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. **Note: The blue part of work is finished and written by Jiaqi Zeng. The black part of work is finished and written by Shan He.**.

In this paper, we will introduce our work on **Evolving Knowledge Graph**. We will first talk through existing work related, form our model structure, and then discuss some method from existing work for EvolvingKG embedding. Next we have a complete workflow for establishing our own Knowledge Graph Database "FEEG". And we made experiments and visualization upon it. The conclusion and acknowledgement through the project will be talked in the end. Our code will be available at https://github.com/HeyyyyyyG/Temporal-aware-knowledge-graph.

2. Related work

Knowledge graph is an important research area. In this section, we will introduce the related work in three aspects: (1) Knowledge Graph Representation Learning (2) Evolving Knowledge Graph Modeling (3) Temporal Fact and Event Extraction.

Knowledge Graph Representation Learning: Knowledge graphs have been verified to be useful in wide applications such as information extraction (Hoffmann et al., 2011), (Daiber et al., 2013), question answering (Lukovnikov et al., 2017), (Yih et al., 2016), named entity disambiguation (Damljanovic & Bontcheva, 2012), (Zheng et al., 2012) and semantic parsing (Berant et al., 2013). One of fundamental and important techniques in knowledge graphs is embedding, whose key idea is to embed the items in knowledge graphs, i.e., entities and relations, in continuous vector spaces, so as to make simplification while preserving the network structure. An enormous amount of research has been done in this field, especially for KG completion or link prediction task (Bordes et al., 2013). (Nickel et al., 2015) provides a detailed review of the recent KG embedding learning methods. These can be broadly categorized into two different paradigms. TransE(Bordes et al., 2013), TransH(Wang et al., 2014), TransR (Lin et al., 2015), TransD (Ji et al., 2015) are the translational distance-based models. Here the main theme is to minimize the distance between two entity vectors where one of them is translated by a relation vector. The realm of matrix factorization based methods includes bilinear model RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2014), HolE (Nickel et al., 2016). Some of the other notable models include Neural Tensor Networks(NTN) (Socher et al., 2013). However, the temporal dimension remains silent in all of these inference methods.

Evolving Knowledge Graph Modeling: A multitude of previous studies such as (Meng et al., 2016) and (Liu et al., 2016) have clarified that network structure evolves over time. Regarding this, some models have been proposed, among which preferential attachment is a simple but useful one. In preferential attachment, for various reasons, nodes with more existing edges are more likely to create a new one. It further leads to a multiplicative process which is known to give power-law distributions. Due to its usability, preferential attachment has been widely used as a basic rule in varying scenes such as social networks (Zuev et al., 2015), protein networks (Eisenberg & Levanon, 2003) and nanoparticles in liquid (Welch et al., 2016). Besides preferential attachment,(Liu et al., 2019) is among the first to extend traditional triples into quadruples, adding the time as the fourth dimension. It views time as a decaying variable which indicates the influence of temporally various tuples on current link predictions. It employ time as a factor but have not fully exploited its potential. Utilizing the form of quadruple, (Dasgupta et al., 2018) learns a hyperplane for each time period, project the triples on the hyperplane and then implement a TransE-like method. This method is straightforward but lack of the capability of predicting current links with previous quadruples.

Temporal Fact and Event Extraction: Time, apart from being an information, also introduces a separate dimension to knowledge. Therefore, temporal scoping of relational facts is an imperative part of automatic knowledge graph construction and completion. T-YAGO (Wang et al., 2010) extracts temporal facts from semi-structured data like Wikipedia Infoboxes, and categories using only regular expressions. On the other hand, systems like PRAVDA harvests temporal information from free text sources using label propagation. CoTS (Talukdar et al., 2012b) uses integer linear program based approach to model temporal constraints and proposes joint inference frame-work with few seed examples. A method for discovering temporal ordering among factual relations was proposed in (Talukdar et al., 2012a). The task of extracting temporally rich events and time expressions and ordering between them is introduced in TempEval challenge (UzZaman et al., 2013). Various approaches such as (McDowell et al., 2017) made for solving the task proved to be effective in other temporal reasoning tasks. However, many of the above methods are designed for constructing traditional knowledge graph, i.e. each node represents an entity and each edge represents a relation. Though this kind of knowledge graph is effective in many aspects, it cannot infer events directly. One promising solution is to build event evolutionary graph, in which nodes represent events and edges represent logical relationships. (Ding et al., 2019) builds a similar event logic graph but ignores the time factor. In our work, we build an temporally aware event evolutionary graph in the field of finance.

3. Method

3.1. Evolving Knowledge Graph Learning

3.1.1. Background

As introduced in section 2, EvolvingKG(Liu et al., 2019) and HyTE(Dasgupta et al., 2018) are two of the most representative works in the area of evolving knowledge graph learning. In this subsubsection, we will briefly introduce them and recognize the shortcomings of these works since our work is largely based on them. Both works improved from TransE, which is the most popular model in traditional knowledge graph modeling. The most significant point of modeling evolving knowledge graphs is how to fully exploit the potential of time. Each method in these works are creative but not perfect.

EvolvingKG is among the first to introduce time dimension into the traditional triples (h, r, t) and extent them to quadruples (h, r, t, τ). In this work, time is treated as a decaying factor, indicating that the influence of a triple decays with time. A triple has a stronger effect on current link prediction if it happens closer to present.

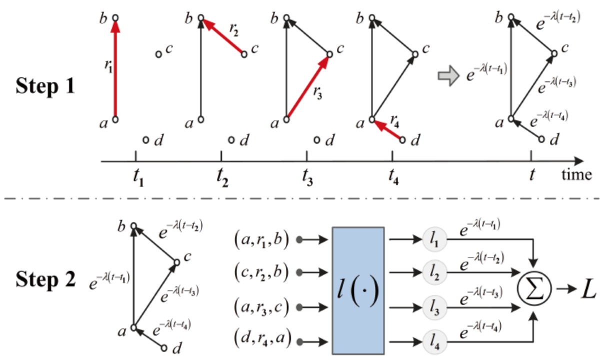

Figure 1. An illustration on the framework of (Liu et al., 2019)

For each triple, they calculate the loss with an additional factor $e^{-\lambda(\tau-\tau_i)}$. The total loss is modified as

$$L(t) = \sum_{\substack{(h,r,t,\tau_i)\in S \\ (h',r,t',\tau_i)\in S'}} [\gamma + e^{-\lambda(\tau-\tau_i)}P - e^{-\lambda(\tau-\tau_i)}N]_+,$$

where S is the set of correct quadruples, S' is the set of corrupted ones, $P = d(h + r, t)$, $N = d(h' + r, t')$, and d is the dissimilarity measure.

The advantage of this method is that it is simple to implement and there is no extra time complexity and space complexity compared to TransE. Furthermore, this can be used as both transductive learning and inductive learning, i.e. the time period of training set and testing set can either intersect or not. However, the shortcoming is that it utilizes a fix decaying function to measure the influence of time which might not fully exploit the effect of time on various triples.

HyTE also utilizes quadruples (h, r, t, τ) as an extend of (h, r, t). The most creative work of it is that it views each time period as a hyperplane. Each triple will be projected to the corresponding time hyperplane and conduct TransE on the hyperplane. In this way, we can learn the representation of a triple on different time period.

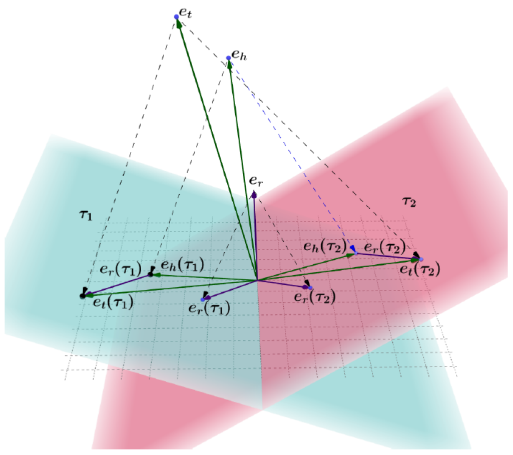

Figure 2. An illustration on the framework of (Dasgupta et al., 2018)

The strength of this method is that it learns a hyperplane for each time period which can fully exploit the feature of time compared to EvolvingKG. However, the deficiencies are obvious: (1) It can just used as transductive learning but not inductive learning. The time period of training set and testing set should be exactly the same. Otherwise, the time representation of testing set cannot be learned while training. (2) For each time period we need a specific vector to represent it which cause an $O(N)$ space complexity. This will be reduced effectively to $O(1)$ in our improvement.

3.1.2. Our Effective Improvement on HyTE for Inductive Learning

In our work, we aim to build a model which can be used for inductive learning based on HyTE. Given the timestamps, the graph can be dismantled into several static graphs consisting of triples that are valid in the respective time steps, e.g. knowledge graph G can be expressed as $G = G_{\tau_1} \cup G_{\tau_2} \cup \cdots \cup G_{\tau_T}$, where $\tau_i, i \in 1, 2, \cdots T$ are the discrete time points. Following HyTE, we represent time as a hyperplane i.e., for T number of time steps in the KG, we will have T different hyperplanes represented by normal vectors $w_{t_1}, w_{t_2}, \cdots, w_{t_T}$. Thus, we try to segregate the space into different time zones with the help of the hyperplanes. Now, triples valid at time τ (i.e., the sub graph G_τ) are projected onto time specific hyperplane w_τ , where their translational distance (TransE) is minimized.

We compute the projected representation on w_τ as following:

$$P_\tau(e_h) = (e_h - w_\tau^T e_h) w_\tau,$$

$$P_\tau(e_r) = (e_r - w_\tau^T e_r) w_\tau,$$

$$P_\tau(e_t) = (e_t - w_\tau^T e_t) w_\tau,$$

where $||w||_2 = 1$.

Intuitively, if we want to conduct inductive learning, i.e. predict the corresponding w_τ given an unseen τ, w_τ should follows a consistent function $w_\tau = g(\tau)$.

Strategy 1:

In (Dasgupta et al., 2018), though training without any limits on w_τ, it has been proved that adjacent time period have similar hyperplane w_τ. A straightforward assumption is that w_τ changes with a fixed Δw along with time. This can be represented as

$$\Delta w_{i+1} = w_{\tau_{i+1}} - w_{\tau_i},$$

$$\Delta w_i = w_{\tau_i} - w_{\tau_{i-1}},$$

$$\Delta w_{i+1} = \Delta w_i = \Delta w$$

where $i \in 2, 3, \cdots, T-1$. In order to verify this assumption, we employ PCA to visualize all Δw_i. The result is shown in Figure 3. It is easy to find out most points

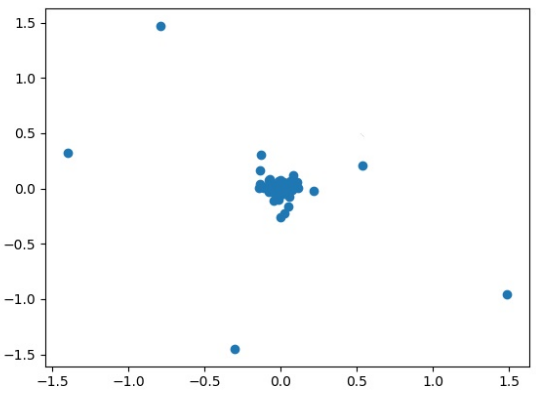

Figure 3. PCA on Δw_i.

aggregate in the middle, indicating the homology between Δw_i. According to this assumption, we get

$$w_\tau = g_1(\tau) = w_{\tau_1} + (\tau - \tau_1)w'.$$

Using this strategy, we only need to learn w_{τ_1} and w' which leads to an $O(1)$ space complexity.

Strategy 2:

In a more complex and general perspective, instead of assuming Δw to be static, we can also assume that $Deltaw$ varies along the time. Describing this change, a possible function is

$$\Delta w = a\Delta w_1 + b.$$

Through this we can derive a quadratic function f_2 for w_τ as

$$w_\tau = g_2(\tau) = w_{\tau_1} + (\tau - \tau_1)w' + (\tau - \tau_1)^2 w''.$$

Similarly, we only need to learn w_{τ_1}, w', and w'' which also leads to an $O(1)$ space complexity.

Strategy 3:

From Figure 3 we see that although most of the points gather, there is still some outliers. This indicates the above strategies might not work in all scenarios. In order to solve this problem, we further employ the idea of sampling. We assume that w' (and w'') are not fixed values, instead they follow Gaussian distribution. When calculating w_τ, they are sampled from the following distributions:

$$w' \sim N(\mu_{w'}, \sigma_{w'}),$$

$$w'' \sim N(\mu_{w''}, \sigma_{w''}).$$

The sampling process not only solves the problem of outliers but also increases the robustness of our method. However, the process cannot support gradient descent as the optimization strategy since it is non-differentiable. Therefore, we employ the reparameterization operation. Instead of sampling directly from the distribution of w' and w'', we sample a random variable δ from the standard normal distribution

$$\delta \sim N(\overrightarrow{0}, I).$$

Maintaining the vectors of $\mu_{w'}$, $\sigma_{w'}$, $\mu_{w''}$, and $\sigma_{w''}$, we calculate w' and w'' as

$$w' \leftarrow \mu_{w'} + \sigma_{w'} \times \delta,$$

$$w'' \leftarrow \mu_{w''} + \sigma_{w''} \times \delta.$$

In this way, the addition and multiplication operation supports back propagation. The space complexity is doubled but still maintained $O(1)$.

3.1.3. Optimization

We expect for all positive quadruples, $P_\tau(e_h) + P_\tau(e_r) \approx P_\tau(e_t)$. Following (Dasgupta et al., 2018), we use the following scoring function.

$$f_\tau(h, r, t) = ||P_\tau(e_h) + P_\tau(e_r) - P_\tau(e_t)||_{l_1/l_2}.$$

We minimize the margin-based ranking loss

$$L(t) = \sum_{\substack{(h,r,t,\tau_i)\in S \\ (h',r,t',\tau_i)\in S'}} max(0, \gamma + f_\tau(h,r,t) - f_\tau(h',r,t')),$$

where S is the set of correct quadruples, S' is the set of corrupted ones.

3.2. Constructing Temporally Aware Event Evolutionary Graph

From our perspective, though traditional knowledge graphs are popular these days, they still have some deficiencies, such as the lack of ability to infer. Instead, event evolutionary graph can handle this problem by setting each node as an event and each edge as an logic relation. In our work, we focus on the cause-and-effect relationship. Furthermore, we add the time of the event pairs as an attribute of the edge. The construction of temporally aware event evolutionary graph is more of an engineering work than a scientific research. We crawl news from the Internet and extract event pairs from it. The whole process is shown as Figure 4. For example, given a piece of news saying "According to the xx times report on March 15, 2020, a new coronary pneumonia with uncertain sources has become a pandemic worldwide, leading to a world economic recession", we can extract a event pairs with its timestamp. The most significant part in the construction is causality extraction. In our work, we try two different methods for it.

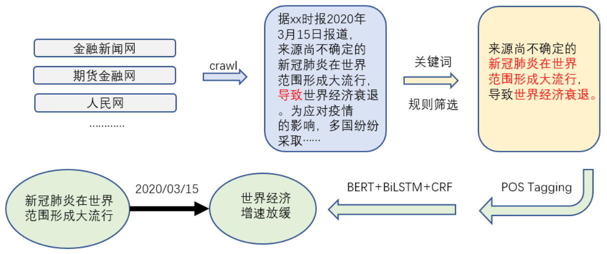
金融新闻网
期货金融网
人民网
....
crawl
据xx时报2020年3月15日报道，来源尚不确定的新冠肺炎在世界范围形成大流行，导致世界经济衰退。为应对疫情的影响，多国纷纷采取……
关键词
规则筛选
来源尚不确定的新冠肺炎在世界范围形成大流行，导致世界经济衰退。
POS Tagging
BERT+BiLSTM+CRF
新冠肺炎在世界范围形成大流行
2020/03/15
世界经济增速放缓

Figure 4. Process of constructing temporally aware event evolutionary graph.

3.2.1. Unsupervised Causality Extraction

The first step to construct event evolutionary graph is to identify cause-effect pairs from unstructured natural language texts. As the amount of data is extremely large (millions of documents), obtaining human-annotated pairs is impossible. We find that causal relations expressed in text have various forms. We therefore provide a procedure similar to this work (Ding et al., 2019), which can automatically identify mentions of causal events from natural language texts.
We construct a set of rules to extract mentions of causal events. Each rule follows the template of ¡Pattern, Constraint, Priority¿, where Pattern is a regular expression containing a selected connector, Constraint is a syntactic constraint on sentences to which the pattern can be applied, and Priority is the priority of the rule if several rules are matched. For example, we use the pattern "[cause] leads to [effect]" to extract the causal relation between two events.

3.2.2. Supervised Causality Extraction

Though unsupervised method requires little human efforts, it usually extracts some extra useless words. For example, for the sentence shown in Figure 4, "According to the xx times report on" is redundant for constructing event evolutionary graph. To extract precisely, we conduct supervised causality extraction. As illustrated in Figure 5, we use Bert and BiLSTM+CRF model to extract causal relations. Language model pre-training has shown to be very effective for learning universal language representations by leveraging large amounts of unlabeled data. Some of the most prominent models are ELMo (Peters et al., 2018), GPT (Radford et al., 2018), and BERT (Devlin et al., 2018). BERT uses the bidirectional transformer architecture. There are two existing strategies for applying pre-trained language models to downstream tasks: feature-based and fine-tuning.
In this paper, we annotate each token in a sentence with following tags: B-cause, I-cause, B-effect, I-effect and O. The tag "B-cause" refers to the beginning token of the cause event and each rest token in the cause event is represented by "I-cause". The tag "O" refers to the normal token which is irrelevant with causality. We feed the hidden representation for each token i after BERT as the input layer of BiLSTM. These hidden representations can be viewed as semantic features learnt from Bert model. The output representation layer of BiLSTM is then fed into the classification layer to predict the causal tags. The predictions in the classification layer are conditioned on the surrounding predictions by using the CRF method.

However, though the classical BERT+BILSTM+CRF model is strong for many natural language processing task, it was originally designed for NER (Named Entity Recognition) task. It does not consider the special attributes of event. Therefore, we make a small improvement in order to extract events.

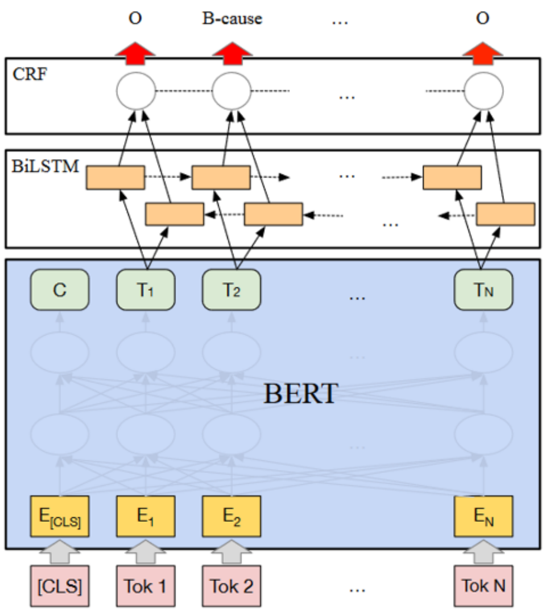

Figure 5. Classical BERT+BILSTM+CRF model for causality extraction.

In our observation, events usually contains some frequently patterns regarding to the part of speech (POS). For example, an event is composed of "nouns + verbs" or " nouns + adjectives". According to this, besides the embeddings generated by BERT, we further learns embeddings for each part of speech. We utilize jieba to conduct POS tagging, which outputs 25 categories of POS. For each category, we learn a d-dimensional vector c_i. The corresponding c_i for each token forms a matrix $C \in N^{n\times d}$. We denote embeddings obtained by BERT as $B \in N^{n\times D}$. We concatenate them and get joint embedding matrix $J = [B, C] \in N^{n\times(D+d)}$. J is then fed into the BILSTM+CRF model. This process is shown in Figure 6. The learning of c_i is conducted by a two-layer MLP. The inputs are one-hot vectors. It can be represented as

$$c_i = W_{c2}(W_{c1}c_{onehot_i} + b_1) + b_2.$$

3.2.3. Active Training for Supervised Causality Extraction

Since labeling a large number of cause-effect pairs clearly does not scale, we adopt active learning as a more guided approach to select examples to label so that we can economically learn an accurate model by reducing the annotation cost. It is based on the premise that a model can get better performance if it is allowed to prepare its own training data,

Table 1. Entity Prediction on YAGO11K for transductive learning

Stats.	Mean Rank		Hit @10	
	tail	head	tail	head
TransE (Bordes et al., 2013)	504	2020	4.4	1.2
TransH (Wang et al., 2014)	354	1808	5.8	1.5
HolE (Nickel et al., 2016)	1828	1953	29.4	13.7
t-TransE (Jiang et al., 2016)	292	1692	6.2	1.3
HyTE (Dasgupta et al., 2018)	**110**	1069	34.9	13.3
Ours-strategy3-1	219	**962**	34.3	**17.5**
Ours-strategy3-2	137	976	**35.8**	**17.5**

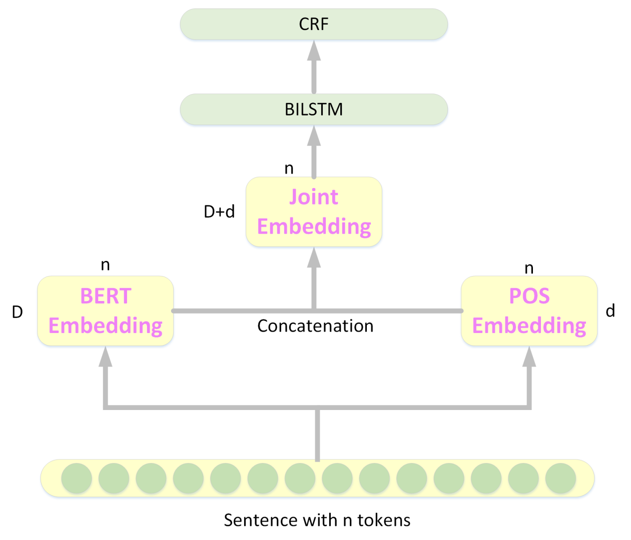

Figure 6. Improved BERT+BILSTM+CRF model for causality extraction.

Algorithm 1 UCS Active Training Algorithm for efficient supervised causality extraction

```
Input: unlabeled dataset D, test dataset T , model f , human labeling H, the number of human labeling samples in each iteration K;
Output: model $\hat{f}$, predict score S
procedure
$i \leftarrow 0$
$D_0 \leftarrow random\_select(D, K)$
$L_0 \leftarrow H(D_0)$
$D \leftarrow D - D_0$
$\hat{f}, fs \leftarrow train\_test(f, L_0, T)$
repeat
    $D_{i+1} \leftarrow D(Top(S_i, \alpha K)) \cup D(Bottom(S_i, (1 - \alpha)K))$
    $L_{i+1} \leftarrow H(D_{i+1}) \cup L_i$
    $D \leftarrow D - D_{i+1}$
    $\hat{f}, fs \leftarrow train\_test(f, L_{i+1}, T)$
    $S \leftarrow \hat{f}(D)$
until fs not improves in n step
end procedure
```

by choosing the most beneficial data points and querying their annotations from annotators. We propose an uncertainty and high confidence sampling strategy (UCS) to select samples which can improve model effectively. The iterative active learning algorithm is shown in Algorithm 1.

4. Experiment

4.1. Experiment for Evolving Knowledge Graph Learning

We evaluate our model and compare with different state-of-the-art baselines based on Link prediction. Evaluation metrics used are same as that of the traditional KG embedding method for link prediction task.

4.1.1. Datasets

In our experiment, we evaluate our method on two datasets: a YAGO dataset (traditional knowledge graph) as well as a financial event evolutionary graph (FEEG) dataset built by ourselves.

YAGO11K: In the YAGO3 knowledge graph (Mahdisoltani et al., 2013), some temporally associated facts have meta-facts as (#factID, occurSince, t_s), (#factID, occurUntil, t_e). The total number of time annotated facts containing both occursSince and occursUntil are 722,494. Out of them, we selected top 10 most frequent temporally rich relations. In order to handle sparsity, we recursively remove edges containing entity with only a single mention in the subgraph. This ensures a healthy connectivity within the graph. Finally, we obtain a purely temporal graph of 20.5k triples and 10,623 entities by following this procedure. Additionally, for inductive learning, we select 10k triples from 1800 to 1986 as training set, 5k triples from 1987 to 2004 as testing set. We randomly split 10% of the training set as validation set.

FEEG: This dataset is built by ourselves using the pipeline

Table 2. Entity Prediction for inductive learning

Stats.	Mean Rank		Hit @10	
	tail	head	tail	head
YAGO11K				
TransE (Bordes et al., 2013)	1737	2845	15.70	6.78
HyTE (Dasgupta et al., 2018)	1573	2608	16.78	8.21
EvolveKG (Liu et al., 2019)	1687	2623	**17.14**	6.78
Ours-strategy3-1	1576	**2589**	16.80	9.40
Ours-strategy3-2	**1559**	2599	16.88	**9.56**
FEEG				
TransE (Bordes et al., 2013)	31708	33329	0.21	0.06
HyTE (Dasgupta et al., 2018)	31644	33103	0.17	0.00
EvolveKG (Liu et al., 2019)	32686	2623	0.25	0.20
Ours-strategy3-1	29335	**29686**	0.31	**0.26**
Ours-strategy3-2	**28978**	29973	**0.34**	**0.26**

described in subsection 3.2. Each quadruple maintains the form of $(cause, relation, effect, timestamp)$. Here the relation only has one type: cause-effect relation. During our construction, we extract totally 609,871 entities and 467,479 quadruples. Limited by the computing power, we only use a subset of it in our experiment. The training set contains 30k quadruples ranging from 2012 to 2017 while the testing set contains 5k quadruples from 2017 to 2018. We randomly split 20% of the training set as validation set.

4.1.2. Implementation Details

For all the methods, we have kept batch size b = 50k on both the datasets. Because of the limited time and computing power, we have not carefully tuned the hyper-parameters. The dimensions of the embeddings are set as 128. The margins for all the methods are set as 10. We use SGD for optimization and the learning rate is set as 0.0001.

4.1.3. Performance

Transductive learning: Following (Dasgupta et al., 2018), we first conduct transductive learning on the YAGO11K dataset to prove that our method is also better than previous methods when the time period of training set intersects the time period of testing test. We evaluate the performance on entity prediction as well as relation prediction as shown in Table 1 and Table 3. "Our-strategy3-1" and "Our-strategy3-2" denote the methods of implementing strategy 3 on strategy 1 and strategy 2 respectively. From the results we can see that our methods outperform the baseline models in most metrics which indicates the effectiveness of our methods. We can find that our methods gain great improvement on the performance of head prediction, which might be the effect of strategy 3.

Inductive learning: We conduct inductive learning on two datasets to validate our design. The performance of entity

Table 3. Relation Prediction on YAGO11K for transductive learning

Stats.	Mean Rank	Hit @1
TransE (Bordes et al., 2013)	1.7	78.4
TransH (Wang et al., 2014)	1.53	76.1
HolE (Nickel et al., 2016)	2.57	69.3
t-TransE (Jiang et al., 2016)	1.66	75.5
HyTE (Dasgupta et al., 2018)	**1.23**	81.2
Ours-strategy3-1	**1.23**	**84.1**
Ours-strategy3-2	1.33	81.8

Table 4. Relation Prediction on YAGO11K for inductive learning

Stats.	Mean Rank	Hit @1
TransE (Bordes et al., 2013)	3.24	**47.1**
HyTE (Dasgupta et al., 2018)	3.48	35.1
EvolveKG (Liu et al., 2019)	3.08	38.4
Ours-strategy3-1	**2.94**	38.2
Ours-strategy3-2	2.98	38.5

prediction is shown in Table 2. Since there is only one type of relation in FEEG dataset, we only conduct relation prediction on YAGO11K dataset, whose results are shown in Table 4. The performance shows that our design outperforms some of the start-of-the-art methods.

4.2. Experiment for causality extraction

Before constructing the temporally aware event evolutionary graph, we need to validate the effectiveness of the classical BERT+BILSTM+CRF model as well as the improved BERT+BILSTM+CRF model. We manually annotate 120 sentences as testing set and evaluate the model after active training. The results are shown in Table 5. From Table 5 we see that our design of adding POS embeddings are useful for causality event pairs extraction.

Table 5. Performance of BERT+BILSTM+CRF model on causality extraction.

STATS.	PRECISION	RECALL	F1-SCORE
CLASSICAL	61.05	80.11	69.29
IMPROVED	65.31	84.53	73.68

5. Visualization

5.1. Rendering methods

For the final work to present our result in a proper way, we determined some method on displaying our database **FEEG**. As the traditional KG is represented as network formed from triples, we firstly establish the base net with triple information: Source Node = source entity, Target Node = objection entity, Edge = relationship.
As for nodes, different nodes have different text contents, and we leave the detailed contents as node label. For edges, we make the detailed relationship name as edge label (or sometimes we can change the edge shape for conspicuousness). As there is only one relationship (causality) in our database, we just to draw a single type of edge.
At last, to show the time evolve of KG, we use different color onto edges and source nodes to label the time element.

5.2. Visualization Results

A common economic issue may be generated from several former changes. In end of 2017, gold price in US had a rush boosting, and many texts in our database have mentioned the problem. So we take information net centering with it(from time 2016-2017), and the result is shown in figure 7. The net snapshot seems reasonable, as it obeys the actual life rules: One issue is caused by many others, and from one as the root , we can get a huge relation tree.
To take a larger view of FEEG, we extract about 800 entities with about 560 relations from the database and use *Gephi* to draw a relationship net. Here we use some way to take quarts to make the edges be as many as possible, just to make the result better. The result is shown in figure 8. Figure 9 shows the detail of it.
However, after drawing the overview of common data, it's easy to observe that FEEG data tends to make small clusters usually with 2 or 3 nodes. This discover indicates that the whole 5k quadruples in FEEG also tend to group into small clusters. As we said in this paper before, the sparsity of KG can lower the efficiency and accuracy of embedding model method.
After using FEEG, we also draw maps with YAGO11K(takes one relation only), and the result shows the same sparsity. It can be indicated that KG data generated in real world for now tends to get sparse due to source limit. If we develop more subtle method to get more information and generate a thick relation map, the performance of existing experiment about Evolving KG will be better.

6. Conclusion

This paper introduces our group's work on Evolving Knowledge Graph (**Evolving KG**), a structure of knowledge graph that counts time slot of each record in, showing the evolving nature of actual relationship network.
We first discuss some existing work that we referred, defining that in our model each record will be saved into quarts. Then we compare two methods for Knowledge Graph embedding: **Evolving KG** and **HyTE**. We raise three strategies for improving the method performance of HyTE.
Then we go through the whole process to construct a temporally aware event evolutionary graph, mentioning different methods of each step and we name it **FEEG**. Then we take experiments on YAGO11K and FEEG datasets to exam our improvement strategies. The result indicates the feasibility of our thoughts.
And at last we make visualization of our database to observe the character of it. The result shows that KG database generated for now is usually node-sparse, causing lower performance in different embedding methods.

7. Acknowledgements

We want to thank our teacher Ms.Fu, to whom we first consulted. Ms Fu provided us useful reading materials to step into this research field. A part of our work certainly refers to Fu's paper ((Liu et al., 2019)). We are also grateful to the TAs, who listed myriad projects for us and arranged colorful presentations.

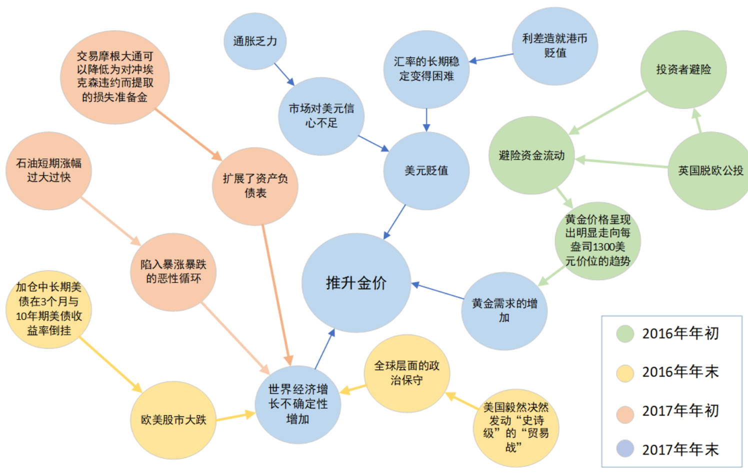

Figure 7. The reasoning knowledge graph drawn centering with ”Gold price boosts” in late 2017.

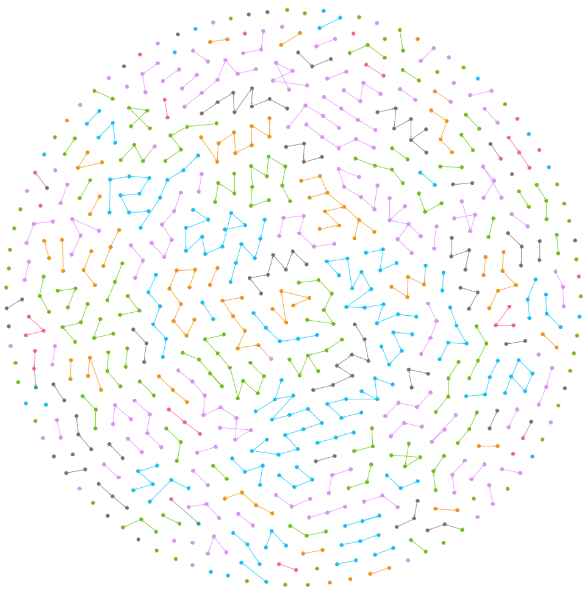

Figure 8. The knowledge network with about 800 entities drawn from FEEG.

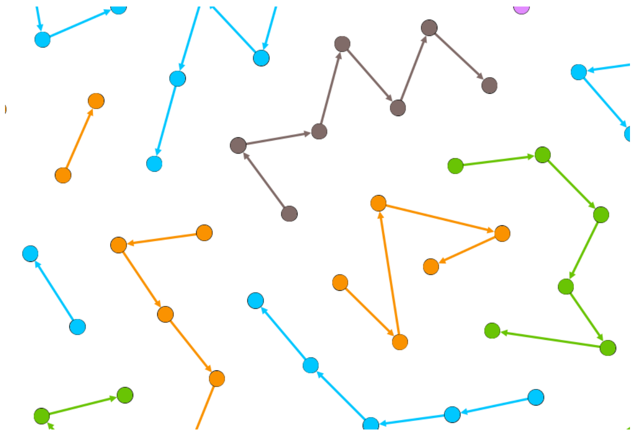

Figure 9. Detailed view for figure8

References

Berant, J., Chou, A., Frostig, R., and Liang, P. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1533–1544, 2013.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pp. 2787–2795, 2013.

Daiber, J., Jakob, M., Hokamp, C., and Mendes, P. N. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pp. 121–124, 2013.

Damljanovic, D. and Bontcheva, K. Named entity disambiguation using linked data. In *Proceedings of the 9th Extended Semantic Web Conference*, pp. 231–240, 2012.

Dasgupta, S. S., Ray, S. N., and Talukdar, P. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2001–2011, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ding, X., Li, Z., Liu, T., and Liao, K. Elg: An event logic graph. *arXiv preprint arXiv:1907.08015*, 2019.

Eisenberg, E. and Levanon, E. Y. Preferential attachment in the protein network evolution. *Physical review letters*, 91 (13):138701, 2003.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., and Weld, D. S. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 541–550. Association for Computational Linguistics, 2011.

Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 687–696, 2015.

Jiang, T., Liu, T., Ge, T., Sha, L., Li, S., Chang, B., and Sui, Z. Encoding temporal information for time-aware link prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2350–2354, 2016.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

Liu, J., Zhang, Q., Fu, L., Wang, X., and Lu, S. Evolving knowledge graphs. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2260–2268. IEEE, 2019.

Liu, Q., Zhao, X., Willinger, W., Wang, X., Zhao, B. Y., and Zheng, H. Self-similarity in social network dynamics. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 2(1):1–26, 2016.

Lukovnikov, D., Fischer, A., Lehmann, J., and Auer, S. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, pp. 1211–1220, 2017.

Mahdisoltani, F., Biega, J., and Suchanek, F. M. Yago3: A knowledge base from multilingual wikipedias. 2013.

McDowell, W., Chambers, N., Ororbia II, A. G., and Reitter, D. Event ordering with a generalized model for sieve prediction ranking. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, 2017.

Meng, L., Hulovatyy, Y., Striegel, A., and Milenković, T. On the interplay between individuals' evolving interaction patterns and traits in dynamic multiplex social networks. *IEEE Transactions on Network Science and Engineering*, 3(1):32–43, 2016.

Nickel, M., Tresp, V., and Kriegel, H.-P. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pp. 809–816, 2011.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

Nickel, M., Rosasco, L., and Poggio, T. Holographic embeddings of knowledge graphs. In *Thirtieth Aaai conference on artificial intelligence*, 2016.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Socher, R., Chen, D., Manning, C. D., and Ng, A. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pp. 926–934, 2013.

Talukdar, P. P., Wijaya, D., and Mitchell, T. Acquiring temporal constraints between relations. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 992–1001, 2012a.

Talukdar, P. P., Wijaya, D., and Mitchell, T. Coupled temporal scoping of relational facts. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 73–82, 2012b.

UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., and Pustejovsky, J. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 1–9, 2013.

Wang, Y., Zhu, M., Qu, L., Spaniol, M., and Weikum, G. Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In *Proceedings of the 13th International Conference on Extending Database Technology*, pp. 697–700, 2010.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.

Welch, D. A., Woehl, T. J., Park, C., Faller, R., Evans, J. E., and Browning, N. D. Understanding the role of solvation forces on the preferential attachment of nanoparticles in liquid. *ACS nano*, 10(1):181–187, 2016.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Yih, W.-t., Richardson, M., Meek, C., Chang, M.-W., and Suh, J. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 201–206, 2016.

Zheng, Z., Si, X., Li, F., Chang, E. Y., and Zhu, X. Entity disambiguation with freebase. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pp. 82–89. IEEE, 2012.

Zuev, K., Boguná, M., Bianconi, G., and Krioukov, D. Emergence of soft communities from geometric preferential attachment. *Scientific reports*, 5:9421, 2015.