Towards Privacy-Preserving and Domain-Aware Knowledge Graph Entity Representations

Suparna Ghanvatkar, Jianxin Wei
School of Computing
National University of Singapore
{suparnag, jianxinwei}@comp.nus.edu.sg

Abstract

Knowledge Graphs have gained importance in Healthcare analytics for various prediction tasks. However, the domain presents us with two major issues - incompleteness of the data and highly-sensitive data. To handle these issues, we propose entity representation in Knowledge Graphs using entity descriptions and privacy preserving training methodology for preventing leak of the highly sensitive data. For the entity representation, our approach focuses on capturing the semantics in the description. We also propose that not all descriptions are equally useful and hence learn the contribution of the description in the entity representation, as opposed to the DKRL. As we have a trade-off between privacy and performance, we first focus on evaluating the effect of additive noise on model convergence. We call our approach a Differential Privacy simulation, which tries to approximate the differential privacy guarantee while still not being restrictive in the type of model. Thus, trying to understand and improve the sweet-spot between privacy and performance. We attempt to handle both these issues in a proof-of-concept fashion to limit the scope for a course project. Our approaches, though motivated by the healthcare domain, are evaluated on standard dataset, against a standard baseline to check if they provide improvements in simpler settings. Though both the issues are can be used many different downstream tasks, we restrict ourselves to the task of link prediction.

Introduction

In recent years, knowledge graph (KG) learning has prompted extensive research. Domains such as Healthcare Analytics use knowledge graphs to improve and support the predictions made by the models e.g. (Vlietstra et al. 2020; Liu et al. 2020). However, the clinical data presents us with an important issue of incomplete KG and highly sensitive content. The incomplete data could be due to missing information regarding the relation, or some entities themselves also might be missing, such as some mutation which has not yet been found. This poses the challenge of handling unseen entities not seen during training. There are two major school of thoughts for dealing with the incompleteness in KGs: logic-based systems, and link prediction systems which models the relations in an vector space. A major disadvantage of many popular models is they usually learn a lookup of the embeddings, and thus, fail when new entities are added (Yang et al. 2014; Bordes et al. 2013). The DKRL (Xie, Liu, and Sun 2016) model approaches this issue by adding the description to the embedding of the entities and builds on with the classical Trans-E (Bordes et al. 2013) model. However, this model drops the contextual and semantic information and treats the descriptions as Bag-of-Words model, and is thus outperformed when it is not treated as a BOW by using BERT embeddings (Daza, Cochez, and Groth 2020). Following the approach in (Daza, Cochez, and Groth 2020), we try to improve the context captured by the model by using Flair embeddings (Akbik, Blythe, and Vollgraf 2018) in place of the BERT embeddings.

We additionally hypothesize that not all entity descriptions are equally useful. A KG could have different types of entities, such as genes, drugs, proteins, etc. Each gene has a plethora of additional context to it, such as the possible non-standard variations, the frequency of occurrence of such variations, etc. Similarly, each drug has different contextual information attached to it. The descriptions for each of these entity types are from different sources and have varying degree of relevance and information in them (Gong et al. 2020). We propose a model to improve the information and context captured from the descriptions, by taking into account that not all descriptions are equally important.

For the second issue of highly sensitive data, we need to ensure the privacy of the patients' data when the model is released to public. Thus, we also focus our efforts on ensuring differential privacy. Differential privacy (Dwork 2008) defines a standard of privacy protection guarantees for arbitrary algorithms on databases. It prevents the privacy leakage caused by a slight change in the database. For instance, consider a database containing Bob, and 4 people in the database weigh more than 100kg, including Bob. Assume that some time in future only Bob was removed from the database. Then a simple repeat query on the database for the number of people with weight more than 100kg gives the response 3. This exposes Bob's private information of his weight being more than 100kg to everybody. This toy example helps understand the need of DP in the setting of healthcare where patients' sensitive data, along with clinical data may be used

for the entity representation using additional contextual information obtained from different sources.

For the task of being able to release the models such that privacy of the patients' data is maintained, we propose an approach for Differential Privacy (DP) simulation - which attempts to simulate the DP achieved, but is not very restrictive on the type of model it is applicable to. The reason we need to do simulation is because many operations, such as Batch Normalization and 2D convolution operations have privacy risk. This restricts the models we can have DP guarantees on. To achieve a privacy-preserving technique while using these popular models, we introduce our approximate version, which we call as DP-simulation.

To limit the scope for our course project, we do not focus on designing a completely private model. Rather, we focus on additive noise and evaluate how this would influence the convergence of our model. In the simulation, we apply Sampled Gaussian Mechanism (SGM) to protect data privacy. We focus on the privacy cost (ϵ) of the learning process, i.e., the stochastic gradient descent at each step. To simplify the analysis and realization, we assume that the module or layer such as Batch Normalization Layer of the learning model would not leak privacy even if they have privacy issues. In the DP computation, we apply a numerically precise procedure (Mironov, Talwar, and Zhang 2019) and introduce its transformation to the most commonly used (ϵ, δ)-DP.

The rest of the article is organized as follows: We first review the necessary background to understand our proposed approaches. We then explain our proposed model in both the aspects: the KG entity representation as well as the DP. Both of these aspects, though proposed with the healthcare context in mind, have been simplified in the Experiments section to have a fair evaluation. We then discuss the Related Work on both the topics and have subsequent Discussion on the results.

Preliminaries

DKRL Model

Description-embodied knowledge representation learning (DKRL) (Xie, Liu, and Sun 2016) extends the TransE model (Bordes et al. 2013) to handle the entity descriptions in addition to the KG triples. Each entity is associated with structure-based representation and the description-based representation. The structure-based representation uses the structured head, relation and tail (h,r,t) from a KG. The description-based representation is obtained using a convolutional neural encoder. Especially in the zero-shot scenario, DKRL has shown to perform much superior than a TransE model. This model introduces us to benefits of having two descriptions for an entity and including the textual descriptions of entities.

Flair Embeddings for entity descriptions

The Flair embeddings (Akbik, Blythe, and Vollgraf 2018) produce what the authors refer to as *contextual string embeddings*. These embeddings have shown an improved performance on a variety of NLP tasks, such as sentiment analysis to offensive language detection (Akbik et al. 2019).

The BERT for Link Prediction (BLP) model (Daza, Cochez, and Groth 2020) uses BERT for entity encoding, though they mention any Transformer model can be used. However, Flair embeddings help to capture word meanings in context, allowing us to embed polysemous words effectively. Also, the Flair embeddings fundamentally model a sequence of characters and thus have forward and backward models for viewing the sequence of characters. In the Flair embeddings paper (Akbik et al. 2019) it has been found that the combination of Glove embeddings along with forward and backward Flair embeddings usually produces state-of-the-art results.

Differential Privacy

To formally define Differential Privacy (DP), we need to define adjacent databases first. Given two databases $\mathcal{D}$ and $\mathcal{D}'$, if they differ in only one tuple, i.e., one data tuple is contained in one database and absent in the other, we call these two adjacent databases.

Definition 1. A randomized algorithm $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ provides (ϵ, δ)-differential privacy guarantee if the following inequality holds.

$$\Pr[\mathcal{A}(d) = O] \leq e^{\epsilon} \Pr[\mathcal{A}(d') = O] + \delta$$

where $d, d' \in \mathcal{D}$ are adjacent data sets and $O \subseteq \mathcal{R}$.

In other words, if the algorithm is applied to any adjacent data sets, the probability of obtaining a specific output should be similar. Observers can hardly detect a slight change in the data set by observing the output results. So, we achieve the purpose of protecting privacy.

Then how can we employ DP? The simplest method is adding randomized noise to the input, output, or intermediate results, in order to cover up the real data. Laplace noise is one of the most commonly used type of additive-noise because the mathematical characteristic of the Laplace distribution exactly fits with the definition of DP. Another favorable additive-noise is Gaussian noise, which is useful when the distance of data tuples is measured by l2 norm.

There are several properties such as composability and robustness that make DP increasingly popular in applications. Robustness means the guarantee would not be affected by any side information and composability allows us to connect differentially private modules without privacy leakage.

In the scenario of deep learning, a model will be released after training where a single data tuple would affect thousands of training steps with some probability. Therefore, we need to protect privacy for each step and analyse the privacy guarantee of the whole process by composition theorem. However, for iterative algorithms especially the deep learning algorithms with additive-noise mechanisms, the basic composition theorem (Dwork and Lei 2009) and even the advanced composition theorems or refinements (Dwork, Rothblum, and Vadhan 2010; Dwork and Rothblum 2016) would consume too much privacy budget so that they are not applicable.

The moments accountant (Abadi et al. 2016) technique based on additive Gaussian noise mechanism provides a

tighter bound than previous composition theorems and significantly decrease the amount of noise. The strong composition theorem saves a factor of $\sqrt{\log(T/\delta)}$.

Rényi Divergence

In our project, instead of using the asymptotic bound, we apply a numerically stable procedure for more precise computation of DP, called Rényi Differential Privacy (RDP) (Mironov, Talwar, and Zhang 2019), whose computation is based on the Rényi Divergence.

Definition 2. Given two distributions P and Q on $\mathcal{X}$ defined over the same probability space. Let p(x) and q(x) be the probability density functions. The Rényi Divergence of a order $\alpha \neq 1$ is defined as ($\alpha = 1, \infty$ are defined by continuity)

$$\mathrm{D}_\alpha(P\|Q) \triangleq \frac{1}{\alpha - 1} \ln \int_{\mathcal{X}} q(x) \left(\frac{p(x)}{q(x)}\right)^\alpha \mathrm{d}x$$

Definition 3 (RDP)**.** A randomized algorithm $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (α, ϵ)-RDP if the following inequality holds.

$$\mathrm{D}_\alpha\left(\mathcal{A}(d) = O \| \mathcal{A}(d') = O\right) \leq \varepsilon$$

where $d, d' \in \mathcal{D}$ are adjacent data sets and $O \subseteq \mathcal{R}$.

We will illustrate the details of privacy cost computation in the next section.

The Proposed Model

Our model focusses on two aspects: adding contextual information and creating privacy-preserving embeddings. We propose novelty in both these aspects, which is evaluated separately. Thus, we explain our model in two subsections to maintain coherency with our evaluations.

Entity Representation

Similar to DKRL, we propose to have a description-based representation and structure-based representation. In lines with our hypothesis about the contextual information, we use a stack of Flair embeddings in forward and backward directions, along with Glove embeddings. However, rather than following the DKRL approach of using CNN, the description-based representation is trained using an LSTM autoencoder to ensure that the representation created can faithfully capture all the important information in the description, without losing the semantics. A weighted combination of this description-based representation and structure-based representation is learnt to obtain a entity representation. This weighted combination is meant to ensure that not every description plays an equally important role in the entity representation. Following the translational assumption in TransE, we assume that the relation r is a translational vector such that the entity representations thus obtained are connect by r with low error.

As compared to the two approaches we build on, namely DKRL and BLP, our proposed model firstly incorporates contextual information using Flair embeddings. Secondly, we propose that not all descriptions are equally useful and hence rather than just using description-based representation as done in BLP, we learn the contribution of the description to the entity.

Let (h, r, t) denote the head, relation and tail of the triple under consideration. The additional text description is denoted by d. The output of a contextual embedding using stacking of Glove, and Flair forward and background can be then obtained as for a sequence of length n tokens.

$$Emb_1(d) = [e_1, e_2, ..., e_n]$$

An LSTM Autoencoder trained indepenently produces a lower dimension representation of this $Emb_1(d)$, which we can denote with f. Let d_{dim} represent the dimension of this representation. Similarly, an embedding for the h, r, t is also obtained as a d_{dim} dimensional vector. So, for each triple we get the following vectors:

$$Emb_{ent}(h) = h_s, h_s \in \mathcal{R}^{d_{dim}}$$
$$Emb_{rel}(r) = r_s, r_s \in \mathcal{R}^{d_{dim}}$$
$$Emb_{aec}f_h = h_t, h_t \in \mathcal{R}^{d_{dim}}$$

Here the f_h represents the Emb_1 of the description for head entity while f_t denotes the same for the tail entity. Now, we learn a weighted combination of the two entity representations.

$$ent_h = w_1.h_s + w_2.h_t$$

We continue this with the TransE formulation and do the loss computation and training accordingly.

$$f_r(ent_h, ent_t) = -\|ent_h + ent_t - r_s\|_{1/2}$$

As the Flair and Glove embeddings are pre-trained, and the LSTM Autoencoder is trained separately, these are not added in our optimization. However, we propose that training the LSTM autoencoder jointly might improve the performance. Due to time constraint though, we do not focus and evaluate the model with joint training.

DP Simulation

For the link prediction models, such as DKRL-BERT, only working on the input training data or the output parameters by adding noise would damage the utility of the model (performance) and would be hard to analyse. Therefore, we turn to control the influence of the training data during the training process, specially in the stochastic gradient descent (SGD) step. We assume the model parameters as θ and the objective loss function as $\mathcal{L}(\theta)$. Then for a random sampled data tuple x_i, its gradient is $\nabla_\theta \mathcal{L}(\theta, x_i)$. Moreover, we define a gradient norm bound C, which is necessary to bound the sensitivity of the algorithm for privacy analysis. Next we describe the steps of our privacy preserving algorithm.

1. Input: data set $\mathcal{D} = \{x_1, \cdots, x_N\}$; model parameters θ; loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$; learning rate η; gradient norm bound C; Gaussian noise scale δ; total traning steps T.
2. For each step $t \in T$,
   (a) Sample a batch of data tuples from $\mathcal{D}$ with a sampling probability $q = B/N$ of each tuple.

(b) For each x_i the batch, compute its gradient $g_t(x_i) := \nabla_\theta \mathcal{L}(\theta, x_i)$

(c) Clip the each gradient by $\overline{\mathbf{g}}_t(x_i) := \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}t(x_i)\|_2}{C}\right)$

(d) Add noise to the gradients $\tilde{\mathbf{g}}_t := \frac{1}{B}\left(\sum_i \overline{\mathbf{g}}_t(x_i) + \mathcal{N}\left(0, \sigma^2 C^2 \mathbf{I}\right)\right)$

(e) Update parameter $\theta_{t+1} := \theta_t - \eta \cdot \tilde{\mathbf{g}}_t$

3. Compute the DP parameters ϵ, δ of the whole training process.

4. Output: Model parameters θ_T and DP parameters ϵ, δ.

For simplifying our analysis, we assume the module or layer of the learning model would not leak any privacy even if they have privacy issues. We mainly focus on the influence of noisy training to the convergence and performance of the model.

Next we illustrate how to compute ϵ, δ. For the above algorithm named as $\mathcal{A}$, we can compute the ϵ with an order α as follows according to (Mironov, Talwar, and Zhang 2019).

$$\begin{aligned}
&\mathrm{D}_\alpha\left(\mathcal{A}(d') = O \| \mathcal{A}(d) = O\right) \\
&\leq \mathrm{D}_\alpha\left((1-q)\mathcal{N}\left(0, \sigma^2\right) + q\mathcal{N}\left(1, \sigma^2\right) \| \mathcal{N}\left(0, \sigma^2\right)\right) \\
&= \frac{1}{\alpha - 1} \ln\left(\sum_{k=0}^{\alpha} \binom{\alpha}{k} (1-q)^{\alpha-k} q^k \exp\left(\frac{(k^2 - k)}{\sigma^2}\right)\right) \\
&= \epsilon
\end{aligned} \tag{1}$$

where $d, d' \in \mathcal{D}$ and $O \subseteq \mathcal{R}$ are same as the previous definition. The order α is a hyper-parameter and it can be selected in a wide range. The optimal one minimize the ϵ parameter.

Now we get the (α, ϵ)-RDP. For the computation of δ, we apply an improved tighter bound proved by (Canonne, Kamath, and Steinke 2020) as follows.

Theorem 1. Given a randomized algorithm $\mathcal{A}$, let $\alpha \in (1, \infty)$ and $\epsilon > 0$. Suppose $\mathrm{D}_\alpha\left(\mathcal{A}(d') = O \| \mathcal{A}(d) = O\right) \leq \tau$ for adjacent $d, d' \in \mathcal{D}$ and $O \subseteq \mathcal{R}$. Then A is (ϵ, δ)-DP for

$$\delta = \frac{e^{(\alpha-1)(\tau-\varepsilon)}}{\alpha - 1} \cdot \left(1 - \frac{1}{\alpha}\right)^{\alpha} \tag{2}$$

The improved bound is strictly better than the basic bound (Mironov 2017) for $\alpha > 1$. It helps us approximate (ϵ, δ)-DP from (α, ϵ)-RDP.

Experiments and Results

Data

Though the aim is to evaluate in the healthcare context, for the scope of the course project, we simplify our problem and focus on first evaluating our proposed model on standard dataset. Though both our tasks, entity representation, and privacy preservation can apply to many downstream tasks, we select the task of link-predictions to compare with the DKRL model. We use FB15k-237, a subset of Freebase, which is a dataset widely used in link prediction literature. We build our models on top of the code and data used in (Daza, Cochez, and Groth 2020). Following the authors, we have also followed a maximum length of entity descriptions to be 32 tokens. We follow the same train-test split of 11,633-1,454 entities as used by the authors. The same three evaluation metrics of MRR, Hits@1, Hits@3 and Hits@10 are used for the task of link prediction. The KG entity representation part replaces the codebase with Flair API[1] to have fair evaluation for addition of Flair API. On the other hand, DP simulation part uses the original code base for the DKRL-BERT.

Proof-of-concept for adding context to KG entity representation

There are three changes from DKRL to this model: first is the changing of the embedding from BERT to stacked Flair embedding, second is the change in the description-based representation, and third is the weighted combination of the two representations. We run two experiments to isolate and understand the effects of these changes. First, in the DKRL model itself, we change the embeddings and evaluate against the BERT-based DKRL model. This model is represented by DKRL-Flair which uses CNN for embedding the text, but just replaces the embeddings with the StackedEmbeddings of Glove, news-forward Flair and news-backward Flair. Due to restrictions in time and compute-resources, we train both the models for 30 epochs and report the comparison.

The second experiment uses these StackedEmbeddings along with the LSTM autoencoder from the torchcoder repository[2] to generate the description-based representation. This is then combined with the structure-based representation using weighted combination. Thus, this experiment is meant to evaluate the second and third change in the DKRL model. This model is represented as Torchcoder-AEC. This model requires more training-time than the previous models, but we train the model initially for 10 epochs due to time-constraint, as each epoch roughly takes 4 hours per epoch on 40 core CPU server. The initializations for entity embeddings follow Xavier initialization, based on the optimal initialization found by (Daza, Cochez, and Groth 2020). However, hyper-parameter tuning of these models would be needed for a rigorous comparison. The results of initial evaluation are presented in Table 1.

Model	MRR	Hits@1	Hits@3	Hits@10
DKRL-BERT	0.082	0.046	0.080	0.145
DKRL-Flair	0.098	0.052	0.096	0.184
Torchcoder-AEC	0.013	0.0004	0.0019	0.0054

Table 1: Comparison of KG Link Prediction

Proof-of-concept for DP simulation

For our DP simulation, we employ SGM on the DKRL-BERT model. We use the same batch_size and learning for convenient comparison with non-DP training. We do a grid

[1] https://github.com/flairNLP/flair

[2] https://github.com/hellojinwoo/TorchCoder

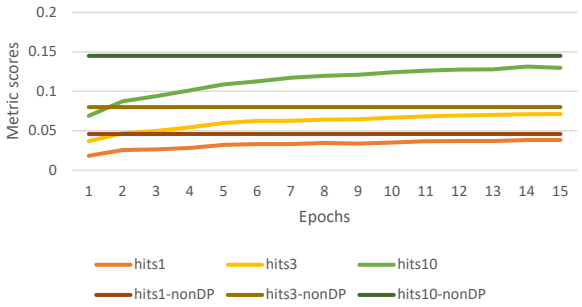

Figure 1: Metric scores over epochs in DP and non-DP settings

search in range $[1, 64]$ with step size 1 to find the best α and δ is set to $1/N$, where N is the size of training set. We train the model for 15 epochs because the ϵ will grow larger and larger after each epoch and an too large ϵ (e.g. $\epsilon > 7$) is useless (its privacy protection is extremely weak). The noise scale σ of Gaussian noise is set to two values, $\sigma = 1.0, 1.2$.

Noise	MRR	Hits@1	Hits@3	Hits@10	ϵ
$\sigma = 1.0$	0.067	0.034	0.064	0.124	6.49
$\sigma = 1.2$	0.063	0.033	0.060	0.112	4.72

Table 2: Test results of different noise scale

The prediction results are shown in Table 2. We can observe that the DP mechanism drop each metric by about $1 - 3\%$. Although the model with $\sigma = 1.2$ performs worse than $\sigma = 1.0$, it gets a more acceptable ϵ. The variation of metric scores over training epochs in DP and non-DP settings are shown in Figure 1. We can find that the model can converge even with a relatively large scale of noise ($\sigma = 1.0$) and its performance is close to the non-DP model in a short training period (training epoch=15).

Actually, the performance of the model with SGM can be finely tuned better if we try plenty of hyper-parameters. Due to the affect of Gaussian noise, the DP model should use different batch_size and learning from the non-DP model. Abadi et al. (Abadi et al. 2016) provides a bound for the number of experiment settings that should be tested before we get an acceptable one.

Related Work

Related Work on Entity Representations

As noted by (Wang et al. 2017) in their survey paper, the symbolic *(head, relation, tail)* representation of KG has it's set of limitation, such as manipulation and changes in KG. Due to this, recent years have seen a surge in the topic of *knowledge graph embeddings* (Wang et al. 2014; Daza, Cochez, and Groth 2020; Yang et al. 2014), to be able to use these learnt embeddings for a variety of different downstream tasks. In addition to the structured information from the KG, many studies have started using additional information such as the type of entities (Xie, Liu, and Sun 2016; Guo et al. 2015), multi-hop relations (Guu, Miller, and Liang 2015), textual descriptions (Wang et al. 2016), logical rules on relations (Guo et al. 2016), and temporal information (Trivedi et al. 2017). For a recent survey on the entity representations in KGs, please refer to (Ji et al. 2020). For further reading on text descriptions used for entity representation, please refer to (Lu, Cong, and Huang 2020).

The papers closest to our approach are the DKRL and the BLP models. Both these models focus on capturing the textual descriptions, but the DKRL does not keep the semantics, while the BLP approach captures the semantics using BERT embeddings. As noted in (Daza, Cochez, and Groth 2020), the type of model to use for evaluation does not matter, i.e. whether the translational model , or the multiplicative models, such as DistMult (Yang et al. 2014) is used. The authors evaluate the generalization properties of these embeddings trained for link-prediction task. Though in this project we focus on the task of link-prediction, our aim is to help the application of KG in various tasks in the healthcare setting. This is why we chose the BLP paper (Daza, Cochez, and Groth 2020) as our base.

Related Work on Differential Privacy in Learning Algorithms

Differential Privacy is widely applicable in database systems and algorithms. The Sampled Gaussian mechanism (SGM) is the most popular mechanism in the area of learning algorithms. It consists of two parts, sampling and additive Gaussian noise. Both components are worth researching. Multiple methods (Kasiviswanathan et al. 2011) build block of differentially private mechanisms for sampling a random subset from a large data set. The additive Gaussian noise was first discussed by (Dwork et al. 2006) while there was no major progress of its application on deep learning until the moments accountant technique (Abadi et al. 2016) proposed a refined analysis of privacy costs in the framework of neural networks. Meanwhile, Concentrated Differential Privacy (CDP) (Dwork and Rothblum 2016) developed refined composition theorems for the Gaussian mechanism. The reformulation of CDP such as zero-CDP (Bun and Steinke 2016) and truncated CDP (Bun et al. 2018) provided a relaxation of CDP. Furthermore, instead of applying asymptotic bound for analysis, (Mironov, Talwar, and Zhang 2019) described a numerically stable procedure for precise computation of privacy costs in SGM based on the notion of Rényi Differential Privacy (RDP) (Mironov 2017).

Discussion

There are some major limitations in the study design and results reported here. Firstly, the results reported in this article are preliminary and we would expect improvements on more fine-tuning. Secondly, though our motivation is drawn from the healthcare domain, we have focussed on a proof-of-concept type evaluation. Our evaluation in this project has thus been to evaluate if our model makes a difference in a standard database. This is because KGs such as SemMedDB (Kilicoglu et al. 2012) are more complicated and noisy. It connects every entity to nearly every other entity through

some path. This makes the KG extremely challenging. Thus, even feasibility of our model is not evaluated on the actual domain KG, and we do not know if the domain inherently poses some challenges for this approach.

The results in the entity representation section show some improvement between DKRL-BERT and DKRL-Flair model. However, it is very small, and as noted earlier, these results are not tuned and have a scope of improvement for both the models. As far the Torchcoder-AEC model is considered, the amount of training for the model was very limited. We did a sanity check on the model by taking a small subset of data and overfitting. This sanity check passed indicating that the model indeed is learning something. However, the model had not converged in the few epochs it was trained for, and more compute-resource and time would be need to tune the hypoer-parameters and train the model until convergence.

The results in our DP section are encouraging as compared to the entity representation. However, we note that we do not have theoretical privacy guarantees for our approach. We have tried to approximate the computations such that they are applicable for the models we are considering in KG link prediction. To realize a strict DP model, we should replace the Batch Normalization layer or other such layers to avoid privacy leakage issues. Keeping in spirit of the DP guarantees, we wanted to do the first step towards applying DP to these KG link prediction models. As the first step, i.e. convergence of the models on addition of the noise seems encouraging, it would be fruitful to evaluate more rigorously now how to guarantee privacy bound, while working on the models which perform well. Also, it would be interesting to evaluate the DP simulation on our torchcoder-aec model, as it could help experiment and realize the privacy-performance tradeoff and help find a sweet spot for providing privacy guarantees such that we do not loose out on performance.

References

[Abadi et al. 2016] Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 308–318.

[Akbik et al. 2019] Akbik, A.; Bergmann, T.; Blythe, D.; Rasul, K.; Schweter, S.; and Vollgraf, R. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 54–59.

[Akbik, Blythe, and Vollgraf 2018] Akbik, A.; Blythe, D.; and Vollgraf, R. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, 1638–1649.

[Bordes et al. 2013] Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*, 1–9.

[Bun and Steinke 2016] Bun, M., and Steinke, T. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, 635–658. Springer.

[Bun et al. 2018] Bun, M.; Dwork, C.; Rothblum, G. N.; and Steinke, T. 2018. Composable and versatile privacy via truncated cdp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 74–86.

[Canonne, Kamath, and Steinke 2020] Canonne, C.; Kamath, G.; and Steinke, T. 2020. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010*.

[Daza, Cochez, and Groth 2020] Daza, D.; Cochez, M.; and Groth, P. 2020. Inductive entity representations from text via link prediction. *arXiv preprint arXiv:2010.03496*.

[Dwork and Lei 2009] Dwork, C., and Lei, J. 2009. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 371–380.

[Dwork and Rothblum 2016] Dwork, C., and Rothblum, G. N. 2016. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*.

[Dwork et al. 2006] Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; and Naor, M. 2006. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 486–503. Springer.

[Dwork, Rothblum, and Vadhan 2010] Dwork, C.; Rothblum, G. N.; and Vadhan, S. 2010. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 51–60. IEEE.

[Dwork 2008] Dwork, C. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, 1–19. Springer.

[Gong et al. 2020] Gong, F.; Wang, M.; Wang, H.; Wang, S.; and Liu, M. 2020. Smr: Medical knowledge graph embedding for safe medicine recommendation.

[Guo et al. 2015] Guo, S.; Wang, Q.; Wang, B.; Wang, L.; and Guo, L. 2015. Semantically smooth knowledge graph embedding. 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)*, 84–94.

[Guo et al. 2016] Guo, S.; Wang, Q.; Wang, L.; Wang, B.; and Guo, L. 2016. Jointly embedding knowledge graphs and logical rules. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, 192–202.

[Guu, Miller, and Liang 2015] Guu, K.; Miller, J.; and Liang, P. 2015. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*.

[Ji et al. 2020] Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv e-prints* arXiv–2002.

[Kasiviswanathan et al. 2011] Kasiviswanathan, S. P.; Lee, H. K.; Nissim, K.; Raskhodnikova, S.; and Smith, A. 2011.

What can we learn privately? *SIAM Journal on Computing* 40(3):793–826.

[Kilicoglu et al. 2012] Kilicoglu, H.; Shin, D.; Fiszman, M.; Rosemblat, G.; and Rindflesch, T. C. 2012. Semmeddb: a pubmed-scale repository of biomedical semantic predications. *Bioinformatics* 28(23):3158–3160.

[Liu et al. 2020] Liu, Y.; Elsworth, B. L.; Erola, P.; Haberland, V.; Hemani, G.; Lyon, M. S.; Zheng, J.; and Gaunt, T. R. 2020. Epigraphdb: a database and data mining platform for health data science. *BioRxiv*.

[Lu, Cong, and Huang 2020] Lu, F.; Cong, P.; and Huang, X. 2020. Utilizing textual information in knowledge graph embedding: A survey of methods and applications. *IEEE Access* 8:92072–92088.

[Mironov, Talwar, and Zhang 2019] Mironov, I.; Talwar, K.; and Zhang, L. 2019. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*.

[Mironov 2017] Mironov, I. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 263–275. IEEE.

[Trivedi et al. 2017] Trivedi, R.; Dai, H.; Wang, Y.; and Song, L. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *International Conference on Machine Learning*, 3462–3471. PMLR.

[Vlietstra et al. 2020] Vlietstra, W. J.; Vos, R.; van den Akker, M.; van Mulligen, E. M.; and Kors, J. A. 2020. Identifying disease trajectories with predicate information from a knowledge graph. *Journal of biomedical semantics* 11(1):1–11.

[Wang et al. 2014] Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

[Wang et al. 2016] Wang, Z.; Li, J.; Liu, Z.; and Tang, J. 2016. Text-enhanced representation learning for knowledge graph. In *Proceedings of International Joint Conference on Artificial Intelligent (IJCAI)*, 4–17.

[Wang et al. 2017] Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29(12):2724–2743.

[Xie, Liu, and Sun 2016] Xie, R.; Liu, Z.; and Sun, M. 2016. Representation learning of knowledge graphs with hierarchical types. In *IJCAI*, 2965–2971.

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