Enriching Wikidata with Semantified Wikipedia Hyperlinks

Armand Boschin, Thomas Bonald

Enriching Wikidata with Semantified Wikipedia Hyperlinks

Armand Boschin and Thomas Bonald

Télécom Paris, Institut Polytechnique de Paris
{armand.boschin,thomas.bonald}@telecom-paris.fr

Abstract. We propose a novel approach to enrich Wikidata with the textual content of Wikipedia. Specifically, we leverage knowledge graph (KG) embedding models to classify the hyperlinks between Wikipedia articles and predict the corresponding facts. For instance, we would like to complete the triple (*Berlin*, *, *Germany*) with the relation *capital of*, given a hyperlink from *Berlin* to *Germany* in Wikipedia. While existing KG embedding models can be used for this task of relation prediction, they were not explicitly designed for it and their performance is not satisfactory. In this paper, we propose two methods that greatly improve the performance of these models on this task: first, a new *negative sampling* method that balances the roles of entities and relations during training; second, a method to exploit the *types* of entities in the selection of candidate relations. We obtain accuracy scores as high as 94% on the popular FB15k237 dataset and 75% on WDV5, an extraction of Wikidata. The efficiency of the approach is illustrated on some Wikipedia pages, where new facts unknown to Wikidata are predicted by our method.

Keywords: Wikidata · knowledge graph · embedding · relation prediction · negative sampling · relation typing · machine learning

1 Introduction

In the recent years, Wikipedia has become the largest open-source collection of knowledge. Its textual content is however mostly unstructured, the structured information being mainly limited to the content of infoboxes (e.g., place and date of birth for articles on humans). The hyperlinks make another structure which is not fully integrated in Wikidata yet. The main challenge is that, in order to know the meaning of an hyperlink, an agent needs to read the text in which the hyperlink is embedded. While some hyperlinks do not correspond to relevant facts, we claim that this is a rich source of information to complete Wikidata. To illustrate this, one can look at the level 5 of Wikipedia vital articles[1], that is about 40,000 pages *serving as a centralized watchlist to track the quality of the most important articles*. These Wikipedia pages are linked by slightly more than 3 million hyperlinks, which is far more than the approximately 200,000

[1] https://en.wikipedia.org/wiki/Wikipedia:Vital_articles/Level/5

facts linking the corresponding entities in Wikidata. For instance, there is a link from the page *Henri Poincaré* to the page *Optics* in Wikipedia. This suggests the existence of a relation linking the two entities, here *field of work*. This fact is *not* present in Wikidata.

Formally, a KG consists of a set of vertices called *entities* (e.g., person, place, date, concept) linked by directed edges in the form of triples (h, r, t) where h (resp. t) is the head (resp. the tail) entity and r is a *relation* carrying the semantic nature of the edge. When a triple is known to be true, it is called a fact.

In this paper, we address the issue of *relation* prediction: finding the relation linking some given head and tail entities. For instance, we would like to complete the triple (*Berlin*, *, *Germany*) with the relation *capital of*, assuming the fact is not in the KG. This task is also known as the semantification of a link. For this, we leverage the embedding of the entities and relations of the KG to compute scores on possible triples. Though most existing works on embeddings have focused on the task of *link* prediction, that is, completing either the triple (*, *capital of*, *Germany*) (head prediction) or (*Berlin*, *capital of*, *) (tail prediction), we show that embeddings for *relation* prediction can also perform notably well. We propose two techniques for that: first, we adapt the training of the models by balancing the role of entities and relations in the negative sampling step and then we use the types of entities to filter candidate relations.

These techniques prove very efficient, allowing a simple embedding model like TransE [3] to reach accuracy of 94% on the popular FB15k237 dataset and 75% on WDV5, an extraction of Wikidata based on the level 5 of Wikipedia vital articles. This suggests that Wikidata can be significantly enriched by the semantification of Wikipedia hyperlinks.

Contributions. The main contributions of this work are the following:

- An approach to enrich Wikidata by the semantification of the hyperlinks of Wikipedia.
- A novel negative sampling technique for improving the ability of KG embedding models to predict relations, without affecting their performance on link prediction.
- A novel filtering technique for relation prediction where candidate relations are selected through the types of the head and tail entities.
- A new dataset, WDV5, consisting of the facts between entities of Wikidata corresponding to the level 5 of Wikipedia vital articles[2].

2 Related work

KG embedding. A KG embedding model is defined as a function f that computes a score for any triple (h, r, t) using some vector representations of h, t and

[2] https://en.wikipedia.org/wiki/Wikipedia:Vital_articles/Level/5

r. By extension, the vectors representing entities and relations are called embeddings. KG embeddings have been specifically designed for link prediction[3]: given an entity h (resp. t) and a relation r, the model is used to predict an entity t (resp. h) so that the fact (h, r, t) is the most likely to be true. This prediction is done by selecting the entity giving the highest score among all entities.

There are three categories of models depending on the form of the scoring function f and thus on the way entities and relations interact in the vector space (see [20] for more details):

- Linear models, where h, r, t are linked by a linear relation in the vector space. Some projections can be added to increase the expressiveness of the model. Examples of such models include TransE [3] and TransH [22].
- Bilinear models, where the relation r is a bilinear form of h and t in the vector space. Examples include RESCAL [10] and ComplEx [18].
- Deep models, based on neural networks, possibly including attention mechanisms [9,21]. These models give state-of-the-art performance in link prediction but are usually heavy and hard to train and prone to over-fitting.

Negative sampling. The scoring function f of an embedding model is expected to discriminate facts from false statements and thus needs to be trained with both. Since most KGs do not record false statements, training is usually done under the Closed World Assumption (CWA), i.e., unknown triples are considered as false. This may seem contradictory as the model is then used to predict unknown facts, that are expected to be true. This is however the only way to learn meaningful scoring functions f. The random generation of false statements is known as Negative Sampling (NS). It has a major impact on the performance of the trained model [7].

Given some known fact (h, r, t), the usual way to create a false statement from it (under the CWA) is to randomly choose either the head entity or the tail entity and to replace it with another random entity of the KG [3]. This technique was improved in [22] by using a Bernoulli parameter (see Section 3). The replacement of the relation is rarely considered. It is mentioned in [23] but not precisely described nor studied, as it is not the main focus of that article. We propose a modification of the Bernoulli NS technique to include random replacement of the relation, to get high performance in both link prediction and relation prediction.

Type Filtering. Most KGs assign one or several type(s) to each entity through a `rdf:type` relation (e.g., the *P31: "instance of"* relation in Wikidata). The types of entities have mainly been used in link prediction, either to enforce type constraints in negative sampling or to select the candidate entities [8,23].

[3] Note that some authors refer to this task as relation prediction, see [9] for instance. We make a clear distinction between link prediction (head or tail entity unknown) and relation prediction (relation unknown).

KBs can also enforce type constraints on relations via `rdfs:domain` and `rdfs:range` constraints. In relation prediction, selecting candidate relations with these constraints seems natural but they can be missing or too coarse grained making the filtering either too restrictive or with no effect. In Wikidata, relation constraints are hints for the editors, not firm restrictions[4]. We propose a method to *infer* such constraints simply from the `rdf:type` relation of the KG at hand and use the resulting constraints in relation prediction. We show that it has a major impact on performance.

NLP for relation prediction There are two main tasks tackled by NLP methods. The first is relation prediction, also known as relation extraction, consisting in predicting the semantic relation linking two entities using sentences describing these entities. The best performing models rely on deep neural networks with attention mechanisms [19,12,25]. The second task is entity linking, that is linking relations of a KG to plain text surface forms. Some interesting articles are [14,24]. Our task of relation prediction in KG is different as it relies on the graph structure of the KG only and on not any textual content. A method combining both approaches is left as an interesting perspective for future research.

Relation linking

Hyperlink semantification. Very few works exist on the semantification of Wikipedia hyperlinks using the graph structure of the KG only. The approach of Galarraga et al. [5] is based on rule mining. A limit of this method is that it can only predict relations for entities matching the body of the mined rule. Our technique based on KG embedding applies to all links.

3 Background

Bernoulli Negative Sampling. The usual negative sampling technique (noted BerNS) relies on relation-specific Bernoulli distributions to choose between the head or the tail which entity of a fact should be replaced to maximize the probability of the resulting triple to be false [22]. Formally, a Bernoulli parameter p^r is computed for each relation r as follows:

$$p^r = \frac{\rho^r_{t,h}}{\rho^r_{t,h} + \rho^r_{h,t}},$$

where $\rho^r_{t,h}$ (resp. $\rho^r_{h,t}$) is the average number of tail entity per head entity (resp. head entity per tail entity) among all known facts involving r. This parameter p^r is the probability to replace the head entity of the fact.

As an example, consider "*author of*" which is a one-to-many relation (one author and many potential books). In that case, the head entity (an author)

[4] https://www.wikidata.org/wiki/Help:Property_constraints_portal

should be more likely replaced than the tail entity (a book), yielding a false statement with greater probability.

Model training. Training a model comes down to finding its parameters (the embeddings) so that the scoring function gives high scores to facts and low scores to false statements. Given a training set of facts denoted (h, r, t), the corresponding false statements (h', r, t') are generated by NS. Then for each pair of facts (h, r, t) and (h', r, t'), a loss measuring the gap between the corresponding scores is computed, $\ell(f(h, r, t), f(h', r, t'))$. This loss ℓ should be high for close scores. Examples include the logistic loss and the margin loss [16]. Minimizing the overall loss (e.g., by gradient descent [13]) gives a scoring function f that is expected to discriminate facts from false statements.

4 Relation prediction

4.1 Approach

Our approach relies on the following techniques.

KG Embedding. KG embedding models can be used for relation prediction the same way they are used in the aforementioned link prediction: given two entities h and t, an embedding model and its scoring function f, the relations of the graph can be ranked by decreasing order of scores: $f(h, r_1, t) > f(h, r_2, t) > \cdots > f(h, r_k, t)$. The relation r_1 is then predicted, corresponding to the fact (h, r_1, t). Note that this method applies to the case of undirected links, by ranking the scores of the predictions for both directed links (h, t) and (t, h). This is especially useful when some relations have no reciprocal.

Balanced Negative Sampling. Simple experiments show that off-the-shelf linear models like TransE perform really badly in relation prediction (3% of Hit@1 on FB15k237, see Table 2a). This suggests that the representation of relations is not as good as that of entities. It turns out that entities and relations play similar roles in the training procedure except for the NS step. Usually, only the entities are randomly replaced to get false statements (see Section 3). We propose a simple modification of BerNS to balance the roles of entities and relations during training. Rather than just replacing one of the two entities of a known fact, we replace the relation with some probability p, and apply BerNS otherwise (See Algorithm 1). This new method is called Balanced Negative Sampling (BalNS). The default value for p is set to $\frac{1}{2}$. Experiments have shown that the value of the parameter has no major impact on the performances of the approach as long as it is greater than 0.1.

Algorithm 1: Balanced Negative Sampling (BalNS).

Input: (h, r, t), a fact
Input: p, probability to replace the relation
Output: (h', r', t'), a false statement
Data: $\mathcal{T}$, the facts in the KG
Data: p^r, Bernoulli parameter for relation r

$(h', r', t') \leftarrow (h, r, t)$
while $(h', r', t') \in \mathcal{T}$ **do**
  $u \leftarrow$ uniform random variable on $[0, 1]$
  if $u < p$ **then**
    $r' \leftarrow$ random relation
  else
    $(h', t') \leftarrow \text{BerNS}(h, r, t)$
return (h', r', t')

Type Filtering for relation prediction. Another key technique to improve the quality of relation prediction is through Type Filtering (TF). An entity e is said to have the type t if the fact (e, `rdf:type`, t) is known. To predict the relation linking h and t, only relations that are known to link entities of the type(s) of h to entities of the type(s) of t should be considered. Formally, we say that a relation r links type a to type b if there exists some known fact (h, r, t) with the head entity h of type a and the tail entity t of type b. Now for predicting the relation missing in $(h, *, t)$, we propose to consider as candidates only the relations r linking any type of h to any type of t. The corresponding algorithm for relation prediction is described in Algorithm 2. Observe that if either the head entity h and/or the tail entity t is not typed, the candidate relations are then the relations that are involved in a training fact with either h as a head entity or t as a tail entity. In the end, if no relation meet any constraint, there is no filtering, i.e., all relations are selected. Regarding speed, this step has no significant impact on the global computation time with proper index: linking entities to their types and types to possible relations.

Algorithm 2: Relation prediction with Type Filtering (TF).

Input: h, t, entities
Input: f, scoring function
Output: r, relation linking h to t
Data: $\mathcal{T}$, the facts in the KG
Data: $\mathcal{R}$, the relations in the KG

$A \leftarrow$ types of h
$B \leftarrow$ types of t
if $|A| > 0$ *and* $|B| > 0$ **then**
  $R \leftarrow \{r : \forall (a, b) \in A \times B, \exists h', t' : \text{type}(h') = a, \text{type}(t') = b, (h', r, t') \in \mathcal{T}\}$
else
  $R \leftarrow \{r \in \mathcal{R} : \exists e : (h, r, e) \in \mathcal{T}\} \cup \{r \in \mathcal{R} : \exists e : (e, r, t) \in \mathcal{T}\}$
if $|R| = 0$ **then**
  $R \leftarrow \mathcal{R}$
$r \leftarrow \arg\max(\{f(h, r, t), r \in R\})$
return r

To summarize, our approach relies on the following steps:

1. Training the model (e.g., TransE or ComplEx) with BalNS (Algorithm 1).
2. Predicting relations with TF (Algorithm 2).

4.2 Evaluation

Given an embedding model trained with BalNS and some known fact (h, r, t) of a test set, all relations selected by TF are ranked by decreasing score. The rank of the true relation r is recorded as the recovery rank (if the true relation r is not selected by TF, the rank is set to the maximum). Usual metrics of link prediction like Mean Reciprocal Rank (MRR: average of the inverses of the recovery ranks) and Hit@k (proportion of tests in which the recovery rank is at most k) can then be reported. In the filtered setting, any relation that is ranked better than r and that is known to lead to a fact (i.e., in the training set) is discarded, so that the model is not penalized for predicting known facts that are simply more likely than the target one.

5 Experiments

The experiments aim at assessing the performance of our approach on existing KGs and at showing its practical interest on a real-world task, i.e., the semantification of Wikipedia hyperlinks. All experiments can be reproduced using the publicly available code[5] and data [6].

[5] https://gitlab.telecom-paris.fr/aboschin/hyperlinks-semantification
[6] https://netset.telecom-paris.fr/pages/wikivitals+.html

Datasets The datasets used in the experiments are shown in Table 1.

Dataset	Entities/nodes	Facts/edges	Relations	Types	Typed entities
FB15k237	14,541	310,116	237	73	2,719
WDV5	39,062	231,744	607	1,206	22,883
Wikivitals+	39,062	3,008,116			

Table 1: Key features of the datasets used in experiments.

One of the most common datasets used to evaluate the quality of KG embeddings is a subset of Freebase called FB15k237 [17]. The typing relation from Freebase is however not included in it and resources are no longer available online since the discontinuation of the Freebase project [2]. Types were then imported from Wikidata using a matching between the two KBs. Attention was paid to prevent data leakage by removing any imported fact that could match an existing validation or test fact. For comparability reasons, the new facts were not used to train the embedding models, only for the TF step. Only 18,6% of entities are typed, see Table 1.

We also introduce WDV5, a new dataset containing the facts linking entities of Wikidata corresponding to the level 5 of Wikipedia vital articles (see Section 1). To type entities, only typing facts included in the dataset are used (i.e., all types are entities of WDV5). In particular, not all entities are typed (only 56%, see Table 1). It is important to note that WDV5 is a raw extract from Wikidata, without any pre-processing. As such, we expect the corresponding experiments to be more representative of real use cases than those based FB15k237.

For the semantification of Wikipedia hyperlinks, we use Wikivitals+, an extraction of the level 5 of Wikipedia vital articles and the hyperlinks between them. We only keep the pages that have a corresponding Wikidata entity. This dataset provides many hyperlinks that are natural candidates for true facts, after relation prediction.

Baseline In order to measure the impact of using a KG embedding model for ranking the candidates selected by TF, we compare our approach to a simple baseline that ranks the candidate relations by popularity in the training set, in number of facts.

Process Two off-the-shelf embedding models were chosen for the experiments:

- TransE [3], the simplest linear model, intuitive and fast to train and apply.
- ComplEx [18], the best bilinear model, with twice more parameters, longer to train and apply.

The models were trained using the Adam algorithm for optimization [6], dropout for regularization [15] and early-stopping with 100 epochs of patience (on the filtered validation MRR for link prediction). All experiments were done

using `Python` 3.8, `PyTorch` 1.7.0 [11], `TorchKGE` 0.16.25 [4] `pytorch-ignite` 0.4.4 and a Nvidia Titan V GPU powered with `Cuda` 10.1. The hyper-parameters of the embedding models were tuned using `hyperopt` 0.2.5. The possible values along with those chosen are listed in the provided supplemental material.

In the case of FB15k237, the split between train, validation and test sets is set by Toutanova et al. [17]. For WDV5, we split the dataset at random with 80% of the facts for training, 10% for validation (for choosing hyper-parameters) and 10% for testing. The reported metrics are averaged over 6 distinct random splits and independent training procedures.

6 Results

6.1 Performance

The results for relation prediction are shown in Table 2 with metrics computed in a filtered setting, for different variants of the model so as to assess the respective gains of the proposed techniques:

- Original: The base model (either TransE or ComplEx) trained with BerNS.
- BalNS: The base model trained with BalNS.
- TF: The base model evaluated with Type Filtering (TF).
- BalNS & TF: The base model trained with BalNS and evaluated with TF.

The original version of TransE is not efficient on FB15k237 (only 3% of Hit@1). ComplEx performs however notably well on the same dataset (89% of Hit@1). It seems less sensitive to the unbalanced role of entities and relations during training. We suspect however that the score of ComplEx on FB15k237 mainly results from over-fitting due to lack of new datasets in the KG embedding literature over the past few years and over-engineering of FB15k237 (it is the second version of the subset). This has already been argued in [1] and it is confirmed by the fact that TransE and ComplEx have almost the same scores (around 45% of Hit@1) on the new dataset WDV5 which is a raw extraction from Wikidata.

Balanced Negative Sampling. Training with BalNS has a strong impact on the relation prediction performance of the models: training TransE on FB15k237 with BalNS rather than BerNS increases the Hit@1 from 3% to 91%. This confirms the intuition that the relation embeddings were not well trained. The difference is less impressive for ComplEx on FB15k237 but the original ComplEx model performs already quite well on this dataset. On WDV5, there is a big increase in Hit@1 for both models: 24% for TransE and 28% for ComplEx.

Type filtering. TF has a strong impact on the performance of the models. Looking at Hit@1 on WDV5, TransE goes from 45% to 58% and ComplEx goes from 45% to 76%. Note that Type Filtering alone (the baseline) performs almost

Base model	Variant	MRR	Hit@1	Hit@5
TransE	Original	0.061	0.033	0.049
	BalNS	0.940	0.914	0.972
	TF	0.405	0.184	0.744
	BalNS & TF	0.957	0.935	**0.983**
ComplEx	Original	0.928	0.894	0.967
	BalNS	0.956	0.934	0.982
	TF	0.953	0.927	**0.983**
	BalNS & TF	**0.961**	**0.943**	**0.983**
Baseline		0.153	0.050	0.262

(a) FB15k237

Base model	Variant	MRR	Hit@1	Hit@5
TransE	Original	0.556 ± 0.116	0.458 ± 0.128	0.664 ± 0.102
	BalNS	0.779 ± 0.006	0.697 ± 0.017	0.881 ± 0.012
	TF	0.711 ± 0.063	0.588 ± 0.092	0.872 ± 0.020
	BalNS & TF	0.821 ± 0.002	0.754 ± 0.003	0.903 ± 0.002
ComplEx	Original	0.546 ± 0.078	0.454 ± 0.108	0.649 ± 0.052
	BalNS	0.816 ± 0.037	0.734 ± 0.059	**0.917 ± 0.011**
	TF	0.826 ± 0.009	**0.765 ± 0.013**	0.902 ± 0.004
	BalNS & TF	**0.827 ± 0.024**	0.762 ± 0.041	0.910 ± 0.003
Baseline		0.516 ± 0.006	0.416 ± 0.006	0.618 ± 0.006

(b) WDV5 (mean ± standard deviation).

Table 2: Results of relation prediction on FB15k237 and WDV5.

as well as the original embedding models. It is however largely beaten by the combination of TF with scoring by an embedding model. The gain of using and embedding model is very important.

Complete model. The combination of BalNS and TF gives the best results. On FB15k237, the increase in performance of TransE is impressive (Hit@1 from 3% to 94%) and makes this model almost as efficient as ComplEx. This is obtained through additional facts imported from Wikidata for TF but the scores of TransE simply trained with BalNS (and without TF) are already close to those of ComplEx.

On WDV5, all performance metrics are significantly improved by our approach. Both models that perform similarly in their original form remain close. On average, ComplEx beats TransE by 1% in Hit@1 but the scores of TransE are much more stable from one split to the other, as shown by the lower standard deviation. The intervals of fluctuation of MRR and Hit@1 tend to be reduced

if the model is trained with BalNS. This is particularly true for TransE, whose standard deviation for each metric is very small.

It is remarkable to get almost identical performance with TransE and ComplEx, knowing that TransE has half the number of parameters of ComplEx, is more geometrically intuitive and requires 6 times less operations for each gradient descent step during training.

6.2 Application to Wikipedia Hyperlinks

In order to predict the relation associated to a hyperlink, we use the TransE embedding of WDV5 trained with BalNS and applied using TF. When two pages are linked and the corresponding Wikidata entities are involved in a fact of Wikidata (110,311 out of 3,008,116 hyperlinks), we can compare the predicted relation to the ground-truth. We obtain 84% of accuracy. This good score is expected as the model is trained on WDV5 facts and some hyperlinks indeed correspond to existing facts. However, it is interesting to look at cases where the prediction is different from the true fact. We have observed that the model can hardly predict directed relations (e.g., parent-child) or semantically close relations (e.g., *employer* and *educated at* for links between scholars and universities). This is not surprising as the only available data is the structure of the KG. Some other mistakes come from the embedding model itself, for example *headquarter location* always has a lower score than *twinned administrative body* for some reason, making the headquarter predictions all wrong.

In Table 3, we report for two pages the semantified hyperlinks that got the highest scores. It is reassuring to see that most of the resulting facts are true, many of them being however already known in Wikidata. A few mistakes could be avoided using a little bit of context (i.e., text information) but these results suggest that our approach is able to correctly semantify many links.

It seems however difficult to produce automatically a full dataset in this way. First, the scores of embedding models are usually not normalized so comparing them works fine when done *locally* (e.g., looking at the links of a particular page) but comparing the scores of the three million possible facts is not feasible. Second, many facts that get a high score are very likely but require additional information not present in data. For example the three most likely facts resulting from semantified hyperlinks of Wikivitals+ are:

- (*Serbia*, *member of*, *World Trade Organization*): Serbia's application is still under review.
- (*Taiwan*, *member of*, *World Trade Organization*): Taiwan is already a member of the WTO through the Chinese Taipei but not in its name.
- (*Kosovo*, *member of*, *Interpol*): Kosovo's application was rejected in 2018.

Clearly, some additional textual content is needed in these cases.

7 Conclusion

We have proposed a novel approach to relation prediction by KG embedding. Our approach is based on two key ideas: Balanced Negative Sampling in the training

Head	Predicted Relation	Tail	Score	Evaluation
Allergy	health specialty	Immunology	-0.728593	
Allergy	has effect	Rhinorrhea	-0.839387	
Allergy	has cause	Allergen	-0.844231	
Allergy	health specialty	Pediatrics	-0.972245	
Allergy	health specialty	Internal medicine	-1.022068	
Allergy	instance of	Disease	-1.022585	
Allergy	drug used for treatment	Adrenaline	-1.040919	
Allergy	medical examinations	Blood test	-1.165760	
Allergy	drug used for treatment	Aspirin	-1.172504	
Allergy	health specialty	Hematology	-1.219639	
Allergy	possible treatment	Medication	-1.221141	
Allergy	symptoms	Abdominal pain	-1.228720	
Allergy	drug used for treatment	Penicillin	-1.235292	
Allergy	subclass of	Pollution	-1.283537	
Allergy	health specialty	Statistics	-1.297987	
Allergy	health specialty	Epidemiology	-1.301585	
Allergy	afflicts	Immune system	-1.374023	
Allergy	afflicts	Blood	-1.376165	
Allergy	possible treatment	Antibiotic	-1.406908	
Allergy	symptoms	Itch	-1.416243	

(a) Top-20 facts predicted from the page *Allergy*.

Head	Predicted Relation	Tail	Score	Evaluation
Henri Poincaré	employer	University of Paris	-0.358521	
Henri Poincaré	employer	École Polytechnique	-0.412040	
Henri Poincaré	occupation	Mathematician	-0.414613	
Henri Poincaré	place of death	Paris	-0.492610	
Henri Poincaré	occupation	Engineer	-0.537747	
Henri Poincaré	field of work	Number theory	-0.561962	
Henri Poincaré	student of	Charles Hermite	-0.592644	
Henri Poincaré	field of work	Epistemology	-0.605310	
Henri Poincaré	field of work	Topology	-0.678377	
Henri Poincaré	field of work	Algebraic geometry	-0.712538	
Henri Poincaré	student of	Wilhelm Wundt	-0.738560	
Henri Poincaré	field of work	Optics	-0.738636	
Henri Poincaré	notable work	Poincaré conjecture	-0.739285	
Henri Poincaré	field of work	Philosophy of science	-0.791077	
Henri Poincaré	field of work	Metaphysics	-0.822683	
Henri Poincaré	place of birth	Nancy, France	-0.824190	
Henri Poincaré	different from	Raymond Poincaré	-0.830676	
Henri Poincaré	student of	Karl Weierstrass	-0.843734	
Henri Poincaré	field of work	Set theory	-0.855588	
Henri Poincaré	field of work	Celestial mechanics	-0.877295	

(b) Top-20 facts predicted from the page *Henri Poincaré*.

Table 3: Relation prediction applied to the semantification of Wikipedia hyperlinks (green = true fact unknown by Wikidata, blue = fact already known by Wikidata, red = false statement).

of the embedding model, and Type Filtering to select candidate relations. We have shown that this approach performs well using an embedding model as simple as TransE, opening the way to robust and explainable predictions. Our results suggest that the model can be used to enrich Wikidata, by the semantification of Wikipedia hyperlinks associated with known entities.

This approach is however not yet fully automatable and performance still needs to be increased for that goal. For future work, we would like to further improve our negative sampling technique by replacing the relation with some probability that depends on the considered fact (h, r, t), instead of some fixed probability. It seems also necessary to integrate some context from textual data for example (like the description of the relations and the articles themselves) in order to help the embedding model in its choices. A fully automatized process of enriching Wikidata with semantified Wikipedia hyperlinks seems however not out of reach.

References

1. Akrami, F., Saeef, M.S., Zhang, Q., Hu, W., Li, C.: Realistic re-evaluation of knowledge graph completion methods: An experimental study. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. p. 1995–2010. SIGMOD '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3318464.3380599
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. p. 1247–1250. SIGMOD '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1376616.1376746
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating Embeddings for Modeling Multi-relational Data. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 2787–2795. Curran Associates, Inc. (2013)
4. Boschin, A.: TorchKGE: Knowledge Graph Embedding in Python and PyTorch. KDD-IWKG 2020 p. 6 (Aug 2020)
5. Galárraga, L., Symeonidou, D., Moissinac, J.C.: Rule Mining for Semantifying Wikilinks. In: LDOW@WWW (2015)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
7. Kotnis, B., Nastase, V.: Analysis of the impact of negative sampling on link prediction in knowledge graphs. arXiv preprint arXiv:1708.06816 (2017)
8. Krompaß, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Thirunarayan, K., Staab, S. (eds.) The Semantic Web - ISWC 2015. pp. 640–655. Springer International Publishing, Cham (2015)
9. Nathani, D., Chauhan, J., Sharma, C., Kaul, M.: Learning attention-based embeddings for relation prediction in knowledge graphs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
10. Nickel, M., Tresp, V., Kriegel, H.P.: A Three-way Model for Collective Learning on Multi-relational Data. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. pp. 809–816. ICML'11, Omnipress, Bellevue, WA, USA (2011)
11. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: Proceedings of the 31st Conference on Neural Information Processing Systems. Long Beach, CA, USA (Oct 2017)
12. Peng, N., Poon, H., Quirk, C., Toutanova, K., Yih, W.t.: Cross-sentence n-ary relation extraction with graph LSTMs. Transactions of the Association for Computational Linguistics **5**, 101–115 (2017). https://doi.org/10.1162/tacl_a_00049
13. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
14. Sakor, A., Singh, K., Patel, A., Vidal, M.E.: Falcon 2.0: An entity and relation linking tool over wikidata. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. p.

3141–3148. CIKM '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3340531.3412777, https://doi.org/10.1145/3340531.3412777
15. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(56), 1929–1958 (2014)
16. Suchanek, F.M., Lajus, J., Boschin, A., Weikum, G.: Knowledge Representation and Rule Mining in Entity-Centric Knowledge Bases. In: Krötzsch, M., Stepanova, D. (eds.) Reasoning Web. Explainable Artificial Intelligence: 15th International Summer School 2019, Bolzano, Italy, September 20–24, 2019, Tutorial Lectures, pp. 110–152. Lecture Notes in Computer Science, Springer International Publishing, Cham (Sep 2019)
17. Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., Gamon, M.: Representing Text for Joint Embedding of Text and Knowledge Bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1499–1509. Association for Computational Linguistics, Lisbon, Portugal (2015). https://doi.org/10.18653/v1/D15-1174
18. Trouillon, T., Dance, C.R., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Knowledge Graph Completion via Complex Tensor Factorization. arXiv:1702.06879 [cs, math, stat] (Feb 2017)
19. Wang, L., Cao, Z., de Melo, G., Liu, Z.: Relation Classification via Multi-Level Attention CNNs. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1298–1307. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/P16-1123
20. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (Dec 2017). https://doi.org/10.1109/TKDE.2017.2754499
21. Wang, R., Li, B., Hu, S., Du, W., Zhang, M.: Knowledge Graph Embedding via Graph Attenuated Attention Networks. IEEE Access **8**, 5212–5224 (2020). https://doi.org/10.1109/ACCESS.2019.2963367, conference Name: IEEE Access
22. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. p. 1112–1119. AAAI'14, AAAI Press (2014)
23. Xie, R., Liu, Z., Sun, M.: Representation learning of knowledge graphs with hierarchical types. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. p. 2965–2971. IJCAI'16, AAAI Press (2016)
24. Yang, X., Ren, S., Li, Y., Shen, K., Li, Z., Wang, G.: Relation linking for wikidata using bag of distribution representation. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Y. (eds.) Natural Language Processing and Chinese Computing. pp. 652–661. Springer International Publishing, Cham (2018)
25. Zhang, Y., Qi, P., Manning, C.D.: Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2205–2215. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1244