Modeling Paths for Explainable Knowledge Base Completion

Josua Stadelmaier and **Sebastian Padó**
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart, Germany
{josua.stadelmaier,sebastian.pado}@ims.uni-stuttgart.de

Abstract

A common approach in knowledge base completion (KBC) is to learn representations for entities and relations in order to infer missing facts by generalizing existing ones. A shortcoming of standard models is that they do not *explain* their predictions to make them verifiable easily to human inspection.

In this paper, we propose the *context path model* (CPM) which generates explanations for new facts in KBC by providing sets of *context paths* as supporting evidence for these triples. For example, a new triple *(Theresa May, nationality, Britain)* may be explained by the path *(Theresa May, born in, Eastbourne, contained in, Britain)*. The CPM is formulated as a wrapper that can be applied on top of various existing KBC models. We evaluate it for the well-established TransE model. We observe that its performance remains very close despite the added complexity, and that most of the paths proposed as explanations provide meaningful evidence to assess the correctness.

1 Introduction

Knowledge bases (KBs), such as Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014) or Yago (Suchanek et al., 2007), are structured representations of knowledge in form of entities and their respective relationships. For example, KBs can comprise facts about persons like their family relations and their occupation or facts about places like the region or country they are located in. A common application of knowledge bases is question answering systems (Bordes et al., 2014; Berant et al., 2013). KBs are also used by Google to better understand search queries, to present fact boxes and to provide explorative search suggestions (Steiner et al., 2012).

The major contemporary construction mode for KBs is collaborative, which is both a major advantage (as long as community interest persists, KBs grow over time) and a major shortcoming, since development is not directed. As a result, collaborative KBs tend to be *incomplete*. Min et al. (2013) show that in Freebase 93.8% of persons have no place of birth assigned and for 78.5% of persons, the nationality is missing. This motivates the task of knowledge base completion (KBC), i.e., the addition of correct but missing facts to existing KBs.

A common approach for KBC is to learn distributed representations for entities and relations that enable the generalization of existing connections in the KB to predict missing facts (Nickel et al., 2011; Bordes et al., 2013; Yang et al., 2015). A connection could be that the country of birth is highly correlated to the nationality of a given person. A fact about the country of birth could therefore be used as evidence when predicting a missing fact about the nationality. Such representation learning methods typically perform rather well and are simple to train. However, they crucially lack in the *explainability* that often comes with more symbolic systems: they do not justify the facts that they propose in a way that is transparent to human reviewers of the system output. Explainability of system outputs is increasingly recognized as an important component in the practical use of NLP, and more generally, AI systems (Holzinger et al., 2017; Ras et al., 2018; Ribeiro et al., 2018).

In this paper, we propose a new KBC model, the *Context Path Model* (CPM), which provides a *path-based explanation* for newly proposed facts. For example, the path (e_1, *city of birth*, *contained by*, e_2) states the country the person e_1 was born in. This path is informative to assess the correctness of the triple (e_1, *nationality*, e_2). To establish a relationship between facts and paths, the CPM explicitly includes the paths from the context of a fact t into the estimation of t's correctness. As part of this process, the CPM also estimates the paths' *rele-*

Proceedings of the Second BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 147–157
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

vance to identify those paths that provide the most convincing evidence for or against the correctness of the fact. The CPM is formulated as a wrapper that can be applied on top of various KBC models that learn a scoring function for individual facts.

We evaluate the CPM instantiated with the established TransE model as fact scorer on the FB15K dataset. We find that CPM, despite the added complexity, performs almost as well as vanilla TransE on scoring facts. The majority of paths assigned a high relevance for a given fact are either equivalent to the fact or provide strong evidence regarding its correctness.[1]

2 Background and Related Work

2.1 Knowledge Base Completion (KBC)

Knowledge bases are often formalized as a directed graph with labeled edges, here called knowledge graph. A knowledge graph G is a set of n facts, or edges, where each edge is defined as a triple t of the form (e_1, r, e_2) with entities e_1 and e_2 and relation r, $G = \{t_i\}_{i=1}^{n}$. We denote the set of entities as E and the set of relations as R. The task of KBC can then be formalized as *assessing the correctness* of a triple $t \notin G$. As usual in studies of KBC, we concentrate on the case where e_1 and e_2 are known, i.e., we add edges, but not nodes, to the graph.

2.2 Representation Learning for KBC

An important current approach to KBC is to learn distributed representations (vectors, matrices, tensors) for entities and relations and define algebraic combination operations to score the correctness of novel triples $t = (e_1, r, e_2)$. This includes models like NTM (Socher et al., 2013), TransE (Bordes et al., 2013), Bilinear (Nickel et al., 2011) and Bilinear-diag (Yang et al., 2015).

For instance, TransE represents relations in the same vector space as entities and models relations as translation from e_1 to e_2. Given respective vector representations $\boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{r} \in \mathbb{R}^d$, TransE predicts the entity that stands in relation r to e_1 as $\boldsymbol{e_1} + \boldsymbol{r}$. The representations are learned using a max-margin objective which minimizes the distance between e_2's predicted and actual positions, $\|\boldsymbol{e_1} + \boldsymbol{r} - \boldsymbol{e_2}\|$, for correct facts, and maximizes it otherwise.

Research on novel neural architectures for KBC is ongoing. Schlichtkrull et al. (2018) replace simple embedding lookups by *Relational Graph Convolutional Networks* which are used as an encoder to learn globally optimized knowledge graph representations. Shen et al. (2017) propose a dynamic memory architecture that learns to perform inference and represents the current state of the art.

2.3 Modeling Paths for KBC

Several previous studies considered paths as information sources. Lao and Cohen (2010) use random walk probabilities for paths that connect e_1 and e_2 as features for scoring the correctness of facts (e_1, r, e_2). Gardner et al. (2014) generalize the random walk approach with a relevance-based component. Unlike the “full” KBC models discussed above, however, these models do not represent entities as vectors, which prevents them from capturing entity specific information and from letting entities directly interact with relations.

Guu et al. (2015) show how vector space models like TransE (Bordes et al., 2013), Bilinear (Nickel et al., 2011) and Bilinear-diag (Yang et al., 2015) can be generalized to not only scoring the correctness of edges $t = (e_1, r, e_2)$ but also the correctness of paths $p = (e_1, r_1, ..., r_k, e_2)$. They propose a training objective that incorporates paths and demonstrate that it improves the performance of KBC models on predicting paths and on predicting single edges as well. In the case of TransE, the relations $r_1, ..., r_k$ of a path p can be represented by their composition $\boldsymbol{r_p} = \boldsymbol{r_1} + ... + \boldsymbol{r_k}$. The distance computed by TransE can then be generalized to paths as $\|\boldsymbol{e_1} + \boldsymbol{r_p} - \boldsymbol{e_2}\|$. The objective proposed by Guu et al. encourages that $\boldsymbol{e_1} + \boldsymbol{r_p}$ is learned to be close to the set of entities that are reached when traversing the knowledge graph over the edges $r_1, ..., r_k$, starting from e_1.

PTransE, proposed by Lin et al. (2015), assesses the correctness of t by considering paths that connect e_1 and e_2 and assigns them scores that aim to indicate how reliable these paths are for estimating the correctness of t. They compute the reliability scores by using a heuristic called *path-constraint resource allocation*, which is based on the sizes of entity sets that can be reached by following the relations in a path step by step. They report improvements in the KBC task over the standard TransE model. This supports the idea of modeling paths explicitly to capture the context of a triple. A similar approach by Toutanova et al. (2016) is based on Bilinear-diag instead of TransE and comprises an

[1] The model and its annotated predictions for FB15K are available at https://github.com/JosuaStadelmaier/CPM

efficient algorithm to incorporate paths.

2.4 Providing Explanations for KBC

One possibility to provide explanations for KBC predictions is to generate logical rules. In the literature, these rules are often formalized as Horn rules (Gusmão et al., 2018) such as $(e_1, r_1, e_2) \wedge (e_2, r_2, e_3) \rightarrow (e_1, r_3, e_3)$. This rule claims that the path with the relation sequence r_1, r_2 between e_1 and e_2 implies the presence of the relation r_3 between the two entities.

Galárraga et al. (2013) propose the system AMIE which mines such rules. Their approach is to adapt association rule mining to incomplete knowledge bases. Rules are assigned confidence values that state how likely the conclusion of the rule is a correct triple. While this can be used to predict and explain new facts based on a single rule, there is no clear way of combining several rules that all have the same triple as conclusion. Furthermore, these rules only make a statement about triples that actually occur in the conclusion of a rule. The rules found by Galárraga et al. always have a positive conclusion and therefore cannot provide evidence for refuting triples. In contrast, representation learning can capture the characteristics of individual entities and can take arbitrary triples as input, provided that the involved entities and relations occur in the training set.

There are several studies that analyze learned representations of neural KBC models like TransE or Bilinear-diag to find Horn rules (Yang et al., 2015) or paths in the knowledge graph (Zhang et al., 2019). While similar in motivation to our model, these approaches share the disadvantage of using a pipeline approach: The rules or paths are extracted *post hoc* and cannot be used by the representation learning step to improve the consistency of its predictions, as would be desirable.

Xie et al. (2017) propose a neural KBC model that provides an alternative kind of explainability: it learns sparse attention vectors which capture abstract concepts shared by multiple relations. Due to the sparsity of attention vectors, the connections can be visualized and interpreted.

3 Context Path Model

As stated in the introduction, the main idea of our Context Path Model (CPM) is to capture the context of a triple $t = (e_1, r, e_2)$ in the shape of the paths surrounding t. The role of the paths is as a data source for estimating the correctness of t as well as providing explanations for the estimate.

3.1 Motivation

Formally, we define a path of length k as a sequence of the form $(e_1, r_1, ..., r_k, e_2)$. Our fundamental intuition is that the correctness of triples and paths in their context can show different *degrees of correlation*, as the following examples illustrate.

Example 1: The triple $t_1 = (e_1,$ *country of birth*, $e_2)$ and the path $p_1 = (e_1,$ *city of birth*, *contained by*, $e_2)$ are logically equivalent. Thus, any KBC model of correctness should assign the same score to p_1 and t_1: If a KB contains p_1, it should also contain t_1. Conversely, the absence of p_1 can be taken as evidence against t_1.

Example 2: The path $p_2 = (e_1,$ *lived in country*, *neighboring country*, $e_2)$ has a weak connection with t_1: it is not unlikely to have lived in a country that adjoins the country of birth. However, as countries very often have several neighboring countries, p_2 cannot provide strong evidence either for or against the correctness of t_1.

We currently concentrate on cases of positive correlation, like Ex. 1, where either the presence of p is evidence in favor of t, or the absence of p is evidence against t. [2] Even though negative correlation (e.g., the presence of p providing evidence against t) is in principle also informative, it is more difficult to capture empirically, since it requires learning exclusion relationships among paths.

3.2 Definition of the Context Path Model

To capture those connections, a KBC model needs to score the *correctness* of paths as well as determine their *relevance* as indicator for the correctness of triples t, that is, the strength of the correlation of the correctness scores of p and t.

We denote the set of paths that are used to model the context of a triple t, its *context paths* as P_t (see Section 3.4 for details). Based on P_t, the CPM estimates the correctness of t, $c(t, P_t)$, as follows:

$$c(t, P_t) = \sum_{p \in P_t} \frac{\rho(t, p)}{Z(t, P_t)} \cdot c(p), \tag{1}$$

$$Z(t, P_t) = \sum_{p \in P_t} \rho(t, p). \tag{2}$$

[2] Absence of paths from a KB is a weaker indicator than presence, since paths can be missing either because they are actually incorrect, or because at least one of its constituent edges are erroneously missing.

Thus, the correctness of a triple t is a weighted average of the *correctness scores* $c(p)$ of its context paths, with the weights given by the normalized *relevance scores* $\rho(t,p)$ of paths p for t, that is, the correlation between the correctness of t and p.

Since we want $c(p)$ to be interpretable as the probability of p being correct, we restrict the range of $c(p)$ to $[0,1]$. We only require $\rho(t,p)$ to be non-negative, since the division by $Z(t,P_t)$ directly yields normalized relevance scores. The property of $c(p)$ being normalized carries over to $c(t,P_t)$ which also has a range of $[0,1]$. Appropriate choices for $c(p)$ and $\rho(t,p)$ are discussed in the following subsubsections.

Applied to Ex. 1 from above, the path $p_1 = (e_1,$ *city of birth*, *contained by*, $e_2)$ should be assigned a high relevance score $\rho(t_1,p_1)$. If p_1 is correct, $c(p_1)$ should be close to 1. A high relevance score combined with a high correctness scores pushes $c(t_1,P_{t_1})$ towards 1. If p_1 is not correct, $c(p_1)$ should be close to 0. In this case a high relevance score is combined with a low correctness score, which pushes $c(t_1,P_{t_1})$ towards 0. Both effects match the intended meaning of $c(t_1,P_{t_1})$ to represent the correctness of t_1. In Ex. 2, the path p_2 has a low relevance for t_1 and should be assigned a low relevance score $\rho(t_1,p_2)$. Since correctness scores are restricted to $[0,1]$, the effect of $c(p_2)$ on $c(t_1,P_{t_1})$ is small, whether $c(p_2)$ is high or low. This properly models that the correctness of p_2 has little effect on the correctness of t_1.

The CPM can serve as a source of explanations for its predictions by considering the context paths that have the highest relevance scores for a triple t. By normalizing relevance scores by $Z(t,P_t)$ (Equation 2), we obtain the normalized weight with which the correctness of a context path contributes to the correctness score of the triple. Furthermore, if $c(p) \approx 1$, p represents evidence in favor of the correctness of t, and if $c(p) \approx 0$, p is evidence against the correctness of t.

3.2.1 Estimating Context Paths Correctness

The first major parameter of the CPM is the context path correctness score $c(p)$, which is required to have two properties: It needs to be able to model paths and its output has to lie in $[0,1]$. The first property is fulfilled by all *composable* KBC models like TransE (Bordes et al., 2013), Bilinear (Nickel et al., 2011) and Bilinear-diag (Yang et al., 2015), that is, models which can produce a functional representation $\boldsymbol{r_p}$ for a path p as a function of the representation of its edges. Regarding the second property, models that are distance-based and do not directly fulfill it, can be adapted as follows. Since they are composable, we can compute the distance between the end of the path, e_2, and the path representation applied to the start of the path, as $dist(\boldsymbol{e_2}, f(\boldsymbol{e_1},\boldsymbol{r_p}))$, and map it to $[0,1]$ via a logistic transformation σ of the negated distance.

For the example of the TransE model (Bordes et al., 2014), the path representation is simply a translation defined by the addition of the relation vectors, $\boldsymbol{r_p} = \sum_{r_i \in p} \boldsymbol{r_i}$ and $f = \lambda x, y \,.\, x + y$. The correctness score for a path is then defined as a transformation of the distance:

$$c(p) = \sigma(-\|\boldsymbol{e_1} + \boldsymbol{r_p} - \boldsymbol{e_2}\|_2^2 + \boldsymbol{b_1}^\intercal \boldsymbol{r_p}) \qquad (3)$$

where $\boldsymbol{b_1} \in \mathbb{R}^d$ is a path-specific bias parameter. This model has $d \cdot (|R| + |E| + 1)$ parameters.

3.2.2 Estimating Context Path Relevance

The second major parameter of the CPM is the context path relevance score $\rho(t,p)$. To our knowledge, no such models have been proposed in the literature, so we propose a simple model which is again inspired by the translation-based TransE model. To estimate ρ for a path $p = (e_1, r_1, ..., r_k, e_2)$ and a triple t, we represent the path as sequence of relations $r_1, ..., r_k$ in order to abstract away from the entities e_1 and e_2 and learn general regularities.[3] We represent each relation r by one vector $\boldsymbol{a_r} \in \mathbb{R}^d$ in order to recognize patterns in the compositional path representation $\boldsymbol{r_p}$ that indicate how relevant the path p is for the relation r. The exponential function is applied to obtain non-negative scores:

$$\rho(t,p) = \exp(\boldsymbol{a_r}^\intercal \boldsymbol{r_p} + \boldsymbol{b_2}^\intercal \boldsymbol{r}) \qquad (4)$$

where $\boldsymbol{b_2} \in \mathbb{R}^d$ is a bias parameter to enable relation specific scaling of $\boldsymbol{a_r}^\intercal \boldsymbol{r_p}$. This model has $d \cdot (|R| + 1)$ parameters.

3.3 Training the Context Path Model

We split the training process into two steps to first learn the parameters of $c(p)$ and then the parameters of $\rho(t,p)$. Learning $\rho(t,p)$ and $c(p)$ jointly could lead to $c(p)$ being influenced by the relevance of p for t, which is undesirable since we want to guarantee that $c(p)$ is interpretable in terms of the correctness of p.

[3] We define ρ as a generic function of t and p to indicate that extensions of the CPM could also make use of the entity representations.

Training $c(p)$. Following the training regimen of Guu et al. (2015), we first train $c(p)$ on the edges of the knowledge graph G before training it on longer paths. This gives the model the opportunity to build up paths from meaningful edges.

We train $c(p)$ on a standard contrastive cross-entropy loss that provides a good fit for the probabilistic interpretation of $c(p)$ that we aim for:

$$-\sum_{p \in P} \frac{\log c(p)}{|P|} - \sum_{p' \in P'} \frac{\log(1 - c(p'))}{|P'|} \tag{5}$$

where P is the set of correct paths and P' a set of corrupted paths. For single edge training, we use $P = G$. In the subsequent path training, we sample a set of positive informative paths P as described below in Section 3.4.

We also need to generate a set of negative samples P', which we generate in the same way as Guu et al. (2015) by *type-matched corruption*– see Section 4.2 for a discussion. Given a path $p = (e_1, r_1, \dots, r_k, e_2)$, let $\mathcal{F}(p)$ be the set of *final entities* of p that can be reached when traversing G via the relations $r_1, \dots, r_k$ starting from e_1. We are guaranteed to corrupt p if we replace e_2 with any entity $e_2' \notin \mathcal{F}(p)$ but matches the type of r_k, i.e., $e_2' \in D_2(r_k)$, with the *right domain* $D_2(r)$ defined as:

$$D_2(r) = \{e_2 \mid \exists e_1 : (e_1, r, e_2) \in G\} \tag{6}$$

Analogously, we define $\mathcal{I}(p)$ as the set of *initial entities* of p. We corrupt e_1 by replacing it with an entity in $D_1(r_1) \setminus \mathcal{I}(p)$, where $D_1(r)$ is the analogous *left domain* of r. In the case of TransE, the parameters to be updated are all $\boldsymbol{e_i}, \boldsymbol{r_j}$ as well as $\boldsymbol{b_1}$ (cf. Equation (3)).

Training $\rho(t,p)$. The relevance scores $\rho(t,p)$ use a similar cross-entropy loss based on $c(t, P_t)$:

$$-\sum_{t \in G} \frac{\log c(t, P_t)}{|G|} - \sum_{t' \in G'} \frac{\log(1 - c(t', P_{t'}))}{|G'|} \tag{7}$$

where G, the KB, is the set of correct triples and G' is a set of triples with either e_1 or e_2 corrupted as above. The objective aims to assign correct triples a score of 1 and incorrect triples a score of 0, which, together with the fixed semantics of $c(p)$, encourages $\rho(t,p)$ to estimate the relevance of paths p for t. Only the parameters of ρ, namely $\boldsymbol{a_r}$ and $\boldsymbol{b_2}$ (compare Equation (4)) are updated.

3.4 Selecting Context Paths

The final part of the Context Path Model (CPM) is the *selection of informative context paths*. Since the number of paths grows exponentially in the path length, it is infeasible to include all paths in the CPM. We now propose several criteria to limit the set of informative context paths P_t for a given triple $t = (e_1, r, e_2)$ to keep the model tractable.

Closed paths. Paths that connect the two entities of a triple express a semantic relation between them. We therefore restrict paths to start with e_1 and end with e_2, i.e., to be closed paths.

Limited length. We limit path lengths to $k \leq 3$. This effectively reduces the number of potential paths and keeps the paths between e_1 and e_2 relatively easy to understand.

Filtering redundant paths. To be able to traverse edges of the knowledge graph in both directions, we need to add the inverse edge (e_2, r^{-1}, e_1) for each edge $(e_1, r, e_2) \in G$ to the knowledge graph G. This has the unwanted consequence that we obtain *redundant* paths which comprise two successive, mutually inverse relations like (e_1, *country of birth*, *contains*, *contains*$^{-1}$, e_2) which is judged to be highly relevant for *country of birth* but for trivial reasons.

We consider the domains $D_1(r_i)$ and $D_2(r_i)$ (compare Equation (6)) to filter out trivial paths effectively. A path $(e_1, r_1, ..., r_k, e_2)$ is defined as trivial if e_1 occurs in any domain of $r_1, ..., r_k$ except $D_1(r_1)$ or e_2 in any domain except $D_2(r_k)$.

This general definition has the benefit of capturing cases of redundant paths caused by relations in the KB that are semantically, but not formally, inverses – such as the relations *contains* and *contained by*. It can also be too strict: E.g., the context path $p = (e_1$, *mother of*, *mother of*, $e_2)$ for the triple $t = (e_1$, *grandmother of*, $e_2)$ is excluded if the mother of e_2 participates in the relation *grandmother of*. This does however not pose a major problem in practice.

Negative context paths. The paths described so far can only be used by the CPM as positive evidence for the correctness of a triple. However, the CPM can also use incorrect (i.e., correctly absent) paths as negative evidence (i.e., as evidence that triples are incorrect). We now describe how such paths can be found.

Entities		14,951
Relations		1,345
Triples	Train	483,142
	Validation	50,000
	Test	59,071
Paths of length 2	Train	3,110,893
	Test	81,124
Paths of length 3	Train	3,711,317
	Test	101,717

Table 1: Statistics on the FB15K dataset

Assuming the first three criteria described above, let P_r be the set of correct context paths corresponding to all triples (e_1, r, e_2) for a fixed relation r. Let furthermore S_r be the set of relation sequences occurring in paths $p = (e_1, r_1, ..., r_k, e_2) \in P_r$. We can then define the set of context paths P_t:

$$P_t = \{p \mid (r_1, ..., r_k) \in S_r \wedge \mathcal{I}(p) \cup \mathcal{F}(p) \neq \emptyset\} \quad (8)$$

In addition to the positive informative paths described so far, P_t contains incorrect paths that connect e_1 and e_2 by corrupted relation sequences that conform to the criteria described above. The use of relation sequences that occur in context paths of other facts about the same relation makes it more likely that incorrect paths are relevant for t.

4 Experimental Evaluation

4.1 Dataset

We evaluate our approach on the FB15K dataset extracted by Bordes et al. (2013) from the FreeBase knowledge base. Table 1 shows the statistics of this dataset, including the number of context paths according to the definition in Section 3.4.

4.2 Experimental Setup

We instantiate the edge scoring model of CPM with the TransE model (Bordes et al., 2013), as shown in Eq. (3). We follow the two-step training regimen as described in Section 3.3. We train 100-dimensional vectors for all representations learned by the model (cf. Section 3.2). Optimization proceeds by applying the gradient-based optimizer Adam (Kingma and Ba, 2015) to minibatches of size 300. We use the learning rate of 0.001 for all parts of the model with the exception of $c(p)$ during path training, where we use 0.0001 based on performance on the validation set.

Choice of negative samples. Since KBs ideally contain only correct information, KBC methods generally need to generate incorrect samples synthetically. This is generally done by corrupting either the first entity e_1 or the last entity e_2 in a path p to obtain negative samples $\mathcal{N}(p)$. Negative samples are used as parts of the ranking problems both at train time (cf. Section 3.3) and at test time. The generation of negative samples is therefore a crucial part of the experimental setup. Unfortunately, there is little consensus on the details of the process in the literature. We discuss the two major approaches below.

The first approach, *random corruption*, corrupts paths by replacing e_1 or e_2 by random entities from the KB (Bordes et al., 2013; Yang et al., 2015). The advantage of this approach is that a large number of negative samples can be generated easily – at the same time, most corrupted paths are arguably not particularly plausible confounders, as when a person is replaced by a country or a record. An alternative approach, *type-matched corruption* (Guu et al., 2015), employs only confounders seen with the same sequence of relations as the original entity (cf. Section 3.3 for a formal definition). This generally ensures that the confounders are plausible. On the downside, there are typically fewer such confounders.

For our study, we adopt the type-matching corruption setup, which we find more appropriate in the context of explainable KBC. For negative samples with incorrect types, the most natural reason for rejection is simply the domain mismatch, while the type-matching setting requires the models to capture fine-grained semantics within domains.

As a result, the evaluation numbers that we report are not directly comparable to numbers obtained with the random corruption approach, and tend to be higher. This is because the smaller numbers of negative samples lead to simpler ranking problems. The average size of $\mathcal{N}(t)$ for FB15k test triples t is 1,738 in the type-matching setting and 29,543 in the setting without type-matching. As negative samples from the training can end up in $\mathcal{N}(t)$ for test triples t, the average number of unseen negative samples in $\mathcal{N}(t)$ is 765 in our setup.

We exclude negative samples that result in correct paths from the training or validation set. Similarly, context paths sampled for training or validation are excluded from the test set.

Model	Training	H@10		MQ	
TransE	Edges	**90.2**		**97.5**	
TransE	Paths	84.2	(-6.7%)	97.1	(-0.4%)
CPM	Paths	83.1	(-7.9%)	96.7	(-0.8%)
CPM$\backslash t$	Paths	80.0	(-11.3%)	96.2	(-1.3%)

Table 2: Results for Evaluation 1 (fact correctness). Numbers in brackets give relative deterioration compared to TransE (edge). CPM$\backslash t$ is the CPM ignoring all information about the target triple itself.

	Edge training		Path training	
Length	H@10	MQ	H@10	MQ
1	**90.2**	**97.5**	84.2	97.1
2	73.4	94.1	**82.7**	**97.5**
3	54.0	89.0	**64.4**	**93.3**

Table 3: Results for Evaluation 2 (path correctness), varying path length and training regimen. Best results for each length shown in boldface.

4.3 Evaluation 1: Fact Correctness

We carry out three evaluations. We start with the traditional task of predicting individual facts (edges). We directly define the evaluation metrics for paths rather than facts for re-use in Evaluation 2 at the path level. The metrics apply to edges because they are paths of length 1.

Evaluation metric. We apply the commonly used ranking metric *hits at 10* (H@10), which is defined as the percentage of correct paths that are ranked within the top 10 of their respective negative samples. Additionally, we use the metric *mean quantile* (MQ), proposed by Guu et al. (2015), which computes the share of incorrect paths ranked lower than the correct path:

$$MQ = \frac{1}{|P|} \sum_{p \in P} \frac{|\{p' \in \mathcal{N}(p) \mid c(p') < c(p)\}|}{|\mathcal{N}(p)|} \quad (9)$$

In contrast to H@10, MQ accounts for the size of $\mathcal{N}(p)$. For $1 \leq |\mathcal{N}(p)| \leq 9$, H@10 always outputs 1. In the fact correctness evaluation, this is the case for 1.5% of used test facts. We exclude 1143 facts with $|\mathcal{N}(p)| = 0$ from the test set because both H@10 and MQ always output 1 in these cases.

Results. Table 2 presents the results of the ranking evaluation for fact prediction. We compare the full CPM model against TransE, the edge scorer "inside" our CPM (cf. Section 4.2), in both its edge-trained and path-trained versions[4] (cf. Section 3.3). We also consider CPM$\backslash t$, a variant of the CPM that excludes t from P_t, that is, does not use any information about the predicted triple. This model examines to which degree the correctness of triples t can be predicted purely on the basis of its KB context. This gives us four models to compare.

We find that the CPM performs somewhat worse than the best model overall, the edge-trained TransE, for both metrics: the drop is noticeable for H@10, and mild for MQ. We believe that the drop is primarily due to two factors: First, path training gives rise to a different optimization problem from edge training, which appears to be more difficult on the FB15K dataset.[5] In fact, as the second row shows, training the original TransE on paths leads to a comparable drop in H@10. Second, the bad results for CPM$\backslash t$, which are still substantially worse than for the plain CPM, indicate that, unsurprisingly, the most important source of information for the prediction of a single triple is the semantics of the triple itself. In other words, the contextual component that CPM adds does not provide additional support to single edge prediction at the technical level (we consider the produced justifications in the third evaluation).

In sum, CPM introduces a mild loss of quality in the prediction of individual facts. Given that the CPM has a more complex objective – modeling the correctness of facts/paths as well as modeling justifications – we see this nevertheless as a promising first evaluation result.

4.4 Evaluation 2: Path Correctness

The second evaluation concentrates on the CPM and its performance on the task it is designed for, namely predicting the correctness of longer paths. Table 3 shows results separated by path length (1 through 3). The results for path length 1 are, by definition, identical to the corresponding conditions in Evaluation 1, with an advantage for single-edge training. This effect reverses for the longer paths: while the edge-trained model loses substantially in quality because it fails to capture dependencies among edges, the path-trained model holds up well

[4] TransE can be seen as a special case of the CPM when all paths except for the triple itself are assigned a relevance of 0. The reported TransE scores are measured on the instantiation of $c(\cdot)$ with TransE.

[5] Path training appears to be beneficial on other datasets, as reported by Guu et al. (2015).

for longer paths. We see these results as validation of our choice of a path-based training regimen (Section 3.3).

4.5 Evaluation 3: Path Relevance

Our third evaluation focuses on relevance, that is, the relation between paths and the triples that they are supposed to provide evidence for or against. Since our goal is to use these paths as human-interpretable justifications for the triples, we perform a small annotation study of the CPM output.

Dataset. We manually select 24 relations for annotation whose relevance can arguably be judged without in-depth expertise of specific domains. The selected relations account for 13.2% of the facts in the test set. For each relation, we randomly sample two correct facts from the test set and obtain incorrect facts by corrupting either e_1 or e_2. This results in 48 positive and 48 negative facts. We furthermore exclude 17 triples for which no path (other than the trivial t itself) was found with a normalized relevance score of at least 10%. For the remaining 79 triples, we annotate all paths with a normalized relevance score of at least 5% – again, with the exception of t itself. We also do not consider paths with relations from the FB15K domains *dataworld* and *commons* because they encode only KB-specific meta information. This results in on average 2.45 context paths annotated per triple, accounting for 80% of the assigned relevance scores.

Annotation. As motivated in Section 3.1, the relevance of context paths P_t describes how strongly their correctness is correlated with the correctness of t. In the annotation we distinguish between three levels (categories) of relevance:

1. **Equivalent:** The correctness of p is logically equivalent to the correctness of t.

2. **Probable:** The path p being correct makes the correctness of t significantly more likely, or p being incorrect makes the correctness of t significantly less likely. However, p provides no guarantee for the (in-)correctness of t.

3. **Unrelated:** The correctness of p and t are not strongly correlated. This class comprises all cases that are not in category 1 or 2.

Qualitative Analysis. Table 4 shows examples of the three annotated categories, both for positive cases (presence of p supports t) and negative cases (absence of p casts doubt in t). Negative cases are marked with asterisks (*), and since all absent paths are corrupted versions of paths in the KB, the point of corruption is marked as well.

The 'equivalent' category shows two cases of equivalence between two Freebase relations – one positive (*profession* is supported by *people_with_profession*$^{-1}$) and one negative (*islands_in_group* is implausible if not accompanied by *island_group*$^{-1}$) – as well as one case of mutual entailment (the CAF has a football league iff there is a team that is a football team and plays in the CAF). The 'probable' category contains cases of defeasible inferences – e.g., Hindi is the most widely spoken language in India, but only by just over half of the population. The 'unrelated' category, finally, shows some paths that are largely irrelevant for their facts (e.g., someone is born in a place vs. someone often eats at a restaurant that uses the same currency as the birth place). The examples demonstrate that CPM is indeed capable of capturing meaningful relations between triples and the paths in its context.

Quantitative Analysis. Table 5 shows that just over half of all pairs we consider falls into the 'equivalent' category. These pairs are assigned a mean relevance of 0.47, and their share of the total sum of relevance scores is 72%. Another quarter of the annotated pairs falls into the 'probable' category, with a considerably lower mean relevance of 0.15, and a share of 15% of the relevance scores. The final quarter of pairs makes up the 'unrelated' category, with similar mean relevance and share of relevance scores.

We see this outcome as rather positive: about half of the paths identified by the CPM are equivalent to the triple in question, with another quarter providing probable evidence. Furthermore, the relevance scores manage very well to separate the 'equivalent' and 'probable' categories. The separation between 'probable' and 'unrelated' is weak, but may be due to our exclusion of the lowest-relevance paths from annotation (see above): these would arguably mostly be mostly 'unrelated' and thus decrease the mean relevance for this category.

5 Conclusion

This paper has considered the generation of explanations for predictions of facts in knowledge base completion (KBC). Our contribution is the *Context Path Model* (CPM), which provides explanations

	Triple	Context path
equivalent	(***Jon Favreau***, *profession*, ***Film director***)	(***Jon Favreau***, *people_with_profession*$^{-1}$, ***Film director***)
	(***Football***, *leagues*, ***Confed. of African Football***)	(***Football***, *teams*, ***Zimbabwe national football team***, *league_participation/team*$^{-1}$, ***Confed. of African Football***)
	*(***Hawaiian Islands***, *islands_in_group*, ***Ireland***)	*(***Hawaiian Islands*** [corrupted from: ***British Isles***], *island_group*$^{-1}$, ***Ireland***)
probable	(***Feroz Khan***, *languages*, ***Hindi***)	(***Feroz Khan***, *nationality*, ***India***, *countries_spoken_in*$^{-1}$, ***Hindi***)
	(***Naval Postgraduate School***, *containedby*, ***USA***)	(***Naval Postgraduate School***, *headquarters/state*, ***California***, *representatives*, ***Richard Nixon***, *nationality*, ***USA***)
unrelated	(***Nashua***, *people_born_here*, ***Mandy Moore***)	(***Nashua***, *currency*, ***US Dollar***, *liabilities_in_currency*$^{-1}$, ***Starbucks***, *eats_at*$^{-1}$, ***Mandy Moore***)
	*(***Jared Harris***, *parents*, ***Aaron Spelling***)	*(***Jared Harris***, *award_nominee*$^{-1}$, ***Mad Men*** [corrupted from ***Beverly Hills, 90210***], *tv_program_creator*, ***Aaron Spelling***)

Table 4: Examples of triple–path pairs, with entities in boldface, and simplified freebase relations. Incorrect triples/paths marked with *, and point of corruption marked.

Annotation category	equiv.	prob.	unrel.
Number of pairs	98	49	47
Share of pairs	51%	25%	24%
Mean Relevance ρ	0.47	0.15	0.12
Share of total $\sum \rho$	72%	15%	13%

Table 5: Statistics for annotated triple–path pairs

by identifying context paths which are highly correlated with the fact: if the path is in the KB, then the triple should be as well; conversely, if the path is not in the KB, then the triple should not be either.

We demonstrate the usefulness of our model by instantiating its fact scorer with a simple but effective KBC model, TransE (Bordes et al., 2013). We find that the performance of the CPM is close to TransE, and manual evaluation confirms that most of the paths the model uses as explanation are meaningful and provide evidence for assessing the correctness of facts. This shows the potential of using paths as explanations for KBC predictions.

Beyond the KBC setting, the output of the CPM can also arguably be useful for a structural analysis of knowledge bases, for example the systematic identification of equivalences among relations or between relations and paths, to improve the consistency of the KB, e.g., by replacing equivalent paths by a canonical version.

The current study has three main limitations. First, we only apply the CPM to TransE. Future work should investigate the practical usefulness of the CPM for other composable KBC models like Bilinear-diag. Second, we use strong heuristics to limit the set of paths under consideration; future work should attempt to relax these. Third, the current CPM can only capture paths that are symmetrically (cor-)related with the fact in question, corresponding to strict or probabilistic entailment. A promising avenue for future work is to generalize the model to asymmetrical relations, i.e., find paths that represent (just) necessary or sufficient conditions for a fact, in order to enable a more comprehensive analysis of the inferential structures in KBs (Hitzler et al., 2009).

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, WA.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, Vancouver, Canada.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 2787–2795.

Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Machine Learning and Knowledge Discovery in Databases*, pages 165–180. Springer.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. Amie: Association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the International Conference on World Wide Web*, pages 413–422, Rio de Janeiro, Brazil.

Matt Gardner, Partha Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 397–406, Doha, Qatar.

Arthur Colombini Gusmão, Alvaro Henrique Chaim Correia, Glauber De Bona, and Fábio Gagliardi Cozman. 2018. Interpreting embedding models of knowledge bases: A pedagogical approach. In *Proceedings of the ICML Workshop on Human Interpretability in Machine Learning*.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 318–327, Lisbon, Portugal.

Pascal Hitzler, Markus Krotzsch, and Sebastian Rudolph. 2009. *Foundations of semantic web technologies*. CRC Press.

Andreas Holzinger, Chris Biemann, Constantinos S. Pattichis, and Douglas B. Kell. 2017. What do we need to build explainable AI systems for the medical domain? *CoRR*, abs/1712.09923.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, San Diego, CA.

Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 705–714, Lisbon, Portugal.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–782, Atlanta, GA.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the International Conference on Machine Learning*, pages 809–816, Bellevue, WA.

Gabriëlle Ras, Marcel van Gerven, and Pim Haselager. 2018. Explanation methods in deep learning: Users, values, concerns and challenges. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 19–36. Springer.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, LA.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607. Springer.

Yelong Shen, Po-Sen Huang, Ming-Wei Chang, and Jianfeng Gao. 2017. Modeling large-scale structured relationships with shared memory for knowledge base completion. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 57–68, Vancouver, Canada.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 926–934.

Thomas Steiner, Ruben Verborgh, Raphaël Troncy, Joaquim Gabarro, and Rik Van De Walle. 2012. Adding realtime coverage to the google knowledge graph. In *Proceedings of the International Semantic Web Conference*, pages 65–68.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706, New York, NY.

Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. 2016. Compositional learning of embeddings for relation paths in knowledge base and text. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1434–1444, Berlin, Germany.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85.

Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. 2017. An interpretable knowledge transfer model for knowledge base completion. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 950–962, Vancouver, BC.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations*, San Diego, CA.

Wen Zhang, Bibek Paudel, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Interaction embeddings for prediction and explanation in knowledge graphs. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 96–104.