Joint Knowledge Acquisition via Extraction and Inference

Swapnil Gupta Rahul Chittimalla Dr. Partha Pratim Talukdar

Abstract

Recent research has resulted in the development of several large knowledge Graphs (KGs) to organize world knowledge. KGs have been applied to many tasks including web search, link prediction, recommendation, natural language processing, and entity linking. But these KGs require significant human supervision and are generally sparse in nature. This attracted significant research endeavors from the research community towards automatic knowledge graph construction (AKGC). In this work we have explored the joint training paradigm for two important tasks for AKGC relation extraction and Knowledge Graph Completion recently proposed in (Han et al., 2018). In this work we also note that with Distant Supervision it is possible to automatically generate large amount of annotated data for Relation Extraction and hence there is key requirement to be able train the models in distributed settings. To address this we have also discussed some limitations of the dataset proposed in the original paper and have proposed a new Distant Supervision based dataset which is considerably larger then the original dataset for the task.

1. Introduction

Knowledge Graphs (KGs) are a special type of information network that represents knowledge using triples $< h, r, t >$, where h represents some head entity and *r* represents some relationship that connects h to some tail entity *t*. In this formalism a statement like Narendra Modi is Prime Minister of India can be represented as <Narendra Modi, PrimeMinisterOf, India>. Recently, a variety of KGs, such as DBPedia (Lehmann, 2015), YAGO, Freebase have been curated in the service of fact checking (Shi & Weninger, 2016), question answering (Lukovnikov & Auer, 2017), entity linking (Hachey & Curran, 2013). Despite their usefulness and popularity, the significant human supervision required in their construction limits their utility. Moreover, these KGs tend to be very sparse. For example, DBPedia, which is generated from Wikipedias infoboxes, contains 4.6 million entities, but half of these entities contain less than 5 relationships.

Motivated by these observations, the research community has been actively involved in various tasks with a broader aim to automatically generate new novel facts. There are two parallel paradigms for fact generation. The first one aims at extracting facts from the text in the tasks Relation Extraction (RE). While in the other paradigm the task is to infer new facts from the existing graph called as Knowledge Graph Completion (KGC). Below, I will briefly describe the two tasks.

1.1. Knowledge graph completion

KGC aims to enrich KGs with novel facts based on the inherent structure of KGs, including graph-based models (Lao & Cohen, 2010), tensor-based models (Socher & Ng, 2012) and translation models (Bordes & Yakhnenko, 2013). The task is defined as:

Definition: Given an incomplete Knowledge Graph *G = (E,R,T)*, where *E*, *R*, and *T* are the entity set, relationship set, and triple set respectively, KGC completes *G* by finding a set of missing triples

$$T^{'} = \{\text{<h, r, t>} \mid \text{h} \in \mathbf{E}, \text{r} \in \mathbf{R}, \text{t} \in \mathbf{E}, \text{<h, r, t>} \notin \mathbf{T}\}.$$

KGC models heavily rely on the connectivity of the existing KG and are best able to predict relationships between existing, well-connected entities. A key challenge here is how to make reliable predictions for poorly connected entities. Research community have proposed with significant success to make use of large unstructured text corpus to attend to this problem of sparsity of context present in the KGs to make more reliable link predictions. Our work also falls in this paradigm of knowledge graph completion making use of text.

1.2. Relation Extraction

Relation Extraction aims at extracting semantic relationships between entity pairs present in a text corpus. Formally the task is defined as below.

Definition: Given an entity pair *(e1,e2)* and an entity annotated sentence the task is to predict the relation between the two entities using the context present in the sentence.

Effective training of relation extraction models require significant amount of annotated data which are quite expensive to generate. To alleviate this (Mintz & Jurafsky, 2009) proposed to a Distant Supervision (DS) based model to automtically construct huge datasets making use of the available KGs. The proposed hypothesis is if two entities have a relationship in a KG, then all sentences mentioning those entities express the same relation. This hypothesis forms the basis of out joint training paradigm.

A key observation is that with the availability of huge KBs of facts and under the distant supervision paradigm increasingly large amount of training data can be generated. And to properly leverage this significant resource it is very important to be able to train these models under distributed computing settings. The aim of this work is to analyze the joint training paradigm of these complimentary tasks for Knowledge Acquisition as well as to analyze some important questions which become relevant while training models in distributed setting. To this effect in this work we are proposing a new dataset for this task which is 6 times the size dataset proposed in (Han et al., 2018). For distributed computing we are using the distributed tensorflow framework.

2. Related Work

The work relates to representation learning of KGs and joint learning with textual relations, relation extraction and neural networks with attention. We review related works as follows.

Representation Learning of KGs A variety of approaches have been proposed to encode entities and relations into a continuous low-dimensional space. TransE (Bordes & Yakhnenko, 2013) regards the relation r in the given fact *(h, r, t)* as a translation from h to t within the low-dimensional space. TransE achieves good results and has many extensions, including TransR (Lin & Zhu, 2015), etc. Tensor-based models, such as RESCAL (), HOLE (Nickel, 2016), are also effective but trained slowly. In this work, TransE is incorporated as representative in the framework to handle representation learning of KGs.

Joint Learning for Knowledge Acquisition. Some works attempt to combine KGs and text for KA. Weston et al. (2013) directly sum up knowledge and text ranking scores. Xie et al. (2016) and Wang and Li (2016) use neural networks to embed text descriptions into KG embedding spaces. Toutanova et al. (2015) extract textual relations using dependency parsing to incorporate text information. These models need well-aligned datasets and cannot be well generalized to most general cases of combining KGs and text. Wang et al. (2014a) train words and entities together to let them share parameters. Riedel et al. (2013) propose universal schema to transmit information between relations of KGs and textual patterns via their common entity pairs. Verga et al. (2016) further incorporate neural networks to relax constraints imposed by entity pairs in universal schema. These models have no need of strictly aligned datasets but only take partial information into consideration. In this paper, we build a general joint learning framework, which aligns words, entities and relations at the same time.

Relation Extraction. Many methods aim to extract relational facts from large-scale text corpora. (Mintz et al., 2009) propose distant supervised model. Then (Xu et al., 2013) propose a multi-instance mechanism. In recent years, convolutional neural networks (CNN) (Zeng et al. 2014; 2015; 2017), recurrent neural networks (RNN) (Zhang and Wang 2015) and long short-term memory networks (LSTM) (Miwa and Bansal 2016) have been proposed to identify relations between entities in given sentences. These neural models are capable of accurately capturing textual relations without explicit linguistic analysis. In this work, CNN and PCNN models are used to embed textual relations due to their time efficiency.

Neural Networks with Attention. In KA, Lin et al. (2016) and Luo et al. (2017) build a sentence-level attention over multiple instances to reduce weights of noisy instances. Verga and McCallum (2016) use neural networks with attention to merge similar semantic patterns in universal schema. We propose a mutual attention in this paper. Our attention combines models and serves as a channel for information sharing. Moreover, the attention lets models of KGs and text use additional information for mutual model improvements.

3. Method

(Han et al., 2018) proposes a general joint representation learning framework for knowledge acquisition (KA) on two tasks, knowledge graph completion (KGC) and relation extraction (RE) from text. The representations of knowledge graphs (KGs) and text within a unified parameter sharing semantic space. Let θ denote the model parameters, the framework aims to find optimal parameters

$$\theta = \operatorname*{argmax}_{\theta} P(G, D|\theta)$$

$P(G, D|\theta)$ is the conditional probability defined over the knowledge graph G and the text corpus D given the parameters θ. Let $\theta = \{\theta_E, \theta_R, \theta_V\}$ where $\{\theta_E, \theta_R, \theta_V\}$ are parameters for entities, relations and words respectively. To learn from relational triples of KGs, the conditional probabilities $P(h|(r,t), \theta_E, \theta_R)$, $P(t|(h,r), \theta_E, \theta_R)$ are optimized.

For each entity pair (h, t) in G, its latent embedding $\mathbf{r}_{h,t}$ is defined as a translation from **h** to **t**, which can be formalized as

$$\mathbf{r}_{h,t} = \mathbf{t} - \mathbf{h}$$

The scoring function is defined as:

$$f_r(h,t) = b - \|\mathbf{r}_{h,t} - \mathbf{r}\|$$

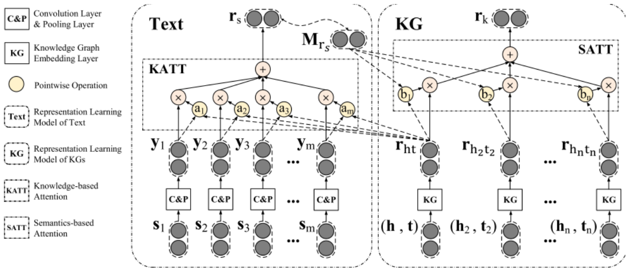
C&P
Convolution Layer & Pooling Layer
KG
Knowledge Graph Embedding Layer
Pointwise Operation
Text
Representation Learning Model of Text
KG
Representation Learning Model of KGs
KATT
Knowledge-based Attention
SATT
Semantics-based Attention
Text
r_s
M_{r_s}
KATT
a_1
a_2
a_3
a_m
y_1
y_2
y_3
y_m
C&P
s_1
s_2
s_3
s_m
KG
r_k
SATT
b_1
b_2
b_n
r_ht
r_{h_2t_2}
r_{h_nt_n}
(h , t)
(h_2 , t_2)
(h_n , t_n)

Figure 1. The framework for joint representation learning of KGs and text with the mutual attention

where b is a bias constant. To learn the representations from textual relations CNN is applied. For each word in a given sentence s containing (h, t) with a textual relation r_s, we concatenate its word embedding $\mathbf{w}_i$ and position embedding $\mathbf{p}_i$ to build its input embedding $\mathbf{x}_i$. A window of size m over the input sequence **s** in the convolution layer. For each move, we can get a hidden layer vector as

$$\hat{\mathbf{x}_i} = [\mathbf{x}_{i-\frac{m-1}{2}}; ...; \mathbf{x}_i; ...; \mathbf{x}_{i+\frac{m-1}{2}}],$$

$$\mathbf{h}_i = tanh(\mathbf{W}\hat{\mathbf{x}}_i + b),$$

where **W** is the convolution kernel, **b** is a bias vector. In the pooling layer, a max-pooling operation over the hidden layer vectors $\mathbf{h}_1$,...,$\mathbf{h}_n$ is applied to get the final output embedding **y**.

3.1. Mutual Attention between KGs and Text

We use latent relational embedding $\mathbf{r}_{h,t}$ as the knowledge-based attention over sentences to highlight important sentences and reduce noisy components as the sentences labeled by the distant supervision algorithm contain some vague and wrong semantic components. There are several entity pairs $\{(h_1, t_1), ..., (h_n, t_n)\}$ for each relation $r \in R$. Let the latent relation embeddings of these pairs be $\{\mathbf{r}_{h_1t_1}, ..., \mathbf{r}_{h_nt_n}\}$. In order to make knowledge graph representation models more effective, semantic information is extracted from text models to help explicit relations fit most reasonable entity pairs as follows,

$$\mathbf{e}_r = tanh(\mathbf{W}_s\mathbf{M}_r + \mathbf{b}_s),$$

$$b_j = \frac{exp(\mathbf{e}_r \cdot \mathbf{r}_{h_jt_j})}{\sum_{k=1}^{n} exp(\mathbf{e}_r \cdot \mathbf{r}_{h_kt_k})}$$

$$\mathbf{r}_k = \sum_{j=1}^{n} b_j \mathbf{r}_{h_jt_j},$$

where $\mathbf{W}_s$ and $\mathbf{b}_s$ are the same weight matrix and bias vector used in knowledge-based attention.

4. Dataset

For our analysis we have performed experiments on two different datasets whose statistics are discussed in table 1. **FB60K** is extended from the dataset developed by (Riedel, Yao, and McCallum 2010), which has been used as the benchmark for RE. **FBCLUE** is a dataset proposed by us which is generated through by extracting sentences from Clueweb11 text corpus under the Distant Supervision assumption using the publicly available FB15K KG.

The key motivation for the new dataset is in support of our initial argument for the requirement of training models in distributed settings to leverage large amount of annotated data that can be generated under Distant Supervision.

Datasets	FB60K	FBCLUE
#Bags	293,120	308,400
#Sentences	570,088	3,134,567
#RE entities	63,696	13846
#RE relations	53	1345
#KG entities	37561	14951
#KG relations	1318	1345

Table 1. Dataset Statistics

The key comparison points here are first the proposed dataset is 6 times as large as the original dataset. The density of sentences in each bag is much more balanced. Also, it is quite strange that in the original dataset there are several entities in the RE dataset which are not present in the KG which is quite counter intuitive for distant supervision itself. In the proposed dataset we have made sure to avoid such instances. And finally in the original dataset the only a very small subset of relations are being used for RE task while in the proposed dataset set of Relations remain the same for both RE and KGC tasks which we believe is more relevant from the joint KA perspective.

5. Experimentation Results

Through these experimentations we aim to study the impact of training model in distributed settings. We have used distributed tensorflow framework for model implementations. Below are some key distributed tensorflow API's that are used for the experimentations

- **tf.train.SyncReplicasOptimizer:** This optimizer avoids stale gradients by collecting gradients from all replicas, averaging them, then applying them to the variables in one shot.
- **tf.contrib.opt.DropStaleGradientOptimizer:** This optimizer records the global step for each worker before computing gradients and compares it with the

global step at the time of applying the gradients. If the difference is larger than a threshold, it will drop all the computed gradients.

This is to note that while the distrubuted tensorflow framework allows to train models in Bulk Synchronous and Asynchronous modes but there are no available API's for training models in true Stale Synchronous mode. Through *tf.contrib.opt.DropStaleGradientOptimizer* API the workers still run asynchronously but we can drop stale gradients.

All the experiments under distributed settings are run in data parallel mode where each worker iterates over a different shard of the data. In this section we have just reported in our observations. Based on the observations we have presented our analysis in section 6.

We have identified the following interesting questions which we aim to answer through our experimentations.

- Effect of training model in data parallel mode with different workers only iterating over a non-overlapping part of data.
- Effect of frequently dropping stale gradients.
- Effect of gradually updating the staleness parameter through the training process.
- Comparing the different modes of model training (Bulk Synchronous, Asynchronous, Asynchronous with dropping stale gradients)
- Effect of changing no. of workers in Asynchronous training.

For all the above points we will use the below evaluation yardsticks:

- Performance on the test set. (PTest)
- Time per epoch. (TEpoch)

Table 2 and 3 summarizes our observations for the above experiments on the two datasets respectively. For FB60K, due to its smaller size we have passed the entire data to all the workers. Doing this was not feasible for the FBCLUE dataset and hence for that we have followed data parallel paradigm.

in the above tables staleness_n corresponds to Asynchronous with dropping stale gradients with n being the staleness parameter.

Model$\|Observations$	PTest(AUC)	TEpoch (mins)
Serial	0.45	78
Synchronous	0.396	62
Asynchronous	0.3	42
staleness_10	0.389	42
staleness_3	0.381	42

Table 2. Experimental Observations for FBCLUE with Data Sharding

Model$\|Observations$	PTest(AUC)	TEpoch (mins)
Serial	0.415	12
Synchronous	0.410	48
Asynchronous	0.412	35
staleness_10	0.41	35

Table 3. Experimental Observations for FB60K without Data Sharding

For all the experiments in distributed settings we have used 2 parameter server nodes and 2 worker nodes. There are few more experimental settings we would like to perform which we hope to complete till the presentation. Primarily, we would like to see the impact of varying the no. of workers in Asynchronous mode and in keeping the staleness parameter in Asynchronous mode with dropping stale gradients.

Further, in figure 2 we present the rate of convergence of all the above models.

The above experiments gave us some idea how training model impacts the generalization performance of the trained models and their convergence rates. In the below experiments we aim to study how different workers tend to behave in data parallel mode with respect to each other in Asynchronous mode.

Figure 3 compares the time taken for each worker to make certain no. of optimization steps. While in figure 4 we present how under data sharding the jointly trained parame-

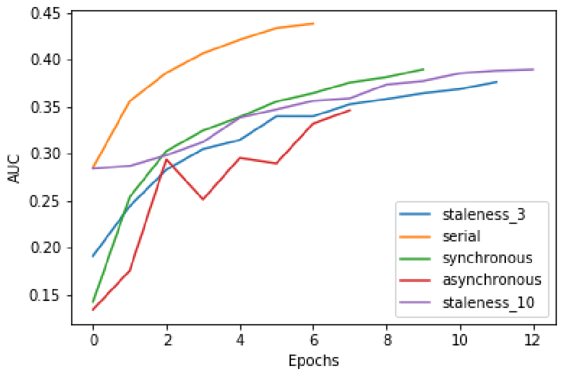

Figure 2. Rate of convergence for FBCLUE

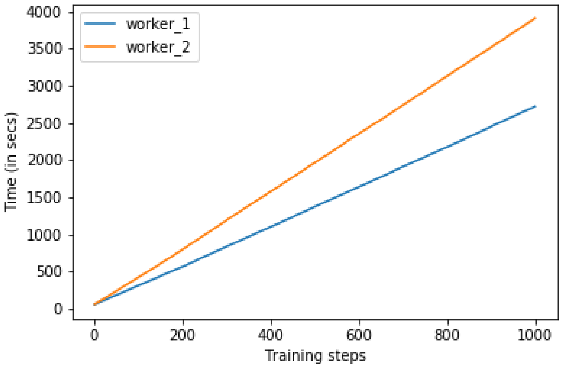

Figure 3. Time taken (sec) by each worker for FBCLUE in ASP mode for

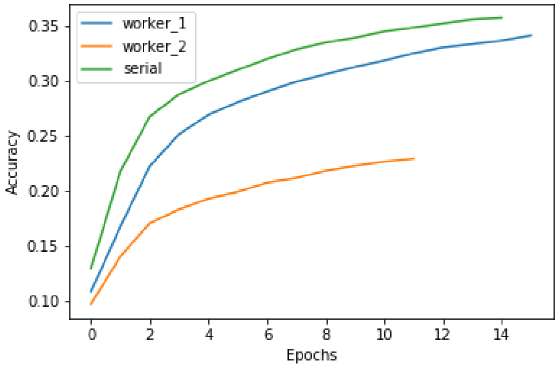

Figure 4. Accuracy on train set for FBCLUE in ASP mode

ters perform on the training data to belonging to different shards.

6. Discussions

The above set of experimental observations helps us to gain several incites about the distributed training paradigm in data parallel mode.

6.1. Impact of data sharding

- For all the experiments data sharding has resulted in a decrease in the performance as the results in FBCLUE show significant drop from serial to parallel training and no real drop in FB60K where both the workers are iterating over the entire data.

- Data sharding has significantly reduced the per epoch time for each case. Sharding the data also allowed us to preprocess sentences and directly store as index matrices to be loaded at run time, which is not feasible while using the entire data in each worker.

- Also it is observed in figure 4, that for the shared parameters the training performance significantly vary for the two shards. The performance on the data shard over which the slower worker operates is comparatively worse than the one iterated over by the faster worker. This reflects a variation in the data distribution for the two shards and also how the model tends to fit better over the part of the data which gets iterated over more number of times.

6.2. Comparing the distributed training paradigms

- As expected Bulk Synchronous model training leads to better generalization performance as compared Asynchronous.

- Dropping stale gradients have a positive impact on the training. But dropping too many gradients (with very small staleness parameter) have some negative impact.

- Bulk Synchronous models take significantly high per epoch training time.

- Dropping stale gradients with optimal staleness scheme can lead to most optimal results. But if two workers tend to be very different in their speed then a situation may arise where one worker is making no updates, discarding a shard of the data all together.

7. Future Work

- **Triple Margin Loss for RE:** Triple Margin Loss is quite common in KGC models which promotes Open Close assumption. We believe the same assumption should be true for RE as well compared to onevsall paradigm in Cross Entropy Loss. Moreover, in the new dataset there are 1345 different relation classes and it would be interesting to see how negative sampling based methods can be used such high no. of classification classes.

- **Multi-GPU:** Exploring the multi-gpu version of the model with parameter server and compare the results with the serial execution.

- **Stale synchronous:** Due to limitations of the high level distributed tensorflow API, we couldn't perform exhaustive experiments using stale synchronous mode.

- **KGC evaluation:** All the results shown so far are the evaluation for RE part of the model and KGC results can be evaluated in depth.

Acknowledgements

This work was made possible due to computing resources made available by the Department of Computational & Data Sciences at IISc, Bangalore.

Also, we would like to express our gratitude towards Dr. Partha Pratim Talukdar for his invaluable guidance throughout this project making this a great learning experience.

References

Bordes, A.; Usunier, N.; Garcia-Duran A.; Weston J.; and Yakhnenko, O. Translating embeddings for modeling multi-relational data. in nips, 27872795. 2013.

Hachey, B.; Radford, W.; Nothman-J.; Honnibal M.; and Curran, J. R. Evaluating entity linking with wikipedia. ai 194:130150. 2013.

Han, Xu, Liu, Zhiyuan, and Sun, Maosong. Neural knowledge acquisition via mutual attention between knowledge graph and text. 2018.

Lao, N. and Cohen. Relational retrieval using a combination of path-constrained random walks. proceedings of machine learning. 2010.

Lehmann, J.; Isele, R.; Jakob-M.; Jentzsch A.; Kontokostas D.; Mendes P. N.; Hellmann S.; Morsey M.; Van Kleef P.; Auer S. Dbpediaa large-scale, multilingual knowledge base extracted from wikipedia. semantic web6(2):167195. 2015.

Lin, Y.; Liu, Z.; Sun-M.; Liu Y.; and Zhu. Learning entity and relation embeddings for knowledge graph completion. in proceedings of aaai. 2015.

Lukovnikov, D.; Fischer, A.; Lehmann-J.; and Auer, S. Neural network-based question answering over knowledge graphs on word and character level. in www, 1211 1220. 2017.

Mintz, Mike, Bills, Steven, Snow, Rion, and Jurafsky, Dan. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pp. 1003–1011. Association for Computational Linguistics, 2009.

Mintz, M.; Bills, S.; Snow-R.; and Jurafsky, D. Distant supervision for relation extraction without labeled data. in proceedings of acl-ijcnlp. 2009.

Nickel, M.; Rosasco, L.; Poggio-T. A. Holographic embeddings of knowledge graphs. in proceedings of aaai. 2016.

Shi, B. and Weninger, T. Fact checking in heterogeneous information networks. in www, 101 102. 2016.

Socher, R.; Huval, B.; Manning-C. D.; and Ng, A. Semantic compositionality through recursive matrix-vector spaces. in proceedings of emnlp-conll. 2012.

Xu, Wei, Hoffmann, Raphael, Zhao, Le, and Grishman, Ralph. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pp. 665–670, 2013.