Fine-grained Entity Type Inference in RDF Knowledge Graphs

A. B. M. Moniruzzaman
Queensland University of Technology
Brisbane, Queensland, Australia
a.moniruzzaman@qut.edu.au

Abstract. Knowledge Graphs (KGs) have been proven to be incredibly useful for enriching semantic Web search results and allowing queries with a well-defined result set. In recent years much attention has been given to the task of inferring missing facts based on existing facts in a KG. Approaches have also been proposed for inferring types of entities, however these are successful in common types. There is still a large gap, however, in the inference of fine-grained types which are highly important for exploring specific lists and collections within web search. Generally there are also relatively fewer observed instances of fine-grained types present to train in KGs, and this poses challenges for the development of effective approaches. In order to address the issue, this paper proposes a new approach to the fine-grained type inference problem. This new approach is explicitly modeled for leveraging domain knowledge and utilizing additional data outside KG, that improves performance in fine-grained type inference. Further improvements in efficiency are achieved by extending the model to probabilistic inference based on entity similarity and typed class classification. Our aim is to automatically identify entities to be semantically interpretable by having fine-grained types in Knowledge Graph. We demonstrate the benefits of our task in the context of fine-grained entity type inference applying on DBpedia, and by producing a large number of resources in different fine-grained entity types for connecting them to DBpedia type classes.

Keywords: Knowledge Graph, Knowledge Base, Fine-grained Type Inference, Tensor Factorization, Semantic Web Search

1 Problem Statement

Recent years have witnessed a rapid growth of knowledge graphs (KGs) such as Freebase, DBpedia, or YAGO. These KGs store billions of facts about real-world entities (e.g. people, places, and things) in the form of triples. KGs are playing an increasingly important role in enhancing the intelligence of Web and enterprise search and in supporting information integration and retrieval. Linked Open Data (LOD) cloud interlinks KGs and other data sources using the W3C Resource Description Framework (RDF) and makes accessible on web querying.
Despite these impressive advances, there are still major limitations regarding coverage with missing information, such as type, properties, and relations.

Types in KG are used to express the concept of classes. According to KG idiomatic usage, a KG object "has X, Y, Z types" is equivalent to an object "is a member of the X, Y, Z classes". In the case of Tom Hanks, the KG object would have the types person and Actor to indicate that the object is a member of the Persons and Actors. However, an entity is usually not associated to a limited set of generic types (Person, Location, and Organization) in KGs but rather to a set of more specific (fine-grained) types.

In recent years much attention has been paid to KG completion tasks, which automatically infer missing facts and missing types in a KG. Although some approaches have been proposed for type inference in KG, these are successful in extracting common types, such as 'Person', 'Artist', 'Movie' or 'Actor'. However, there is still a big gap in fine-grained types of entities to be inferred that is highly important for exploring specific lists and collections with web searching. Generally, very limited numbers of observed data for fine-grained types present in KGs, and this poses a high challenge in developing approaches for inferring entities of the fine-grained types. In order to address the issue, this paper proposes a new approach based on tensor factorization and probabilistic inference to the fine-grained type inference problem.

To present the model for knowledge graph fine-grained type inference task, we denote KG triples by \((e_s, r, e_o)\), where \(e_s\) and \(e_o\) denote the subject and object entities, respectively. We build model that rank candidate entities for given queries \((e_s, r, ?)\) which ask about the object of a given fine-grained type relation. The model we build score possible triples \((e_s, r, e_o)\) using continuous representations (latent features) of the three elements of the triple from joint tensor factorization of KG and textual-based information. The model we use rank of triples \((e_s, r, e_o)\) to represent the model’s confidence in the existence of the triple.

2 Relevancy

Recent years have witnessed a rapid growth of KGs driven by academic and commercial efforts, such as Yago, Freebase, NELL, Google’s Knowledge Graph, Microsoft’s Satori, Probase, and Google Knowledge Vault. These KGs have reached an impressive size, for instance, DBPedia a large-scale KG extracted from Wikipedia contains many millions of entities, organized in hundreds to hundred thousands of semantic classes, and billions of relational facts (triples) involving a large variety of predicates (relation types) between entities. KGs are playing an increasingly important role in enhancing the intelligence of Web and enterprise search and in supporting information integration and retrieval. For example, Freebase KG powered Google Knowledge Graph that supports Google’s web search, or Microsoft’s Satori that supports Bing by providing

\(^2\) http://dbpedia.org/resource/Tom_Hanks
richer data for Entity Pane, Carousel, and Facts Across Segments in the search panel. Additionally, KGs are becoming important resources for different Artificial Intelligence (AI) and Natural Language Processing (NLP) applications, such as Question-Answering Query Understanding through Knowledge-Based Conceptualization, and Short Sentence Texts Understanding and Conceptualization using a probabilistic Knowledge bases. The Semantic Web’s Linked Open Data (LOD) cloud interlinks KGs and other data sources using the W3C Resource Description Framework (RDF) and makes accessible on web querying through SPARQL. This LOD cloud is growing rapidly. At the time of this writing, the LOD cloud contains 1,234 datasets with 16,136 links \(^1\). Several hundred data sets on the Web publish RDF links pointing to DBpedia themselves and thus make DBpedia one of the central interlinking hubs in the Linked Open Data (LOD) cloud. Despite these impressive advances, there are still major limitations regarding coverage and freshness, these KGs are incomplete with missing information, such as type, properties, and relations \([15]\) \([16]\) \([21]\) \([6]\) \([27]\) Defining fine-grained types of entities in KG allows Web search queries with a well-defined result set. Evidence suggests that performance of Web search queries (in case of exploring lists and collections) can be dramatically improved by defining large numbers of these fine-grained entity types in KG.

3 Related work

Given the importance of missing type assertions task, there are some approaches based on Reasoning, Probabilistic Method \([23]\), Hierarchical Classification Approach \([13]\) \([14]\), Association Rule Mining \([8]\) Topic Modelling \([25]\) and Tensor Factorization \([18]\) \([15]\) have been proposed and applied on different in KGs. In \([19]\) authors exploit interlinks between the KGs and Wikipedia predict types in a KG. In \([20]\), exploit such association rules to predict missing types in DBpedia based on redundancies different type systems of KGs. As topics, the interlinks and the type redundancies between KGs can only be covered limited associated top level types, these approaches are not capable for fine-grained type inference. Paulheim, H. and Bizer, C. proposed SDType algorithm \([22]\) \([23]\) a probabilistic method for inferring missing type of entities in KGs. Since this approach is based on statistical method and there are not enough facts exits in KG for fine-grained types \([22]\), \([23]\), this approach is not capable to produce meaningful results with this approach. Melo et al. proposed SLCN \([13]\) algorithm which is based on top-down class prediction strategy with local classifier per node (LCN) approach \([13]\). They assumed the types to be structured in a hierarchy, and it is a multilabel problem because instances are allowed to have more than one type \([13]\) \([24]\). Since fine-grained types basically are not systematically structured with a type hierarchy in KGs and SNCN relay on the hierarchical type structure in KG, this approach is not efficient in fine-grained type inference. Besides, some KGs (such as Freebase) are not systematically structured with a type hierarchy, in this case, this approach is not useful for this task. Though, SDType, and

\(^1\) https://lod-cloud.net/
SNCN algorithm is successful in predicting higher-level classes (Actor), while predicting fine-grained classes (such as VoiceActor or AmericanActor) is much more difficult. None of these methods are able to capture the interactions of latent features that obtain probabilistic likelihoods of type-relations existing between entities (objects) in KG. Tensor modeling, a well-known approach to represent and analyze latent relationships inherent in multi-dimensional data [10]. RESCAL [17] [18] is the state-of-the-art method for link prediction and type inference in KGs that follows DEDICOM tensor factorization model. The key idea behind factorization method RESCAL is that it uses three-dimensional arrays (tensor) to represent KGs and obtain probabilistic likelihoods of type-relations existing between entities (objects) by applying tensor factorization (TF) techniques on KGs. It defines statistical models for modeling tensor representation of binary relational data on KGs and explains triples via pairwise interactions of latent features. RESCAL has been used for type inference on YAGO entire KG [18]. Though, RESCAL has been achieved good performance among the state-of-the-art approaches for type inference in KG, the performance of type inference (fine-grained types) are significantly lower than the performance of top level (common) types in KG. Furthermore, this method and other existing methods for type inference in KG are not able to utilize the domain knowledge in KG and the additional sources of information outside of KG. Our approach is based on tensor factorization method which is specially designed for fine-grained type inference in KG. This approach is able to capture interactions of entities outside of KG for efficient fine-grained type inference by joint factorization of KG and Text. Furthermore, this model is designed to learn type-domain knowledge and type hierarchy structure; and utilize these information to improve fine-grained type inference results in KG.

4 Research Questions

**RQ1:** How to represent knowledge graph to apply tensor factorization method for fine-grained type inference?

**RQ2:** How to model knowledge graph for leveraging domain knowledge in fine-grained type inference?

**RQ3:** How to model knowledge graph for utilizing additional data outside KG in fine-grained type inference?

**RQ4:** How to improve fine-grained type inference efficiency by probabilistic inference model?
5 Hypotheses

5.1 Representation of Knowledge Graph

From modelling perspective, tensor representations are appealing to RDF knowledge graphs because they provide an elegant way to represent multiple RDF triples. The populated tensor is then represented the collaborative activities of the type domain can be modelled subject, object, predicate and domain in 4th-order tensor presentation as illustrated in fig. 1.

Figure 1: (a) 4th-order tensor representation. (b) Tensor entries.

The incorporation of Text data for type inference in KG becomes possible via a tensor factorization model where KG and text data consisting of e entities and r different relations can then be represented in form of a tensor. The values (’2’ or ’1’) and (’0’) of $X_{ijk}$ come from tensor model are regarded as observed and un-observed data respectively for tensor factorization.

5.2 Leveraging Domain Knowledge

Most KGs (such as DBPedia, Freebase, or Yago) store facts about real-world objects covering only numbers of specific domains (e.g. ”Movie”, ”Book”, or ”Place”).

Figure 2: (left) Representation of a KG with different domains. (right) DrWT method mapping from relations in KG.
For instance, types in KG such as actor, film, director, or producer and fine-grained types such as filmActor, TVActor, regularActor, guestActor, executiveDirector, AssistantProducer are in "film" domain. Given the importance the fine-grained inference task in KG, typed entities (objects) for given fine-grained types in one domain (such as "film" domain) are less likely to be entities in other domains (such as "book", or "place"). For instance, inferring entities for fine-grained types (such as regularActor, guestActor) would be a typed in Actor, those entities generally are in same domain in KG. The collaborative activities between the entities in "film" domain are therefore higher importance for fine-grained type inference in this domain.

5.3 Utilizing Additional Data

Semantic Web KGs are represented in the form of the Resource Description Framework (RDF) triples <subject, predicate, object>, where the subject and object are entities and the predicate is the relation type, (e.g. <Steve Jobs, founderOf, Apple>). Uniform Resource Identifier (URI) uniquely identifying each entity in Semantic Web KGs. For instance, entity Donald Trump can be found as http://dbpedia.org/page/Donald_Trump in DBpedia, and in Wikipedia as https://en.wikipedia.org/wiki/Donald_Trump. Every entity in Semantic Web KGs (e.g. DBpedia or Freebase) has anchor link to Wikipedia page, where a document of information is available for an associated entity.

Each entity in Freebase is uniquely identified by a MID (Machine ID). For instance, /m/0cqt90 as MID is used for Donald Trump entity [?]. Using MID and the predicate http://www.w3.org/2002/07/owl#sameAs for each entity in Freebase, we can issue a SPARQL query to retrieve the corresponding DBpedia, or Wikipedia URIs.

5.4 Probabilistic Inference with Collective Multilevel Type Classification and Typed Entity Similarity

The probabilistic inference in an alternative approach to define the probability of one set typed entities to be other fine-grained typed by observing data from classification of type classes, and the latent similarity of entities in KG.

Figure 3: (left) A part of DBPedia type hierarchy (right) Entities as subject and object in relational domain.
In this case, the list of fine-grained types is ranked and generated based on the probability score of an entity to select candidate entities, in which the entity typed classes, and entity similarity to the fine-grained type classes are taken into account. Actor typed is also typed of Artist, Person, and Thing classes (see in Fig. 3). The underlying idea of probabilistic ranking is that collective multilabel classification of entities where entities are allowed to have more than one class types and related entities often share identical classes in RDF KG. For instance, an entity of fine-grained type inference further can be cast as a multilabel classification problem, by introducing classes as entities to the data, introducing a relation isClassOf and inferring the probability of relationships isClassOf(i-th entity; j-th class).

6 Approach

We propose the domain-relevance weighted tensor (DrWT) method for fine-grained type inference, that explicitly model a mapping from the domain-based representation to the observed tensor in factorization. Given the KG and additional data along with list of labelled domain, DrTW classifies the entries correspondence to the domains in KG and additional dataset.

DrWT Algorithm:

Input: Knowledge Graph $G = (D, E, R)$; where set $D$ is a set of domains in KG, $D = \{d_1, ..., d_{|D|}\}$; set $E$ is a set of entities (objects), $E = \{e_1, ..., e_{|E|}\}$, and $E_s$ as subject set of entities which occur as subject in relation links, where $E_s \in E$; and $E_o$ as object set of entities which occur as object in relation links where $\{E_o \in E\}$ and, the set $R$ is a set of relations (Predicates) between entities, $R = \{r_1, ..., r_{|R|}\}$.

Output: set $\nabla T_d$ is a list of triples $<<(E_s)_i, (R)_i, (E_o)_i>>$ for each $d_i \in D$ (Domains)

Start
01: function DomainRelevance ($D$, $E$, $R$)
02:    $d \leftarrow$ domain
03:    for each $r_i \in R$ do
04:        for each $r_i \in d$ do
05:            source ($e_s$) : $R \rightarrow E_s$ \hspace{1cm} $\triangleright$ return source($e$)
06:            target ($e_o$) : $R \rightarrow E_o$ \hspace{1cm} $\triangleright$ return target($e$)
07:            $e_d \leftarrow (e_s \cup e_o)$
08:            while $e_i \in e_d$ or $r_i \in d$ do
09:                $T_d \leftarrow \text{SELECT-TRIPLES} \{(e_s)_i, (r)_i, (e_o)_i\}$
10:            end while
11:        $\nabla T_d \leftarrow T_d$
12:    end for
13: end function
This DrWT method makes model to obtain the groups of diverging components into group relevance to the domains in tensor for useful learning of latent features from the relationships in different domains among KG and text data in factorization, in order to improve fine-grained type inference performance.

6.1 Probabilistic Inference

As RDF KG is multi-relational data, the similarity of entities is therefore determined by the similarity of their relationships, following the intuition that "if two objects are in the same relation to the same object, this is evidence that they may be the same object". The collaborative activities of entities as subjects $A_s \in R^{S \times P}$ and objects $A_o \in R^{O \times P}$ in relations in a domain can be modelled by the entity matrix $\hat{A}$, where $\hat{A}$ is QR matrix factorization of $\sum (A_s + A_o)$. For each domain the the latent space $\hat{A}$ therefore reflects the similarity of entities in the relational domain. The type or fine-grained type classes set $C_e = \{t_1, t_2, t_3, ..., t_n\}$ where $C_e$ is a set of Types in one KG. A list of type or fine-grained type classes that are considered for given fine-grained type. For each fine-grained type in $C_e$ the candidate entities set, $E_t = \{\hat{e}_1, \hat{e}_2, \hat{e}_3, ..., \hat{e}_n\}$ where $E_t$ is a set of typed entities in one KG.

We use the Bayes’ theorem [30] for predicting the class candidate entity $E_t$ that have the highest posterior probability given $C_e$, $p(C_e|E_t)$. The posterior probability is utilized to calculate the preference probability of an entity $e$ to be fine-grained typed $t$ in $C_e$ type classes by observing current type classes of entity $e$, and latent similarity of entity $e$ to fine-grained typed entity. The conditional probability can be formulated as:

$$p(C_e|E_t) = \frac{p(E_t|C_e)p(C_e)}{p(E_t)} \quad (1)$$

$$p_{e_n,t_e} = p(t_e|C_e) = \frac{|C_e|}{|E_t|} \sum_{t=1}^{|C_e|} P(e_n|C_e = t) \prod_{\hat{e}_e} P(\hat{E}_e|C_e) \quad (2)$$

where, $P(\hat{E}_e|C_e)$ is probability of likelihood for $t_e$ in $C_e$, is derived from the entities set, $E_t = \{\hat{e}_1, \hat{e}_2, \hat{e}_3, ..., \hat{e}_n\}$ where values from reconstructed tensor $\hat{X}$, and entity similarity values from $\hat{A}$ are used. Further accuracy improvements are therefore achieved by extending the inference model by probabilistic inference with collective multilevel type classification and typed entity similarity.

7 Evaluation plan

We evaluate our approach on Freebase FB15K dataset [2]; FB15K-237 Knowledge Base Completion Dataset [1] and DBpedia 2016-10 release dataset (see Table 01). The FB15K (Bordes et al. 2013), is a subset of Freebase which has been commonly used to evaluate various KG completion models [4] [28] [12] [9] [11] [7] [29]. In the FB15K-237 Knowledge Base Completion Dataset, the
triples (entity-textual-entity) are derived from 200 million sentences from the ClueWeb12 corpus coupled with Freebase entity. There are around 3.9 million text descriptions corresponding to the relation types in Freebase. Two standard

tasks, **Entity Prediction** and **Type Triple Classification** as are proposed to evaluate our approach for fine-grained type inference.

**Evaluation protocol of entity prediction task.** We follow the same protocol in TransE [4]: For each testing triplet (h; r; t), we replace the tail t by every entity e (14951 in FB15K) in the KG and calculate a similarity score obtained by the model on the corrupted triple (h; r; e). Ranking the scores in ascending order, we then get the rank of the original correct triplet. Aggregated over all the testing triplets, the proportion of ranks not larger than 10 (denoted as Hits@10) metric is reported for evaluation. For this metric a higher Hits@10 is better. The results are reported in Table 4.

**Evaluation protocol of type triple classification task.** We follow the same protocol introduced by Socher et al in NTN [26]. Corrupting observed triples for all relations is not required experiment for evaluation to our approach, as it is out of the scope of this work. The decision rule for classification is simple: for each type triple (h; r; t), if the similarity score (obtained by the models) is below a relation-specific threshold $T$, then the triple will be classified as positive. Otherwise, it will be classified as negative. The relation-specific threshold $T$ is determined according to each fine-grained typed entities results. The results are reported in Table 3.

### Table 1. Datasets used in the experiments.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Entities</th>
<th>Relations as object properties</th>
<th>Relations as datatype properties</th>
<th>Relations as specialized datatype properties</th>
<th>Entity class types</th>
<th>YAGO class types</th>
<th>RDF triples from DBpedia 2016-10 release</th>
<th>RDF triples from online DBpedia by SPARQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBpedia</td>
<td>6.14 million</td>
<td>1,105</td>
<td>1,622</td>
<td>132</td>
<td>760</td>
<td>570,276</td>
<td>494 million</td>
<td>1.2 million</td>
</tr>
<tr>
<td>Freebase</td>
<td>14,951</td>
<td>2,766,477</td>
<td>3,977,677</td>
<td></td>
<td></td>
<td></td>
<td>944 million</td>
<td></td>
</tr>
</tbody>
</table>

8 Preliminary results

**Type triple classification Task.** We evaluate our approach by type triple classification task on FB15K with six general types and eight fine-grained types in three different domains: 'Film', 'TV' and 'Music' to compare the strengths and weaknesses of proposed tensor model and the relevant state-of-the-art method RESCAL [17] [18]. As we have mentioned previous section, RESCAL has been
achieved good performance among the state-of-the-art approaches for type inference in KG [18]. As SDType [23] and SNCN [14] are not latent factor model, we can not directly compare our model with them.

Table 2. Types Inference Comparison with RESCAL the state-of-the-art approach

<table>
<thead>
<tr>
<th></th>
<th>RESCAL (Top Class)</th>
<th>OUR APPROACH</th>
<th>RESCAL (Fine-grained)</th>
<th>OUR APPROACH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td>98.0</td>
<td>99.0</td>
<td>TV Director</td>
<td>82.0</td>
</tr>
<tr>
<td>Director</td>
<td>80.0</td>
<td>98.0</td>
<td>TV Director</td>
<td>80.0</td>
</tr>
<tr>
<td>Producer</td>
<td>89.0</td>
<td>93.0</td>
<td>TV Writer</td>
<td>73.0</td>
</tr>
<tr>
<td>Musician</td>
<td>78.0</td>
<td>89.0</td>
<td>TV Producer</td>
<td>81.0</td>
</tr>
<tr>
<td>Singer</td>
<td>74.0</td>
<td>51.0</td>
<td>Film Actor</td>
<td>57.0</td>
</tr>
<tr>
<td>Writer</td>
<td>70.0</td>
<td>76.0</td>
<td>TV Actor</td>
<td>73.0</td>
</tr>
<tr>
<td>Author</td>
<td>89.0</td>
<td>92.0</td>
<td>Singer-Actor</td>
<td>97.0</td>
</tr>
<tr>
<td>Movie</td>
<td>96.0</td>
<td>98.0</td>
<td>Guest TV Actor</td>
<td>97.0</td>
</tr>
</tbody>
</table>

Entity Prediction Task. Entity prediction aims to predict the missing head (h) or tail (t) entity given relation type (r) and other entity, i.e. predicting h given (?, r, t) or predicting t given (h, r, ?) where ? denotes the missing element. Since the dataset is same, we directly report the experimental results of several baselines from the literature, (see table no. 4). Instead of taking overall link prediction task results, we take link prediction reported results for predicting heads and prediction tails based on all types of relations: one-to-one (1-1), one-to-many (1-N), many-to-one (N-1) and many-to-many (N-N) separately.

9 Reflections

This paper proposes a new approach based on a tensor model for fine-grained type inference in KG. This approach is explicitly modelled for leveraging domain knowledge in KG, and for utilizing the additional large amount of interactions among multiple entities and text descriptions. We further develop probabilistic
inference based on collective multilevel type classification and latent similarity of typed entities. Experiment results show that utilizing additional data outside of KG, leveraging domain knowledge with tensor modelling; and developing 'probabilistic inference' by observing data from classification of type classes, and the latent similarity of entities significantly improves performance for inferring fine-grained types of entities in KG. Results in type triple classification and entity prediction tasks show the proposed model outperforms the state-of-the-art approaches for type inference in KG (10% to 19% on different fine-grained types compare to the state-of-the-art approach Rescal), and achieves high results in many-to-one relation in predicting tail for KG completion task.

References


