Knowledge Base Completion With Analogical Inference on Context Graphs

Nada Mimouni and Jean-Claude Moissinac and Anh Tuan Vu

LTCI, Télécom Paris
Institut Polytechnique de Paris
France
Email: nada.mimouni, jean-claude.moissinac, anh.vu @telecom-paris.fr

***Abstract*—Knowledge base completion refers to the task of adding new, missing, links between entities. In this work we are interested in the problem of knowledge Graph (KG) incompleteness in general purpose knowledge bases like DBpedia and Wikidata. We propose an approach for discovering implicit triples using observed ones in the incomplete graph leveraging analogy structures deducted from a KG embedding model. We use a language modelling approach where semantic regularities between words are preserved which we adapt to entities and relations. We consider excerpts from large input graphs as a reduced and meaningful context for a set of entities of a given domain. The first results show that analogical inferences in the projected vector space is relevant to a link prediction task.**

***Keywords*–Knowledge Base; Context graph; Language embedding model; Analogy structure; Link discovery.**

I. Introduction

General purpose knowledge Bases (KB), such as Yago, Wikidata and DBpedia, are valuable background resources for various AI tasks, for example recommendation [1], web search [2] and question answering [3]. However, using these resources bring to light several problems which are mainly due to their substantial size and high incompleteness [4] due to the extremely big amount of real world facts to be encoded. Recently, vector-space embedding models for KB completion have been extensively studied for their efficiency and scalability and proven to achieve state-of-the-art link prediction performance [5], [6], [7], [8]. Numerous KB completion approaches have also been employed which aim at predicting whether or not a relationship not in the KG is likely to be correct. An overview of these models with the results for link prediction and triple classification is given in [9]. KG embedding models learn distributed representations for entities and relations, which are represented as low-dimensional dense vectors, or matrices, in continuous vector spaces. These representations are intended to preserve the information in the KG namely interactions between entities like similarity, relatedness and neighbourhood for different domains.

In this work, we are particularly interested in adapting the language modelling approach proposed by [10] where relational similarities or linguistic regularities between pairs of words are captured. They are represented as translations in the projected vector space where similar words appear close to each other and allow for arithmetic operations on vectors of relations between word pairs. For instance, the vector translation $v(Germany) - v(Berlin) \approx v(France) - v(Paris)$ shows relational similarity between countries and capital cities. It highlights clear-cut the analogical properties between the embedded words expressed by the analogy "$Berlin$ is to $Germany$ as $Paris$ is to $France$". We propose to apply this property to entities and relations in KGs as represented by diagrams (a) and (b) in Figure 1. The vector translation example is likely to capture the $capital$ relationship that we could represent by a translation vector $v(capital)$ verifying the following compositionality [10]: $v(France)+v(capital)-v(Paris) \approx 0$. We use the analogical property for KB completion and show that it is particularly relevant for this task. Our intuition is illustrated by diagrams (b) and (c) in Figure 1 where an unobserved triple can be inferred by mirroring its counterpart in the parallelogram. To the best of our knowledge, leveraging analogy structure of linguistic regularities for KB completion has never been investigated prior to this work. We consider to apply such properties on excerpts from large KGs, we call context graphs, guided by representative entities of a given domain where interactions between entities are more significant. Context graphs show to be bearer of meaning for the considered domain and easier to handle because of their reduced size compared to source graphs.

In the following, Section II gives an overview of related work, Section III describes our approach to build context graphs and learn features for link prediction and Section IV gives the initial results.

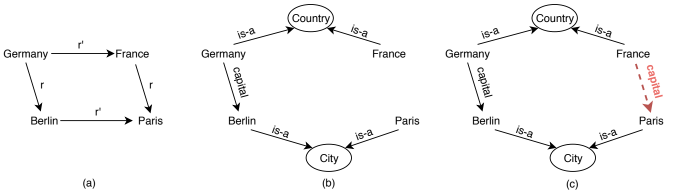

Figure 1. (a) Analogy relation diagram (parallelogram) between countries and capital cities. In KGs (b) and (c), r corresponds to the relation $capital$ and r' is decomposed into two type relations (*is-a*) to concepts $Country$ and $City$.

II. Related work

A closely related approach to our work is described in [11]. The RDF2vec approach uses the neural language model to generate embeddings on entities from walks on two general knowledge bases namely DBpedia and Wikidata. Short random walks are created for the whole set of entities in an image of the KB at a given date. Walks using RDF graph kernels are also used on small test datasets due to scalability limitation. The trained models are made available for reuse. The approach we propose here differs in several aspects. First, we consider undirected labelled edges in the RDF graph to adapt the neural language model, compared to directed graph. Second, we use biased walks guided by the application domain to generate sequences compared to random walks. Third, beside using object properties to build the sequences, we consider DataType properties and literals because we assume that they hold useful information for our application domain (e.g., dates, textual descriptions). Last, we propose to handle scalability issues by contextualizing the input graphs assuming that more relevant information is centralized within a perimeter of α hops around our main entities (α is defined later).

A more general technique called Node2vec is proposed in [12]. It aims to create embeddings for nodes in a (un)directed (a)cyclic (un)weighted graph $G(V, E, W)$ where V is the set of vertices, E the set of edges with weights W. The embeddings are learnt using the Skip-Gram model [10] trained on a corpus of sequences of nodes generated using the sampling strategy. The input graph is turned into a set of directed acyclic sub-graphs with a maximum out degree of 1 using two hyper-parameters for sampling: Return R (probability to go back to the previous node) and Inout Q (probability to explore new parts of the graph).

III. Approach

Here, we define a context graph and show how to build it, then we present how to create a model from our context graph.

A. Building Context Graphs

We define a Context Graph (CG) as a sub-graph of a general KG (e.g. DBpedia) representative of a domain D. The first step to build CG is to identify a list of seeds defining the domain. A seed is an entity from KG corresponding to a concept which is considered relevant for D. For example, if the concept is '*Musée du Louvre*', the corresponding entity in DBpedia is <http://dbpedia.org/resource/Louvre>. In some domains this list is obvious as for museums, hotels or restaurants. In general case, the common practice is to rely on a reference dataset (such as IMDB for cinema).

The second step extracts from KG the neighbourhood for each seed within a given depth filtering useless entities (not informative for D) and returns the final CG as the union of elementary contexts. We use CG in the following as basis for the embedding model.

We create the algorithm `CONTEXT BUILDER` (Algorithm 1) to build a context graph $context$ from a knowledge graph $\mathcal{KG}$ for a given domain D. For a set of seeds ($seedsEntities$), `findNeighbors`(s) extracts a neighbouring context C_v from a knowledge graph $\mathcal{KG}$ for each seed s. The final context, $context$, is updated adding C_v. A list of new seeds, $newSeeds$, is updated with the new collected entities after filtering the terminal nodes with the `EntityFilter` method. The exploration depth $level$ is incremented by 1 at each step up to the desired $radius$ limit. At the end of process, the resulting context $context$ is expanded with the classes of entities extracted from $\mathcal{KG}$ by the methods `AddClasses` and `Entities`.

Algorithm 1: *CONTEXT BUILDER*

```
Function ContextBuilder($KG$, $seedsEntities$, $radius$, $filteredEntities$)
    Input : A knowledge graph $KG$
    A neighbourhood depth to reach $radius$
    A set of entities which are used as seeds $seedsEntities$
    A set of entities which are excluded from the seeds $filteredEntities$
    Output: Context Graph $context$
    $level \leftarrow 0$
    $context \leftarrow \emptyset$
    while $level < radius$ do
        $newSeeds \leftarrow \emptyset$
        foreach $s \in seedsEntities$ do
            $C_v \leftarrow$ FindNeighbors($KG$, $s$)
            $context \leftarrow context \cup C_v$
            $newSeeds \leftarrow newSeeds \cup$
              EntityFilter($C_v$,$filteredEntities$)
        end
        $level \leftarrow level + 1$
        $seedsEntities \leftarrow newSeeds$
    end
    $context \leftarrow context \cup$ AddClasses($KG$,
     Entities($context$))
    return $context$
```

B. Feature Learning

First, we adapt the language modelling approach to KG embedding. We transform the entities and relations in the CG as paths that are considered as sequences of words in natural language. To extract RDF graph sub-structures, we use the breadth-first algorithm to get all the graph walks or random walks for a limited number N. Let $G = (V, E)$ be an RDF graph where V is the set of vertices and E is the set of directed edges. For each vertex v, we generate all or N graph walks P_v of depth d rooted in the vertex v by exploring direct outgoing and incoming edges of v and iteratively direct edges of its neighbours v_i until depth d is reached. The paths after the first iteration follow this pattern $v \rightarrow e_i \rightarrow v_i$ where $e_i \in E$. The final set of sequences for G is the union of the sequences of all the vertices $\bigcup_{v \in V} P_v$.

Next, we train a neural language model which estimates the likelihood of a sequence of entities and relations appearing in the graph and represents them as vectors of latent numerical features. To do this, we use the continuous bag of words (CBOW) and Skip-Gram models as described in [10]. CBOW predicts target words w_t from context words within a context window c while Skip-Gram does the inverse and attempts to predict the context words from the target word. The probability $p(w_t | w_{t-c} ... w_{t+c})$ is calculated using the $softmax$ function.

Finally, we extract analogical properties from the feature space to estimate the existence of new relationships between entities. We use the following arithmetic operation on the feature vectors (entities of Figure 1): $v(Berlin) - v(Germany) + v(France) = v(x)$ which we consider is solved correctly if $v(x)$ is most similar to $v(Paris)$. On the left-hand side of the equation, entities contribute positively or negatively to the similarity according to the corresponding sign. For example, $Germany$ and $France$ having the same type $Country$ contribute with different signs, $Berlin$, of a different $City$ type, contribute with the opposite sign of the corresponding Country. The right-hand side of the equation contains the missing corner of the diagram which remains to be predicted. We then use cosine similarity measures between the resulting vector $v(x)$ and vectors of all other entities of the same type in the embedding space (discarding the original ones of the equation) in order to rank the results.

IV. Experimental Evaluation

A. Case Study

We test our approach on a sub-graph of DBpedia representing a target domain: here we chose museums of Paris. We propose to address the scalability issue by contextualizing the input graphs assuming that more relevant information is centralized within a perimeter of α hops around main entities of this domain (we used $\alpha = 2$ as suggested by [13]). We build our KG as the union of individual contextual graphs of all entities representing the input data from the cultural institution *Paris Musées* (12 sites). We identify each site by its URI on DBpedia-fr after an entity resolution task (in the following, we denote the URI http://fr.dbpedia.org/resource/entity shortly as dbr:entity). The final graph contains 448309 entities, 2285 relations and 5122879 triples. To generate sequences of entities and relations we use random graph walks with $N = 1000$ for depth $d = \{4, 8\}$. We also consider for each entity all walks of depth $d = 2$ (direct neighbours).

We then train the Skip-Gram word2vec model on the corpus of sequences with the following parameters: window size = 5, number of iterations = 10, negative samples = 25 (for the purpose of optimisation) and dimension of the entities' vectors = 200. We use gensim implementation of word2vec [14]. We also trained our model with CBOW method and with larger vector dimension (500). We notice in general better performance with Skip-Gram method, but cannot do any assertion about vector dimension. Our method can't be evaluated against other ones using standard datasets such as FB15K, WN18 [6], [7]. It requires to define a context and extracts a subgraph from it, none of the other methods uses such a context in the available evaluations.

B. Evaluation Protocol

Existing benchmarks for testing analogy task in the literature are designed for words from text corpora. To the best of our knowledge, using language model driven analogy for link prediction in knowledge graphs has not been investigated yet. To evaluate our approach, we build a ground-truth for analogy between entities in the KG. Each entry corresponds to a parallelogram as described in Figure 1 with one unobserved triple in the KG. For each entity, corresponding to a museum site in our application, we collect a list of well-known artists for this site as follows: find in DBpedia-fr the list of artists (dbo:Artist) or otherwise, individuals (dbo: Person) who are associated with the site. For some sites, we manually create the list, for example by searching for well-known artists for a museum on the website [15].

The evaluation test aims at discovering $artist_a$ for $museum_a$ considering a known triple <$museum_b$, $artist_b$> while varying b and measuring the mean of the returned results. We use conventional metrics: Mean Reciprocal Rank (MRR) and the number of correct responses at a fixed rate (Hits@).

The evaluation protocol is as follows: for each $Muri_i$, URI of a museum, let $Auri_i$ be the URI of the first artist identified for $Muri_i$, consider all $Muri_j \mid j \neq i$, find the top most similar entities of the predicted vectors with positives = [$Auri_i$,$Muri_j$] and negative =[$Muri_i$]. In the list of results, we filter by type $Artist$, we then examine the intersection with artists $Auri_l$ associated with $Muri_j$. It is worth noticing that we frequently find loosely defined links between museums and artists; such links are very common in DBpedia and use the property `wikiPageWikiLink` representing an untyped link. Subsequent work is required to qualify them.

C. Results

Table I shows results of MRR and Hits@$\{3, 5, 10\}$ (%) for $d = \{4, 8\}$ and $N = 1000$. The final row of the table with columns $d = 8$ shows the impact of considering longer paths on the performances of the approach. In fact, longer paths capture richer contexts for entities and results in better vectors estimation by the neural language model.

We compared our approach with the one presented in [11] which creates a model, modelDB, for all entities in DBpedia. For each entity in our ground-truth built on DBpedia-fr, we look for its equivalent in DBpedia and verify that it is contained in the vocabulary of modelDB built with $d = 4$. Only 7 out of 12 museum entities are in modelDB, as well as their first associated artist among others. The analogy tests return globally poor results. ModelDB were unable to retrieve relevant entities in top 100 returned answers, as for our model trained on the CG, without any improvement even if extended to top 5000. This is not a surprising result if we look at the following table which shows that our CG has a better coverage of the ground-truth domain entities, mainly artists, compared to DBpedia.

TABLE II. Ground-truth entities in DBpedia and DDBpedia-fr.

	dbo:Person	dbo:Artist	No type	dbo:Museum
DBpedia	272	190	44	7
DBpedia-fr	272	327	6	12

The first row of Table II shows that not all dbo:Artist are linked to dbo:Person (ex: dbr:Sonia_Delaunay). With 12 museums and 334 artists in the reference list, 97.90% can be identified as an artist in our context graph vs. 56.88% in DBPedia, which partly explains the poor results. As we filter the returned results by type Artist (or more generally by Person), several relevant answers are filtered.

We also compared our approach with a random selection of entities of type dbo:Artist in the vocabulary of the model. The results, given in columns $d = 4R$ of table I, show a great benefit of leveraging the regularities in the vocabulary space to extract relationships between entities.

While analysing values on Table I, we noticed wide discrepancies between results of different museums. For example, Hits@10 values for dbr:Musée_d'art_moderne and dbr:Musée_de_Grenoble are respectively: 0.83 and 0.33. This impacts the global performance of all museums (last row of table I). The result means for the second value that the system was not able to retrieve the corresponding artist for dbr:Musée_de_Grenoble in top returned results. We argue this

TABLE I. MRR and Hits@$\{3, 5, 10\}$ (%) of a subset of representative examples of *Paris Musées* data for $d = \{4, 8\}$ and $N = 1000$ with analogy and random for $d = 4$ (d=4R).

	MRR			Hits@3			Hits@5			Hits@10		
Entity	d=4R	d=4	d=8	d=4R	d=4	d=8	d=4R	d=4	d=8	d=4R	d=4	d=8
dbr:Musée_Bourdelle	0,05	0,39	0,43	0,09	0,50	0,42	0,18	0,50	0,42	0,18	0,66	0,50
dbr:Musée_Carnavalet	0,01	0,43	0,59	0,00	0,58	0,67	0,09	0,66	0,75	0,09	0,83	0,75
dbr:Musée_Zadkine	0,00	0,43	0,44	0,00	0,41	0,42	0,00	0,50	0,50	0,00	0,50	0,50
dbr:Musée_Cernuschi	0,01	0,42	0,50	0,00	0,50	0,58	0,00	0,58	0,67	0,09	0,75	0,67
dbr:Petit_Palais	0,04	0,38	0,63	0,09	0,50	0,75	0,09	0,66	0,75	0,09	0,66	0,75
dbr:Maison_de_Balzac	0,03	0,23	0,44	0,09	0,25	0,58	0,09	0,41	0,58	0,09	0,41	0,58
dbr:Musée_Cognacq-Jay	0,09	0,33	0,49	0,09	0,33	0,58	0,09	0,33	0,58	0,09	0,33	0,58
dbr:Musée_d'art_moderne	0,04	0,36	0,71	0,09	0,41	0,75	0,09	0,50	0,83	0,09	0,58	0,83
dbr:Musée_Romantique	0,03	0,34	0,48	0,09	0,41	0,50	0,09	0,41	0,58	0,09	0,50	0,58
dbr:Palais_Galliera	0,00	0,36	0,48	0,00	0,50	0,50	0,00	0,50	0,58	0,00	0,50	0,58
dbr:Maison_de_Victor_Hugo	0,01	0,38	0,55	0,00	0,50	0,58	0,00	0,58	0,58	0,18	0,58	0,67
dbr:Musée_de_Grenoble	0,00	0,34	0,33	0,00	0,41	0,33	0,00	0,50	0,33	0,00	0,50	0,33
All entities in *Paris Musées*	0,02	**0,37**	**0,52**	0,04	**0,44**	**0,58**	0,06	**0,51**	**0,62**	0,09	**0,57**	**0,64**

is mostly related to the representativeness of this artist's entity in the KG and how it is linked to the museum's entity; less interlinked entities (directly or indirectly through neighbours) have less chance to be related with the analogy structure in the embedding space. To explain this, we run another evaluation as follows: for each $Auri_i$, URI of an artist, consider all known triples <$Muri_j$, $Auri_j$ > | $j \neq i$, find the top most similar entities of the predicted vectors ranked by similarity. In the list of results, we filter by type $Museum$, we then examine the intersection with museums $Muri_l$ associated with $Auri_i$.

Table III shows results of MRR and Hits@$\{3, 5, 10\}$ (%) for $d = 4$ and $N = 1000$. The wide differences between artists' results in the last column of the table (Hits@10) (ex. dbr:Victor_Hugo and dbr:Geer_Van_Velde) reveals the impact of the triple interlinkage in the graph on the analogy prediction test. Thus, good prediction performance of new triples could be achieved with a good representativeness of known triples by the context graph.

TABLE III. MRR and Hits@$\{3, 5, 10\}$ (%) of representative examples of artists exhibited in museums of *Paris Musées* for $d = 4$ and $N = 1000$

Entity	MRR	Hits@3	Hits@5	Hits@10
dbr:Antoine_Bourdelle	0,61	0,72	0,81	0,81
dbr:Israël_Silvestre	0,08	0,09	0,13	0,13
dbr:Gustave_Courbet	0,38	0,45	0,45	0,72
dbr:Ossip_Zadkine	0,67	0,81	0,90	0,91
dbr:Xu_Beihong	0,74	1,0	1,0	1,0
dbr:Honoré_de_Balzac	0,53	0,72	0,81	0,81
dbr:François_Boucher	0,65	0,72	0,81	0,81
dbr:Geer_Van_Velde	0,09	0,09	0,09	0,09
dbr:Ary_Scheffer	0,62	0,72	0,81	0,91
dbr:Jacques_Heim	0,18	0,18	0,45	0,72
dbr:Victor_Hugo	0,53	0,63	0,72	0,91

V. Conclusion

In this paper, we presented an approach for link discovery in KBs based on the neural language embedding of contextualized RDF graphs and leveraging analogical structures extracted from relational similarities which could be used to infer new unobserved triples from the observed ones. The test of our approach on a domain-specific ground-truth shows promising results. We will continue to expand upon the research and compare it with state-of-the-art approaches for KB completion on the standard baselines.

Acknowledgment

This work is conducted as part of the Data & Musée project selected in the $23rd$ call for projects of the Fonds Unique Interministériel (FUI) and certified by Cap Digital and Imaginove. http://datamusee.fr/le-projet/. This work is an extension of a poster paper scheduled for publication in the ESWC 2019 supplementary proceedings.

References

[1] M. Al-Ghossein, T. Abdessalem, and A. Barré, "Open data in the hotel industry: leveraging forthcoming events for hotel recommendation," J. of IT & Tourism, vol. 20, no. 1-4, 2018, pp. 191–216.

[2] S. Szumlanski and F. Gomez, "Automatically acquiring a semantic network of related concepts," in Proceedings of the 19th ACM CIKM, 2010, pp. 19–28.

[3] J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, and X. Li, "Neural generative question answering," in Proceedings of IJCAI. AAAI Press, 2016, pp. 2972–2978.

[4] B. Min, R. Grishman, L. Wan, C. Wang, and D. Gondek, "Distant supervision for relation extraction with an incomplete knowledge base," in Proceedings of NAACL-HLT, 2013, pp. 777–782.

[5] M. Nickel, L. Rosasco, and T. A. Poggio, "Holographic embeddings of knowledge graphs," in AAAI, 2016.

[6] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in NIPS, 2013.

[7] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings," in Proceedings of ICML, 2017, pp. 2168–2178.

[8] A. García-Durán, A. Bordes, N. Usunier, and Y. Grandvalet, "Combining two and three-way embedding models for link prediction in knowledge bases," J. Artif. Intell. Res., vol. 55, 2016, pp. 715–742.

[9] D. Q. Nguyen, "An overview of embedding models of entities and relationships for knowledge base completion," CoRR, vol. abs/1703.08098, 2017.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in NIPS, 2013.

[11] P. Ristoski and H. Paulheim, "Rdf2vec: Rdf graph embeddings for data mining," in Proceedings of ISWC, 2016, p. 498 – 514.

[12] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in ACM SIGKDD, 2016.

[13] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene, "Unsupervised graph-based topic labelling using dbpedia," in Proceedings of WSDM, 2013, pp. 465–474.

[14] "Gensim implementation of word2vec," https://radimrehurek.com/gensim/models/word2vec.html, accessed: 2019-08.

[15] "Paris musée collection website." http://parismuseescollections.paris.fr/fr/recherche, accessed: 2019-03.