Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Bachelorarbeit

Modeling paths in knowledge graphs for context-aware prediction and explanation of facts

Josua Stadelmaier

Course of Study:	Informatik
Examiner:	Prof. Dr. Sebastian Padó Dr. Roman Klinger
Supervisor:	Prof. Dr. Sebastian Padó
Commenced:	October 1, 2018
Completed:	April 1, 2019

Abstract

Knowledge bases are an important resource for question answering systems and search engines but often suffer from incompleteness. This work considers the problem of knowledge base completion (KBC). In the context of natural language processing, knowledge bases comprise facts that can be formalized as triples of the form (*entity 1*, *relation*, *entity 2*). A common approach for the KBC problem is to learn representations for entities and relations that allow for generalizing existing connections in the knowledge base to predict the correctness of a triple that is not in the knowledge base.

In this work, I propose the *context path model*, which is based on this approach. In contrast to existing KBC models, it also provides explanations for predictions. For this purpose, it uses paths that capture the context of a given triple. The context path model can be applied on top of several existing KBC models. In a manual evaluation, I observe that most of the paths the model uses as explanation are meaningful and provide evidence for assessing the correctness of triples. I also show in an experiment that the performance of the context path model on a standard KBC task is close to a state of the art model.

Kurzfassung

Wissensbasen sind eine wichtige Ressource für Frage-Antwort-Systeme und Suchmaschinen. Oft sind Wissensbasen jedoch unvollständig. Diese Arbeit beschäftigt sich mit dem Problem der Vervollständigung von Wissensbasen. Im Kontext der maschinellen Sprachverarbeitung bestehen Wissensbasen aus Fakten, die als Tripel der Form (*Entität 1*, *Relation*, *Entität 2*) dargestellt werden können. Ein häufiger Ansatz für die Vervollständigung von Wissensbasen ist, Repräsentationen für Entitäten und Relationen zu lernen, die es erlauben, bestehende Zusammenhänge in einer Wissensbasis zu erfassen und zu generalisieren, um damit die Korrektheit von Tripeln vorherzusagen, die nicht in der Wissensbasis vorkommen.

In dieser Arbeit stelle ich das *context path model* vor, welches auf diesem Ansatz basiert. Im Gegensatz zu bisherigen Modellen generiert es auch Erklärungen für Vorhersagen. Für die Erklärungen werden Pfade genutzt, die den Kontext des betrachteten Tripels erfassen. Das vorgeschlagene Modell kann auf mehrere bestehende Modelle zur Vervollständigung von Wissensbasen angewandt werden. Die manuelle Auswertung von Erklärungen zeigt, dass das context path model zum Großteil sinnvolle Pfade ausgibt, die Hinweise über die Korrektheit des betrachteten Tripels geben. In einem Experiment zeige ich außerdem, dass das context path model die Genauigkeit eines aktuellen Modells bei der Vorhersage von neuen Fakten fast erreicht.

Contents

1 Introduction

The task of knowledge base completion (KBC) became an active field of natural language processing (NLP) research in the last years.

In this work, the term knowledge base refers to a structured representation of knowledge in form of entities and their respective relationships. Knowledge bases such as Freebase [BEP+08], Wikidata [VK14] or Yago [SKW07] for example comprise facts about persons like their family relations and their occupation or facts about places like the region or country they are located in.

Knowledge bases are applied, for example, in Google search to understand search queries more deeply, to present fact boxes that might already answer a query and to provide explorative search suggestions [SVT+12]. Another common application of knowledge bases is question answering systems. Such systems aim to answer natural language questions using facts of knowledge bases [BCFL13; BWU14].

Knowledge base completion denotes the prediction of new facts that are not present in a given knowledge base. This task is motivated by knowledge bases that are collaboratively built and therefore tend to be incomplete. Min et al. show that in Freebase 93.8% of persons have no place of birth assigned and for 78.5% of persons, the nationality is missing [MGW+13].

In NLP, knowledge bases are often formalized as a directed graph with labeled edges, here called knowledge graph.[1] A knowledge graph G_{KB} consisting of n facts is defined as a set of edges where each edge is denoted as a triple t of the form (e_1, r, e_2) with entities e_1 and e_2 and relation r:

$$G_{KB} = \{t_i\}_{i=1}^{n} \tag{1.1}$$

[1] In the context of NLP research, the term knowledge graph is used with different interpretations. Like Guu et al. [GML15], I use the term knowledge graph when referring to the mathematical representation of a knowledge base as graph.

This definition identifies an edge not only by the entities e_1 and e_2 it connects, but also by its label r. This allows two entities to be connected by several edges when these edges represent different relations. It is also possible that an entity is connected to several entities over the same relation. This is necessary to represent, for instance, a person having several children.

The task of KBC can then be formalized as assessing the correctness of a triple t that is not an element of the given knowledge graph.

A common approach for KBC is to learn representations for the entities and relations that allow for generalizing existing connections in the knowledge base to predict new facts. A simple example for this is facts about family relations like the parents and grandparents of a given person. In this example a KBC model should capture the relationship that grandparents are the parents of the parents. Using this learned relationship the model could predict a missing fact about the grandparents for a certain person if both facts about the parents and their respective parents are given. While the missing fact can be strictly concluded in this example, KBC models are also interested in finding less strict but very likely connections in knowledge bases.

With the popularity and success of complex statistical models in the last years, the explainability of predictions performed by artificial intelligence systems suffered [Hol18]. This has led to growing interest in explainable artificial intelligence. The predictions of automatic methods typically do not provide absolute certainty. Applied to KBC, predictions for missing facts often have to be verified by humans before they can be added to the knowledge base. Not just providing a confidence score for the correctness of a triple but also paths that lead to the prediction can help the reviewer in his own assessment. When applied to a question answering system, these explanations can provide additional information the user might also be interested in like the exact city of birth and not only the country of birth.

In this thesis, I propose a new KBC model, the *context path model* (CPM). For estimating the correctness of a triple, the CPM explicitly models the paths that capture the context of the triple. More concretely, the CPM aims to find paths that can provide evidence for or against the correctness of the triple. Those paths can then be used as explanation for the prediction.

Formally, I define a path of length k as a sequence of the form $(e_1, r_1, ..., r_k, e_2)$. A path could for example start with an entity representing a person, go over the edges *city of birth* and *contained by* and end in an entity representing a country. This path states the country

the person was born in. Such a path might give a hint for a fact about the nationality relation of this person. This motivates a core element of my approach, namely to model the relevance of paths for facts about a certain relation.

While it is natural to use paths known to be correct to find evidence for the correctness of a triple, refuting a triple may require also taking into account potentially incorrect paths that are not present in the knowledge graph. Transferred to the example from above, this means that knowing that a person is not born in a town of a certain country makes it less likely that the person has nationality of that country. Due to the incompleteness assumption of knowledge bases, paths are not necessarily incorrect if they cannot be found in the knowledge graph. For that reason, the second core element of the context path model deals with scoring the correctness of paths.

In order to compute the correctness score for a given triple, the context path model combines the relevance scores and correctness scores for paths that capture the context of the triple in the knowledge graph.

I conduct experiments on a knowledge graph extracted from Freebase. The experiments show that most of the paths the model uses as explanation are meaningful and provide evidence for assessing the correctness of triples. The performance of the context path model on a standard KBC task is close to a state of the art model.

My thesis is structured as follows: In Chapter 2, I give a brief introduction to the machine learning concepts that I use in my work. Then I describe more detailed how representation learning can be applied to KBC and I discuss several existing KBC models. In Chapter 3, I define the context path model. The experiments in Chapter 4 aim to evaluate the meaningfulness of explanations and the performance of the CPM on the KBC task. Chapter 5 concludes my work and names future research directions.

2 Background

2.1 Machine Learning for KBC

As described in Chapter 1, the task of knowledge base completion can be formalized as predicting the correctness of a triple t. This can be done by learning a real valued function $score(t)$ that outputs high values for correct and low values for incorrect triples. The term *learning* refers to parameters Θ of $score(t)$ which determine the semantics of $score(t)$. The parameters Θ are typically real valued and can have the form of vectors or matrices, for example.

To specify the goal of learning formally, an objective $J(\Theta)$ is defined, which measures how well $score(t)$ fulfills the intended prediction task on a given data set. Often it is formulated as loss function which takes small values if predictions are correct.

The parameters Θ are learned based on a training data set which provides examples for correct and incorrect triples. The process of learning uses optimization techniques that minimize the loss function with respect to the parameters Θ.

Very commonly, optimizers are gradient based. For a given training data set, they take steps in the direction of the negative gradient to update the parameters Θ to Θ' in order to minimize $J(\Theta)$. In its most basic form, this method is called *gradient decent* and can be described formally as follows:

$$\Theta' = \Theta - \gamma \nabla J(\Theta) \tag{2.1}$$

The step size γ is also referred to as learning rate. Gradient based optimizers require $J(\Theta)$ to be differentiable. At the beginning of the training, the parameters Θ need to be initialized. For example, they can be drawn from a continuous probability distribution. [WFH11]

While the parameters Θ are optimized by the described method, the learning rate and the choice of parameter initialization are not part of the objective. Such parameters are called hyperparameters.

The main goal when selecting hyperparameters is to obtain a model that generalizes the training data and can then be applied to unseen examples. In order to evaluate the capability of the model to generalize, a validation data set can be used. This data set and the training data set have to be disjoint. To finally evaluate the performance of a model, a test data set is used which again contains different data points than the training and validation sets.

2.2 Related Work

Recent years have shown that very often models based on learned distributed representations outperform conventional models, which are partially based on rules.[1] This can be observed in many areas of artificial intelligence like object recognition, speech recognition, machine translation or sentiment analysis [BCV13]. As the context path model also uses learned distributed representations, I give an introduction to a common form of representation learning for knowledge graphs in NLP.

Due to its graph structure, it is obvious to represent a knowledge graph by the entities and relations it comprises. More concretely, an entity e and a relation r can be represented by vectors $\boldsymbol{e} \in \mathbb{R}^d$ and $\boldsymbol{r} \in \mathbb{R}^d$. Vectors in $\mathbb{R}^d$ are commonly used for distributed representations. A vector representation is called distributed when the characteristics of an entity (or a relation) are represented by patterns over several elements of the vector, while at the same time an element of the vector is involved in representing several entities (or relations) [HMR86].

In order to learn meaningful distributed representations, an objective has to be defined, which measures how well the intended meaning is captured by the representations. Formally, I denote the set of all representations for entities and relations as Θ. Then the objective can be defined as a function $J_G : \Theta \rightarrow \mathbb{R}$ for a given knowledge graph G. Representations which accurately capture their intended meaning are assigned small values by J_G [BCV13].

I now have a look at how an objective function for knowledge graph representations can be defined. In order to do that, I first consider the structural properties of knowledge graphs.

In knowledge graphs, relationships between concrete entities like persons or places are modeled as labeled edges. Abstract concepts like the property of being human are modeled in the same way, by defining a new entity for the property human. This means that all information of a knowledge base is captured in the graph structure, whereas isolated entities

[1] I discuss a rule-based KBC approach by Galárraga et al. [GTHS13] later in this section.

or relations comprise no information about their semantics. In other words, the semantics of an entity is completely defined by its relationships with other entities and the semantics of a relation is defined by the entities it connects. Therefore, meaningful representations of entities and relations need to be based on their context. This context is not symmetric because the edges of a knowledge graph are directed. More concretely, a labeled directed edge (e_1, r, e_2) can be seen as mapping e_1 to e_2 by applying the relation r. This suggests an objective function that uses $\boldsymbol{r}$ to transform $\boldsymbol{e_1}$ into a representation that is similar to $\boldsymbol{e_2}$.

Below I give an example for an objective function J_G that implements this idea by using translations in vector space. This objective originates from a KBC model called TransE proposed by Bordes et al. [BUG+13].

$$J_G(\Theta) = \sum_{(e_1, r, e_2) \in G} \sum_{(e_1', r, e_2') \in G'} \left[1 + \|\boldsymbol{e_1} + \boldsymbol{r} - \boldsymbol{e_2}\|_2^2 - \|\boldsymbol{e_1'} + \boldsymbol{r} - \boldsymbol{e_2'}\|_2^2\right]_+ . \tag{2.2}$$

where $[x]_+$ denotes the positive part of x and

$$\begin{aligned} G' &= (G_1' \cup G_2') \setminus G \\ G_1' &= \bigcup_{r \in R} \{(e_1', r, e_2) \mid \exists e \in E : (e_1', r, e) \in G \wedge \exists e_1 \in E : (e_1, r, e_2) \in G\} \\ G_2' &= \bigcup_{r \in R} \{(e_1, r, e_2') \mid \exists e \in E : (e, r, e_2') \in G \wedge \exists e_2 \in E : (e_1, r, e_2) \in G\}. \end{aligned} \tag{2.3}$$

R is defined as the set of all relations and E is defined as the set of all entities in the knowledge graph G.

Intuitively, this objective function reduces the distance between $\boldsymbol{e_1} + \boldsymbol{r}$ and $\boldsymbol{e_2}$ if (e_1, r, e_2) is a correct triple and increases it otherwise. The objective function aims at achieving a margin of at least 1 between distances for correct and distances for corrupted triples. Corrupted triples are generated by replacing either e_1 or e_2 of a given triple such that the resulting triple is not part of the given knowledge graph. By the definition of G', the replacing entity matches the type of its position. This encourages J_G to produce representations that can capture more fine grained semantics of entities and relations.

In the example given in Chapter 1, the country of birth can be found by following a path which first leads to the city of birth and then to the country this city is located in. The two relations *city of birth* and *contained by* could be merged into one relation *country of birth*. The path (e_1, *city of birth*, *contained by*, e_2) would then have the same semantics as the triple (e_1, *country of birth*, e_2). This example motivates the idea of representing a sequence of relations $(r_1, ..., r_k)$ of a path $(e_1, r_1, ..., r_k, e_2)$ in the same vector space as relations by composing the relation representations $\boldsymbol{r_1}, ..., \boldsymbol{r_k}$ into one vector $\boldsymbol{r_p}$. This extends the definition of TransE from edges to paths.

Guu et al. [GML15] show how this idea can be applied to the objective J_G from Equation 2.2. Instead of directly using the edges of the knowledge graph G, they perform random walks of length k on G in order to generate paths. Let P be a set of paths of the form $(e_1, r_1, ..., r_k, e_2)$, then the objective J_G can be generalized to paths by defining:

$$J_P(\Theta) = \sum_{p \in P} \sum_{p' \in P'} \left[1 + \|\boldsymbol{e_1} + \boldsymbol{r_p} - \boldsymbol{e_2}\|_2^2 - \|\boldsymbol{e'_1} + \boldsymbol{r_p} - \boldsymbol{e'_2}\|_2^2\right]_+ \tag{2.4}$$

where the representation for the composition of the relations $r_1, ..., r_k$ is defined as:

$$\boldsymbol{r_p} = \boldsymbol{r_1} + ... + \boldsymbol{r_k}. \tag{2.5}$$

P' is the set of paths that have either e_1 or e_2 corrupted.

$$\begin{aligned} P' &= \bigcup_{p \in P} (\mathcal{N}_1(p) \cup \mathcal{N}_2(p)) \\ \mathcal{N}_1(p) &= \{(e_1', r_1, ..., r_k, e_2) \mid \exists e \in E : (e_1', r, e) \in G \wedge e_1' \notin C_1(r_1, ..., r_k, e_2)\} \\ \mathcal{N}_2(p) &= \{(e_1, r_1, ..., r_k, e_2') \mid \exists e \in E : (e, r, e_2') \in G \wedge e_2' \notin C_2(e_1, r_1, ..., r_k)\}. \end{aligned} \tag{2.6}$$

To ensure that these corrupted paths actually do not occur in the knowledge graph, the sets of correct path endings C_1 for the left entity and C_2 for the right entity have to be defined:

$$\begin{aligned} C_1(r_1, ..., r_k, e_2) &= \begin{cases} \{e_1 \mid (e_1, r_1, e_2) \in G\} & \text{if } k = 1, \\ \{e_1 \mid \exists e \in C_1(r_2, ..., r_k, e_2) : (e_1, r_1, e) \in G\} & \text{if } k > 1. \end{cases} \\ C_2(e_1, r_1, ..., r_k) &= \begin{cases} \{e_2 \mid (e_1, r_k, e_2) \in G\} & \text{if } k = 1, \\ \{e_2 \mid \exists e \in C_2(e_1, r_1, ..., r_{k-1}) : (e, r_k, e_2) \in G\} & \text{if } k > 1. \end{cases} \end{aligned} \tag{2.7}$$

This definition encourages that representations are learned which give $\boldsymbol{e_1} + \boldsymbol{r_1} + ... + \boldsymbol{r_k}$ the semantics of the set of entities $C_2(e_1, r_1, ..., r_k)$ that are reached when traversing the knowledge graph over the edges $r_1, ..., r_k$, starting from e_1. Calculating the distance $\|\boldsymbol{e_1} + \boldsymbol{r_1} + ... + \boldsymbol{r_k} - \boldsymbol{e_2}\|_2^2$ can then be interpreted as a continuous generalization of the membership test $e_2 \stackrel{?}{\in} C_2(e_1, r_1, ..., r_k)$ in the sense that for a small distance it is more likely that $e_2 \in C_2(e_1, r_1, ..., r_k)$ holds than for a large distance.

This results in the following instantiation of the function $score(p)$, which I abstractly describe in Section 2.1 for triples that are now generalized to paths $p = (e_1, r_1, ..., r_k, e_2)$.

$$score_{TransE}(p) = \|\boldsymbol{e_1} + \boldsymbol{r_1} + ... + \boldsymbol{r_k} - \boldsymbol{e_2}\|_2^2 \tag{2.8}$$

Vectors are not the only possibility for representing relations. The Bilinear model from Nickel et al. [NTK11] uses matrices $\boldsymbol{W_r} \in \mathbb{R}^{d \times d}$ to represent relations. I directly show the generalization of the Bilinear model to estimating the correctness of paths by $score(p)$:

$$score_{Bilinear}(p) = \boldsymbol{e_1}^\intercal \boldsymbol{W_{r_1}} ... \boldsymbol{W_{r_k}} \boldsymbol{e_2} \tag{2.9}$$

Guu et al. give an intuitive motivation for multiplying matrices that represent relations in order to model paths by interpreting them as low-dimensional adjacency matrices for relations. A vector for the entity e can be seen as low-dimensional representation of an indicator vector that has a 1 in the entry corresponding to entity e and is otherwise 0. Using this intuition, $score_{Bilinear}(p)$ can be interpreted as counting the number of paths that connect e_1 and e_2 by traversing the relations $r_1, ..., r_k$. Consequently, high values of $score_{Bilinear}(p)$ represent that it is more likely that p is a correct path. Except for this difference in sign, an objective for $score_{Bilinear}(p)$ can be defined similarly to the objective for TransE.

Restricting the matrices $\boldsymbol{W_r}$ to be diagonal results in the Bilinear-diag model proposed by Yang et al. [YYH+14]. As a special case of the Bilinear model, it also supports estimating the correctness of paths. Bilinear-diag can also be seen as a variant of TransE which uses multiplication instead of addition to let entity and relation representations interact.

Due to matrix multiplication being non-commutative, the Bilinear model can capture the order of relations in a path, which is not the case for TransE and Bilinear-diag.

While the authors of those models report improvements over the state of the art KBC performance, such results are highly dependent on the used data sets. This can be seen in the experiments from Guu et al., who compare the three mentioned models on two data sets. Kadlec et al. [KBK17] show that the choice of hyperparameters has a very strong impact on the performance of KBC models and that even a simple model like Bilinear-diag can outperform a wide range of KBC models when good hyperparameters are chosen.

The main contribution of this work, generating explanations for predictions, is similar to the task of finding logical rules in a knowledge base. In the literature these rules are often formalized as horn rules. With a premise of length two and the conclusion (e_1, r_3, e_3), a horn rule can have the following form in the context of knowledge bases:

$$(e_1, r_1, e_2) \wedge (e_2, r_2, e_3) \rightarrow (e_1, r_3, e_3) \tag{2.10}$$

Galárraga et al. [GTHS13] propose the system AMIE which mines such rules. Their approach is to adapt association rule mining to incomplete knowledge bases by defining new measures for support and confidence. 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 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 mentioned KBC models that use a function of the form $score(e_1, r, e_2)$ can take arbitrary triples as input, provided that the involved entities and relations occur somewhere in the training set.

The rules found by Galárraga et al. always have a positive conclusion and therefore cannot provide evidence for refuting triples.

Yang et al. [YYH+14] first learn representations for relations with common KBC models like TransE or Bilinear and then use these representations to mine horn rules. More concretely, they use the similarity of the representations of the premise and the conclusion to prune the search space of potential rules. The representation for the premise is obtained similar to Guu et al. as the composition of relations that occur in the premise. The conclusion is represented by the relation it contains (r_3 in 2.10). Despite these rules are extracted with the help of learned relation representations, they do not explain how predictions are actually derived because models like TransE or Bilinear do not explicitly use these rules.

Association rules in general are found by counting how often they apply in a given knowledge base. This allows for assigning comprehensible confidence scores to rules. These scores reveal which rules are strictly satisfied in the knowledge base and which rules provide less certainty for predictions. On the other hand, these rules cannot capture the characteristics of individual entities.

PTransE, proposed by Lin et al. [LLL+15] uses paths surrounding a triple and assigns them relevance scores that determine the reliability of paths for estimating the correctness of the given triple. The relevance scores in PTransE are not learned parameters. They instead use a heuristic called path-constraint resource allocation, that is based on the sizes of entity sets (similarly defined as C_1 and C_2 in 2.7) that can be reached by following the relations in a path step by step. Lin et al. 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.

3 Context Path Model

The context path model is based on the KBC formalization, namely to estimate the correctness of a triple $t = (e_1, r, e_2)$, as introduced in Chapter 1. The CPM uses paths that capture the context of t to estimate the correctness of t and to provide explanations for its predictions. I denote the set of paths that are used to capture the context of a triple t in the knowledge graph as P_t or as *context paths*. Those paths do not necessarily have to occur in the knowledge graph and can be potentially incorrect. In Section 3.1, I motivate the CPM by giving examples how both correct and incorrect context paths can be used to assess the correctness of triples. I then define the context path model in Section 3.2 and show how it uses such paths to collect evidence for or against the correctness of triples. In Section 3.3, I explain how context paths can be used for explaining predictions and in Section 3.4, I describe how I generate context paths.

3.1 Motivation

Especially when dealing with real world knowledge, which comprises connections with different degrees of certainty, probabilities are a suitable tool for describing connections between triples and paths. For a triple t, I define the binary random variable C_t whose value is 1 iff. t is a correct fact. Then the probability of t being a correct fact can be denoted as $\mathrm{P}(C_t = 1)$ and the probability of t being incorrect as $\mathrm{P}(C_t = 0) = 1 - \mathrm{P}(C_t = 1)$. In the same way, the probability of a path p being correct can be defined. With $\mathrm{P}(C_t = C_p)$, I denote the probability of t and p being both correct or both incorrect. I use the term *correctness* when referring to the gold standard. In contrast to knowledge bases, the gold standard misses no facts.

The following examples illustrate how paths can be used to predict new facts and to provide explanations for the predictions.

1. I again consider the example from Chapter 1 with the triple $t_1 = (e_1, \textit{country of birth}, e_2)$ and the path $p_1 = (e_1, \textit{city of birth}, \textit{contained by}, e_2)$. The correctness of p_1 and t_1 are logically equivalent. In terms of probabilities this means that $\mathrm{P}(C_t = C_p) = 1$ holds. The path p_1 can be used as evidence for showing that t_1 is correct and for showing its incorrectness.

2. The path $p_2 = (e_1, \textit{nationality}, e_2)$ provides some evidence for the correctness of t_1 though it is not logically equivalent to t_1. The correctness of p_2 only makes it more likely that t_1 is correct. Likewise, p_2 being incorrect makes it more likely that t_1 is incorrect as well. Therefore $\mathrm{P}(C_{t_1} = C_{p_2})$ should still be high but smaller than one. The path p_2 can be used as evidence when assessing the correctness of t but it should have less influence on the prediction than p_1 from the first example.

3. There are also paths that provide no or almost no hints about the correctness of a triple. An example for that is the path $p_3 = (e_1, \textit{lived in country}, \textit{neighboring country}, e_2)$. This path has a weak connection with t_1 in the sense that it is not unlikely to have lived in a country that adjoins the country of birth. As countries very often have several neighboring countries, t_1 can hardly be used as evidence for or against the correctness of t_1. I expect the probability $\mathrm{P}(C_{t_1} = C_{p_3})$ to be small.

4. There are cases where the connection between a triple and a corresponding path is not symmetric like in the examples above but rather describes a necessary or sufficient condition. As an example, I consider the triple $t_2 = (e_1, \textit{city of birth}, e_2)$ and the path $p_3 = (e_1, \textit{country of birth}, \textit{contains}, e_2)$. This path is a necessary condition for t_2, although it is not sufficient for t_2 because the person e_1 might be born in a city different from e_2 but in the same country. The probability $\mathrm{P}(C_{t_2} = C_{p_3})$ could be around 0.5. The path p_3 cannot be used as evidence for the correctness of t_2. Instead, it can be used to collect evidence against the correctness of t_2 if p_3 is not correct. Analogously, a path p that is a sufficient condition for a triple t can only be used to collect evidence for the correctness of t but not against it. In this case, $\mathrm{P}(C_t = C_p)$ can take the same values as $\mathrm{P}(C_{t_2} = C_{p_3})$.

I only consider positive connections between p and t in the sense that correctness of p can only make the correctness of t more likely but p cannot provide evidence for t not being correct. Analogously, p being incorrect can only provide evidence against the correctness of t. For example the logical connection that a person being female implies she is not the father of any other person cannot be captured in this form. Although, this connection can also be formulated as a positive connection that it is a necessary condition for being a father to be male.

Ideally, an explainable KBC model should be able to determine the reliability of paths as indicator for the correctness of a triple t. This means that the model has to differentiate between equivalence, necessary conditions and sufficient conditions and between varying degrees of certainty in order to appropriately use paths for reasoning and for explanations in the way described above.

The idea of the context path model is to use $\mathrm{P}(C_t = C_p)$ to estimate the reliability with which paths can give evidence for or against the correctness of triples. As shown in the last example, $\mathrm{P}(C_t = C_p)$ does not differentiate sufficient and necessary conditions. Therefore, it is not a perfect indicator for the reliability of paths. Thus, I use the term *relevance* to describe the role of $\mathrm{P}(C_t = C_p)$ in the CPM more generally as an indicator for paths that provide evidence for or against the correctness of t.

3.2 Definition of the Context Path Model

Given the context paths P_t for a triple t, the context path model estimates the correctness of t given its context paths. I denote this estimation with $score(t, P_t)$ and define it as follows:

$$score(t, P_t) = \frac{1}{Z(t, P_t)} \sum_{p \in P_t} r(t, p) \cdot score(p), \tag{3.1}$$

$$Z(t, P_t) = \sum_{p \in P_t} r(t, p). \tag{3.2}$$

The probability of a path p being correct $P(C_p = 1)$ is estimated with $score(p)$. The term $r(t, p)$ estimates the probability $\mathrm{P}(C_t = C_p)$ or the relevance of p for t. I also denote $r(t, p)$ with *relevance score*. I call $score(p)$ the *correctness score of p* and $score(t, P_t)$ the *context score of t*. Like its probabilistic counterpart $P(C_p)$, the estimation $score(p)$ is restricted to lie in $[0, 1]$. I only require $r(t, p)$ to be non-negative. Despite estimating $\mathrm{P}(C_t = C_p)$, it is not necessary for $r(t, p)$ to lie in $[0, 1]$ because normalization is already applied with $Z(t, P_t)$. The property of $score(p)$ being normalized carries over to $score(t, P_t)$ which also lies in $[0, 1]$.

On a technical level, this model has the following properties: Paths with high correctness scores push $score(t, P_t)$ towards 1 by increasing the nominator and the denominator by the same amount. Paths with low correctness scores push $score(t, P_t)$ towards 0 by only increasing the normalization term in the denominator. The magnitude of both effects depends on the respective relevance scores.

Applied to the first example from Section 3.1, the path $p_1 = (e_1, \textit{city of birth}, \textit{contained by}, e_2)$ should be assigned a high relevance score $r(t_1, p_1)$ when it is used as context path for the triple $t_1 = (e_1, \textit{country of birth}, e_2)$. If p_1 is correct, $score(p_1)$ should be close to 1 since it estimates $P(C_{t_1} = 1)$. A high relevance score combined with a high correctness scores pushes $score(t_1, P_{t_1})$ towards 1. If p_1 is not correct, $score(p_1)$ should be close to 0. In this case a high relevance score is combined with a low correctness score which pushes $score(t_1, P_{t_1})$ towards 0. Both effects match the intended meaning of $score(t_1, P_{t_1})$ to represent the correctness of t_1.

The third example from Section 3.1 features the path $p_3 = (e_1, \textit{lived in country}, \textit{neighboring country}, e_2)$, which has a low relevance for t_1 and should be assigned a low relevance score $r(t_1, p_3)$. Since correctness scores are restricted to lie in $[0, 1]$, the effect of $score(p_3)$ on $score(t_1, P_{t_1})$ is small, independently of $score(p_3)$ being high or low. This properly models that the correctness of p_3 provides no evidence for the correctness of t_1.

3.2.1 Estimating the correctness of context paths

A model for $score(p)$ must have the following properties: It needs to be able to model paths and its output has to lie in $[0, 1]$. The first property applies to all composable KBC models like TransE [BUG+13], Bilinear [NTK11] and Bilinear-diag [YYH+14]. I show in the following how the output of distance-based models can be mapped to $[0, 1]$ to make them suitable for the CPM.

I choose the TransE Model [BUG+13] as basis for assessing the correctness of context paths. Bordes et al. use the distance $\|\boldsymbol{e_1} + \boldsymbol{r} - \boldsymbol{e_2}\|_2^2$ directly as a measure for the correctness of a triple (e_1, r, e_2). This is possible due to the max-margin objective (Equation 2.2) which encourages correct triples to be assigned a distance that is at least by the margin 1 smaller than distances of incorrect triples. While distances can take any non-negative values, the context path model expects the correctness scores to lie in $[0, 1]$. I use the logistic sigmoid function σ to map the TransE scores to the interval $[0, 1]$ and add a path-specific bias $\boldsymbol{b_1^\intercal r_p}$ ($\boldsymbol{b_1} \in \mathbb{R}^d$) that can adapt varying distance scales between paths to the nonlinearity of the sigmoid. I define the correctness score for a path $p = (e_1, r_1, ..., r_k, e_2)$, using the compositional representation $\boldsymbol{r_p} = \boldsymbol{r_1} + ... + \boldsymbol{r_k}$, as follows:

$$score(p) = \sigma(-\|\boldsymbol{e_1} + \boldsymbol{r_p} - \boldsymbol{e_2}\|_2^2 + \boldsymbol{b_1^\intercal r_p}) \tag{3.3}$$

with the logistic sigmoid function σ:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{3.4}$$

When $score(p)$ is used in the context path model, it is only calculated for paths that cannot be found in the knowledge graph. Paths p that occur in the training knowledge graph are directly assigned $score(p) = 1$.

3.2.2 Estimating the relevance of context paths

For estimating the relevance of context paths $p = (e_1, r_1, ..., r_k, e_2)$ for a triple t, the CPM represents the path as sequence of relations $r_1, ..., r_k$. By using only these relations, the meaning of the path is abstracted from the actual participating entities e_1 and e_2. This makes it possible to generalize the question of which path is relevant for a triple to finding connections between the relation of a given triple and relation sequences of paths.

Even when considering paths of length 2 or 3, the number of combinations of relations and relation sequences can be high, for paths of length k, it is bounded by $|R|^{k+1}$. Using one parameter per combination to learn $r(t, p)$ would not only significantly increase the number of parameters, but also suffer from sparsity of certain combinations.

I propose a simple model for learning $r(t, p)$, that uses the compositional representation of relation sequences. The model learns a vector $\boldsymbol{c_r} \in \mathbb{R}^d$ for each relation r in order to recognize patterns in the compositional path representation $\boldsymbol{r_p}$ (defined in Equation 2.5) that indicate how relevant the path p is for the relation r. As relevance scores have to be non-negative, the exponential function is applied.

$$r(t, p) = \exp(\boldsymbol{c_r}^\top \boldsymbol{r_p} + \boldsymbol{b_2}^\top \boldsymbol{r}). \tag{3.5}$$

The bias term $\boldsymbol{b_2}^\top \boldsymbol{r}$ with $\boldsymbol{b_2} \in \mathbb{R}^d$ creates additional degrees of freedom to adapt to the nonlinearity of the exponential function and to the relation specific scaling of $\boldsymbol{c_r}^\top \boldsymbol{r_p}$.

In contrast to learning one parameter per combination, this model can use the distributed representations for paths and relations to generalize from common path-relation combinations to more sparse combinations or to combinations that do not appear in the training set. This relevance model only adds $d \cdot (|R| + 1)$ parameters to the $d \cdot |R| + d \cdot |E|$ parameters of TransE.

3.2.3 Training

Since $r(t, p)$ estimates $\mathrm{P}(C_t = C_p)$ and $score(p)$ estimates $\mathrm{P}(C_p = 1)$, they should be learned separately, each with an objective that encourages them to estimate their probabilistic counterpart. Learning both $r(t, p)$ and $score(p)$ jointly would lead to $score(p)$ being

influenced by the relevance of p. Then $score(p)$ would not estimate $P(C_p = 1)$ anymore. Though, it is important that $score(p)$ actually estimates $P(C_p = 1)$ because only then the path correctness scores are interpretable by users when considering paths as explanation for predictions. Consequently, I split the training process into two phases to first learn the parameters of $score(p)$ and then the parameters of $r(t, p)$.

Learning $score(p)$

Following Guu et al., I first train $score(p)$ on the edges of the knowledge graph G_{KB} before training it on longer paths. This allows the model to build up paths from meaningful edges. Capturing the semantics of paths accurately is crucial for generating explanations because this makes it more likely that the semantics captured by the model correspond to what users understand when considering the paths. Instead of generating completely random paths by traversing G, I only sample paths that fulfill the criteria described in Section 3.4.2. This focuses the training on relation sequences that are actually used by the context path model.

TransE is often trained using a max-margin objective [BUG+13; GML15; YYH+14]. However, I use the cross-entropy loss, which fits well to the sigmoid function that outputs scores in [0,1]. These scores can be interpreted as probabilities. The same holds for the correct labels. They take either the value 1 or 0, but are implicitly represented by the sets of correct paths P, consisting of paths with label 1, and the set of incorrect paths P' with label 0. The cross-entropy measures the similarity of the probability distribution generated by $score(p)$ and the probability distribution of the labels. This leads to the objective $J_s(\Theta_s)$ for $score(p)$:

$$J_s(\Theta_s) = \frac{1}{|P|} \sum_{p \in P} -log(score(p)) + \frac{1}{|P'|} \sum_{p' \in P'} -log(1 - score(p')). \tag{3.6}$$

I define $P = G_{KB}$ for the single edge training and $P = \hat{P}$ for the subsequent path training. Informally, $\hat{P}$ is the set of all correct context paths in G_{KB}. Formally, it is defined using the sets $\hat{P}_t$, which I define in Equation 3.13.

$$\hat{P} = \bigcup_{t \in G_{KB}} \hat{P}_t \tag{3.7}$$

P' is defined as in Equation 2.6. J_s is minimized by updating the following parameters:

$$\Theta_s = \{\boldsymbol{r} \mid r \in R\} \cup \{\boldsymbol{e} \mid e \in E\} \cup \{\boldsymbol{b_1}\}. \tag{3.8}$$

This encourages $score(p)$ to represent the semantics of paths. The intuition I give for the max-margin loss of Equation 2.4 directly transfers to the cross-entropy loss $J_s(\Theta_s)$.

Learning $r(t, p)$

The relevance scores for paths are learned by minimizing the following cross-entropy loss $J_r(\Theta_r)$:

$$J_r(\Theta_r) = \frac{1}{|G|} \sum_{t \in G} -log(score(t, P_t)) + \frac{1}{|G'|} \sum_{t' \in G'} -log(1 - score(t', P_{t'})). \tag{3.9}$$

G and G' are defined as in Equation 2.3. Only the parameters for the relevance scores are updated:

$$\Theta_r = \{\boldsymbol{c_r} \mid r \in R\} \cup \{\boldsymbol{b_2}\}. \tag{3.10}$$

This cross-entropy loss aims to assign correct triples a context score of 1 and incorrect triples a context score of 0. At the same time, $score(p)$ is already trained to estimate $\mathrm{P}(C_p = 1)$. Thus, $J_r(\Theta_r)$ encourages $r(t, p)$ to estimate $P(C_t = C_p)$.

3.3 Generating Explanations for Predictions

The context path model is able to produce explanations for its predictions by listing the context paths that have the highest relevance scores for a triple t.

The relevance scores can be normalized to sum up to 1. This does not change the model since it performs the same normalization internally with the term $Z(t, P_t)$ as defined in Equation 3.2. A normalized relevance score represents the percentage to which the correctness score of the corresponding context path contributes to the output $score(t, P_t)$. This means that the paths with the highest relevance scores have the most impact on the prediction.

The cross-entropy loss aims to assign correct triples a context score of 1 and incorrect triples a context score of 0. The model is unsure about the correctness of triples if it outputs 0.5. A context path with a correctness score larger than 0.5 pushes the output towards 1. I denote those paths as *pro paths*, because when they are assigned a high relevance score, their role can be interpreted as collecting evidence for the correctness of t. Analogously, I denote context paths with correctness scores smaller than 0.5 as *con paths*.

In contrast to the approach from Yang et al. [YYH+14], which extracts interpretable rules using the learned representations for relations, the context paths are actually used for the prediction and therefore can better explain how the model actually computed the prediction.

3.4 Generating Context Paths

In large, dense knowledge graphs, the number of paths that can be walked starting from a given entity grows exponentially in the path length. This makes the training computationally very intensive while many of those paths are not informative. In this section, I give several criteria for selecting paths in order to keep the computations tractable by directing them to the most promising paths. Before doing that, I present a modification to the knowledge graph that increases the number of paths even more, but also adds potentially relevant paths.

3.4.1 Inverse relations

I again consider the example $t = (e_1, \textit{country of birth}, e_2)$. In contrast to the previous examples, it might be the case that instead of the relation *contained by*, now the relation *contains* provides the connection between the country of birth and the city of birth. In this case the path $(e_1, \textit{city of birth}, \textit{contained by}, e_2)$ does not exist. To incorporate a path with the same semantic, the relation *contains* has to be inverted. Then the previous path can be expressed as $(e_1, \textit{city of birth}, \textit{contains}^{-1}, e_2)$. More general, let G_{KB} be the knowledge graph as defined in Equation 1.1, then the graph G, which is used for generating context paths, is obtained by adding inverse relations as follows:

$$G = G_{KB} \cup \{(e_2, -r, e_1) \mid (e_1, r, e_2) \in G_{KB}\}. \tag{3.11}$$

3.4.2 Path selection criteria

Closed paths

The first path selection criterion aims for finding paths that can support the correctness of a triple $t = (e_1, r, e_2)$. Naturally, paths that connect the entities of a triple express some connection between them. Restricting paths to start with e_1 and end with e_2 - to be closed paths - makes it more likely that the connection they describe is related to the triple t and might provide evidence for the correctness of t. This can be seen in the previous examples regarding the relation *country of birth*, since all of them satisfy the criterion.

Paths of limited length

The second criterion is the limitation of path lengths to $k \leq 3$. This effectively reduces the number of potential paths, since the number of paths grows exponentially in the path length. Although, there is also an intuitive motivation regarding the potential relevance of long paths. While for example two friends tend to have several similarities like interests, place of residence etc., the set of similarities can significantly decrease when considering the friend of the friend. When applying this observation several times, it is obvious that the connection between two entities over very long paths becomes more and more blurred.

Trivial paths

The third criterion seeks to filter out trivial paths. For the path length $k = 2$, I denote paths as trivial if they contain a relation that is close to the identity relation. An example could be the relation *near by*. The path (e_1, *country of birth*, *near by*, *Switzerland*) might be assigned a high correctness score by the context path model, while the correct fact is (e_1, *country of birth*, *Italy*). It is obvious that nearby countries have similar representations and are harder to distinguish by the model. As I show in Section 3.2.2, the context path model estimates the relevance of a path based on the composition of the relations it contains. Since the composition of *country of birth* and *near by* is still close to *country of birth*, the path might also be assigned a high relevance. In summary, paths of length two that comprise a relation close to the identity, might be seen as relevant by the model while introducing additional noise.

For the path length $k = 2$, I denote paths as trivial if they comprise two successive, mutually inverse relations which mostly cancel out each other. Examples for trivial paths are (e_1, *country of birth*, *contains*, *contains*$^{-1}$, e_2) and (e_1, *country of birth*, *contains*, *contained by*, e_2). They rather add noise than providing evidence for the relation *country of birth*. While the former path can easily recognized as trivial by its structure, the latter is harder to detect without additional information about the semantics of *contains* and *contained by*.

In order to implement a filter that detects trivial paths reliably, I first define the left domain $D_1(r)$ and the right domain $D_2(r)$ of a relation r:

$$\begin{aligned} D_1(r) &= \{e_1 \mid \exists e_2 \in E : (e_1, r, e_2) \in G\}, \\ D_2(r) &= \{e_2 \mid \exists e_1 \in E : (e_1, r, e_2) \in G\}. \end{aligned} \tag{3.12}$$

A path $(e_1, r_1, ..., r_k, e_2)$ is defined as non-trivial if e_1 occurs only in $D_1(r_1)$ and e_2 occurs only in $D_2(r_k)$ and if e_1 and e_2 do not occur in any of the other domains of $r_2, ..., r_{k-1}$.

In trivial paths of length 2, the identity relation either maps e_1 to e_1 or e_2 to e_2. The same holds for two successive, mutually inverse relations in trivial paths of length 3. This property is used when identifying trivial paths by analyzing their relation domains.

The exclusion of trivial paths can be too strict in some cases. An example can be the context path $p = (e_1, \textit{mother of}, \textit{mother of}, e_2)$ for the triple $t = (e_1, \textit{grandmother of}, e_2)$. The path p is excluded if the mother of e_2 herself participates in the relation *grandmother of* either as grandmother or as grandchild because all paths over grandmothers or grandchildren are excluded. Despite that, the remaining paths still capture a wide range of connections.

The paths $\hat{P}_t$, which are found by the methods presented so far, can be summarized formally:

$$\begin{aligned}\hat{P}_t = \bigcup_{k=1}^{3} \{(e_1, r_1, ..., r_k, e_2) \mid e_2 \in C_2(e_1, r_1, ..., r_k) \\ \wedge\, e_1, e_2 \notin D_2(r_1) \cup D_1(r_k) \cup \bigcup_{i=2}^{k-1} (D_1(r_i) \cup D_2(r_i))\}.\end{aligned} \tag{3.13}$$

The first line implements the closed path criterion and the second line excludes trivial paths.

Potentially incorrect paths

The paths described so far can only be used by the CPM to find evidence for the correctness of a triple. All those paths are found by performing walks on the knowledge graph and therefore must be correct. This is because KBC only assumes an incomplete knowledge graph, while existing edges are considered correct. The context path model uses incorrect paths as evidence for incorrect triples. With incorrect paths I denote correctly absent paths.

Typically a knowledge graph has significantly less edges than its corresponding complete graph, a graph in which all entities are pairwise connected by all relations. This means that there are even more potentially incorrect paths than correct paths. Of course, many of those incorrect paths are nonsensical and not relevant for t. I propose a selection criterion for potentially incorrect but relevant paths $\widetilde{P}_t$ that is based on the set of correct paths $\hat{P}_t$. It is formally defined as:

$$\begin{aligned}\widetilde{P}_t = \bigcup_{k=1}^{3} \{(e_1, r_1, ..., r_k, e_2) \mid \exists \hat{e}_1, \hat{e}_2 : \hat{t} = (\hat{e}_1, r, \hat{e}_2) \wedge \hat{t} \in G \wedge (\hat{e}_1, r_1, ..., r_k, \hat{e}_2) \in \hat{P}_{\hat{t}} \\ \wedge\, (C_2(e_1, r_1, ..., r_k) \neq \varnothing \vee C_2(r_1, ..., r_k, e_2) \neq \varnothing)\}.\end{aligned} \tag{3.14}$$

Intuitively, this criterion selects paths that comprise sequences of relations which are already considered potentially relevant by the criteria for $\hat{P}_t$. In other words, the paths in $\widetilde{P}_t$ provide negative examples for the paths in $\hat{P}_t$. This enables the model to find connections between the correctness of paths and the correctness of a given triple. Additionally, paths in $\widetilde{P}_t$ are required to be constructed by actual walks on the knowledge graph. The only potentially incorrect part in those paths is either e_1 or e_2. This assures that paths in $\widetilde{P}_t$ represent the context of t, despite being potentially incorrect.

Applying all criteria together, I define the context paths P_t for a given triple t by:

$$P_t = \hat{P}_t \cup \widetilde{P}_t. \tag{3.15}$$

4 Experiments

For my experiments, I use a data set called FB15K, which is extracted from the Freebase knowledge base by Bordes et al. [BUG+13]. I summarize the background and the main characteristics of FB15K in Section 4.1.

It is common to evaluate the KBC task using ranking metrics. I introduce two ranking metrics that I use in my experiments in Section 4.2.

In Section 4.3, I analyze how meaningful the explanations produced by the CPM are. I consider two aspects that make explanations meaningful. First, the semantics of paths as captured by the model should be close to their semantics as perceived by users. I cover this aspect by evaluating how accurate the model for $score(p)$ predicts the correctness of paths. For this purpose, I use the ranking metrics introduced in Section 4.2. The second aspect of meaningful explanations concerns the relevance scores. The examples in Section 3.1 already provide an intuitive idea that meaningful paths should be reliable as indicator for the correctness of a triple. I specify this aspect more concretely in 4.3.2. This aspect cannot be evaluated automatically because there is no data set available that provides information about the reliability of paths. Thus, I manually annotate a sample of prediction explanations. To make the annotation more comprehensible, I explain the annotation scheme in 4.3.3 by giving annotation examples. The hyperparameters used for the experiments are stated in Section 4.5.

In Section 4.4, I evaluate how accurate the CPM predicts missing facts by applying the ranking metrics of Section 4.2. The baseline in this evaluation setting is the TransE Model. It can be seen as a special case of the CPM when all context paths except for the triple itself are assigned the relevance score 0. I compare the baseline with two variants of the CPM: The first variant uses the context paths P_t as defined in Equation 3.15. In the second variant, I exclude t from P_t in order to examine to which degree the correctness of triples can be predicted just by considering their context. This variant is referred to as exclusive context path model (ECPM).

4.1 Data Set

Freebase [BEP+08] comprises general human knowledge. It has been built collaboratively to support a broad range of topics. In 2010, it was bought by Google.[1] Now it is integrated into the Google Knowledge Graph.[2] The public version of Freebase has been shut down in 2016, but a dump of the data set is still publicly available.[3] The last version of Freebase consists of 1.9 billion triples. Freebase triples have the same form (e_1, r, e_2) as introduced in Chapter 1. A Freebase relation r has the following internal structure: *domain/type/attribute*. Domains provide a coarse classification of entities, for example into *people*, *location* or *music*. Types refine these classes. For the domain *music*, possible types are *artist* or *album*. Entities can be part of several domains and types at the same time. The attribute stands for the relationship between e_1 and e_2 [Cha18]. For example, the triple (e_1, *music/album/genre*, e_2) describes the fact that e_1 is a music album of the genre e_2. There are also facts with composite relations in FB15K. They have the form (e_1, $domain_1/type_1/attribute_1./domain_2/type_2/attribute_2$, e_2). Here, the entity e_1 has $type_1$ and e_2 has $type_2$. The second part of the composite relation specifies the role of e_2 more concretely. An example is the triple (*San Francisco*, */travel/travel_destination/how_to_get_here./travel/transportation/mode_of_transportation*, *Air travel*).

The data set FB15K is the subset of the Freebase triples that fulfill the following constraints: The entities e_1 and e_2 and the relation r of a triple must each have at least 100 mentions in Freebase. Additionally, the entities have to be present in the Wikilinks database.[4] For obviously inverse relations like */people/person/nationality* and */people/person/nationality*$^{-1}$, Bordes et al. filter out one of them. On top of that, I remove 14 relations with the domains *dataworld* and *commons*, because they contain rather technical meta information than real world knowledge. The statistics of the resulting data set and its split into training, validation and test data are given in Table 4.1.

The sizes of samples from $\hat{P}$, which is defined in Equation 3.7, for training and evaluating the path correctness scores are also listed in Table 4.1. For generating the path test sets, the set P_t is computed for test triples t. From this set, only those paths are used that comprise

[1] https://googleblog.blogspot.com/2010/07/deeper-understanding-with-metaweb.html
[2] https://blog.google/products/search/introducing-knowledge-graph-things-not/
[3] https://groups.google.com/d/msg/freebase-discuss/WEnyO8f7xOQ/RY1gjmkNBQAJ
[4] https://code.google.com/archive/p/wiki-links/downloads

		Size
Entities		14,916
Relations		1,331
Triples	Train	470,972
	Valid	48,804
	Test	57,613
Paths of length 2	Train	3,426,314
	Test	82,641
Paths of length 3	Train	3,602,001
	Test	202,235
Context paths per triple (avg)	length 1	3.4
	length 2	20.7
	length 3	430.3

Table 4.1: Statistics of the used data sets. The number of context paths in P_t is averaged over triples in the training set. The triple t itself is also counted as context path.

at least one edge of the test set. This ensures that only paths are tested that do not occur in the path training set. Additionally, Table 4.1 lists the average number of context paths per triple.

4.2 Ranking Metrics for KBC

In the following, I explain how correctness scores for paths p can be evaluated. Since paths of length 1 are the same as triples, the explanations can be directly transferred to them.

It is common to evaluate the KBC task using ranking metrics. [BUG+13; GML15; LLL+15; YYH+14]. This can be done by listing a correct path together with all corresponding paths that have one entity corrupted and sorting them in descending order by their correctness scores. One reason for using ranking metrics is that the KBC task is asymmetric in the sense that there are many negative samples for one correct path. Finding few positive examples in a large set of negative examples is a typical setting for ranking problems. Furthermore, many common KBC models do not provide correctness scores that are normalized to lie in $[0, 1]$. In these cases, the prediction for a single path cannot be directly used to classify it as correct or incorrect. Despite this does not apply to the context path model, I stick with the ranking evaluation to make the results more comparable.

I use the same ranking metrics as Guu et al., *hits at 10* and *mean quantile*. For both metrics, I use $\mathcal{R}_p = \{p\} \cup \mathcal{N}_1(p)$ or $\mathcal{R}_p = \{p\} \cup \mathcal{N}_2(p)$ as the set of paths that are ranked when evaluating the predictions for e_1 or e_2 for a test path p. The sets of corrupted paths $\mathcal{N}_1(p)$ and $\mathcal{N}_2(p)$ are defined in Equation 2.6. When generating the sets $\mathcal{N}_1(p)$ and $\mathcal{N}_2(p)$, all triples from the training, validation and test data sets are used. This prevents correct paths of the training or validation set from ending up in the set of corrupted test paths.

The ranking metric hits at 10 is defined as the percentage of correct paths that are ranked within the top 10 paths of their respective sets $\mathcal{R}_p$.

Mean quantile computes the average fraction of incorrect triples that are ranked after the correct triple, or formally:

$$mean\,quantile = \frac{1}{|P|} \sum_{p \in P} \frac{|\{p' \in \mathcal{N}(p) \mid score(p') < score(p)\}|}{|\mathcal{N}(p)|} \tag{4.1}$$

where $\mathcal{N}(p)$ can be either the set of paths $\mathcal{N}_1(p)$ with e_1 corrupted or $\mathcal{N}_2(p)$ with e_2 corrupted. Mean quantile accounts for varying sizes of $\mathcal{N}(p)$. This is not the case for hits at 10, which tends to output higher values for smaller sizes of P'. For $|\mathcal{N}(p)| \leq 9$, the hits at 10 metric always outputs 1, independently from the prediction. I average the results for predicting e_1 and e_2.

4.3 Evaluation of Explanations

4.3.1 Prediction of context paths

When displaying paths as explanation to a user, it is important that the semantics of paths are captured correctly by the model. I evaluate the correctness scores for paths using ranking metrics. In Table 4.2, I compare the ranking evaluation of the TransE model for single edge training and for path training by calculating the relative improvement of path training over single edge training.

Similarly too Guu et al., I observe that training on paths improves the prediction of paths significantly. On paths of length 3, the model trained on paths improves over the single edge model by 10.4 % in hits at 10 and by 4.0 % in mean quantile. For paths of length 2 the improvements are smaller but still significant. The performance on predicting edges suffers from path training. This is in contrast to what Guu et al. report. They show that path

Test path	Single edge training		Path training		%path-improvement	
length	Hits at 10	MQ	Hits at 10	MQ	Hits at 10	MQ
1	**92.8**	**97.7**	90.5	97.6	- 2.5	- 0.1
2	86.2	95.7	**91.7**	**98.3**	6.4	2.7
3	74.2	91.8	**81.9**	**95.5**	10.4	4.0

Table 4.2: Path ranking evaluation different path lengths using models trained on edges and on paths. %path-improvement is the relative improvement of the path trained model compared to the model trained on single edges.

training improves the performance on edges as well. This difference may be specific to the used data sets.[5] Despite the inferior performance on single edges, I use the path trained model for the CPM to make explanations that comprise longer paths more meaningful.

Interestingly, the path trained model performs slightly better on paths of length 2 than on edges. Note that due to the context path selection criteria, the size of the test set for paths of length 2 is considerably smaller than that for paths of length 3. This makes the results for paths of length 2 less resilient. The performance of the path trained model drops by only 2 % when predicting on paths of length 3 compared to the single edge prediction. This suggests that one can trust correctness scores for paths of different lengths to a similar degree when considering the explanations of the CPM.

Keep in mind that the model for $score(p)$ is only used in the CPM when paths are not present in the training knowledge graph. Paths that occur in the training knowledge graph are directly assigned a correctness score of 1 and can be trusted under the assumption that the knowledge graph contains only correct edges.

4.3.2 Reliability of context paths

As already stated in Section 3.1, the relevance of paths does not differentiate between necessary and sufficient conditions. It only measures the probability that the correctness of a path and a triple matches. Despite that, I evaluate the CPM using the notion of reliability, which makes the distinction between necessary and sufficient conditions. This gives a more detailed view on how a model with the limited expressiveness of relevance treats sufficient and necessary conditions.

[5] A test on the Freebase subset used by Guu et al. yields results consistent to what they report.

I now explain more concretely what makes a path a reliable indicator depending on whether it is used as evidence for the correctness of t or against the correctness of t.

1. **Paths as evidence for the correctness of t:** In the strict logical sense, a path is reliable for showing that a triple t is correct if it is logically equivalent to t or if it is a sufficient condition for t. The examples in Section 3.1 show that many connections in the real world cannot be captured by plain logical rules. Therefore I give a probabilistic formulation of the same criterion: A path has a high reliability for t if its correctness makes the correctness of t significantly more likely. It is hard to define and to quantify objectively what *significant* means when considering real world facts. When explaining the annotations scheme, I give several examples in order to give an intuition for my understanding of significance.
2. **Paths as evidence against the correctness of t:** In the logical sense, a path is reliable for showing that a triple t is incorrect if it is logically equivalent to t or if it is a necessary condition for t. In the probabilistic formulation, a path has a high reliability for t if its incorrectness makes it significantly more likely that t is incorrect.
3. The paths that do not fulfill any of those criteria have low or no reliability.

4.3.3 Annotation scheme

The formulations of reliability directly lead to the annotation scheme presented in Figure 4.1.

Paths are classified in **0 a)** if they fulfill the logical formulations of point 1 and point 2 in the section above. Paths that are not in **0 a)** but fulfill the logical formulation of point 1 are classified as **1 a)**. In the same way are paths that are not in **0 a)** but fulfill the logical formulation of point 2 classified as **2 a)**.

The classes with **b)** are the probabilistic counterparts of the classes with **a)**:

Paths are classified in **0 b)** if they are not in **0 a)** and if they fulfill both probabilistic formulations of point 1 and 2. I denote those paths as *almost equivalent* to t. Paths are classified in **1 b)** if they are not in **0 b)** or **1 a)** and if they fulfill the probabilistic formulations of point 1. I denote those paths with *almost sufficient*. Paths are in class **2 b)** if they are not in **0 b)** or **2 a)** and if they fulfill the probabilistic formulation of point 2. I call those paths *almost necessary*.

The point 3 of the previous section directly corresponds to class **3)**.

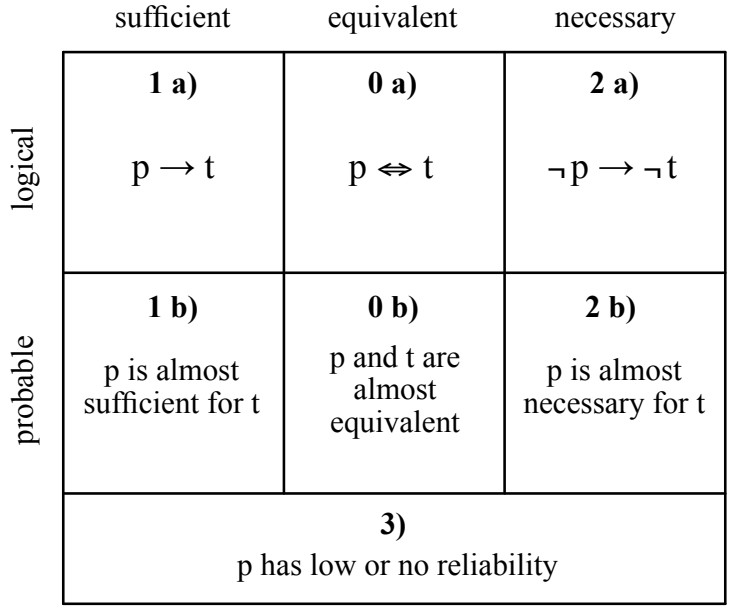

Figure 4.1: Annotation scheme for the reliability of paths.

4.3.4 Annotation examples

In order to make the annotation more comprehensible, I give examples for each annotation class. As these examples are actually generated by the CPM, they also give an impression how meaningful and understandable explanations based on paths can be.

Unlike my previous notation for paths as $(e_1, r_1, ..., r_k, e_2)$, I now also insert the entities a path contains or state it if there are no entities in the training knowledge base that connect the relations of the path.

0 a) p and t are equivalent

Common examples for this class are paths of length 1 that consist of a redundant Freebase relation:
Correct triple t:
(***Jon Favreau filmography***, */people/person/profession*, ***Film director***)
Context path p:
(***Jon Favreau filmography***, */people/profession/people_with_this_profession*$^{-1}$, ***Film director***)
I also classify the following, more complex example as equivalence:
Correct triple t:
(***Football***, */sports/sport/leagues*, ***Confederation of African Football***)

Context path p:
(***Football***, */sports/sport/teams*, ***Zimbabwe national football team***,
/sports/sports_league/teams./sports/sports_league_participation/team^{-1},
Confederation of African Football)

A league is a football league if it comprises football teams and if a team plays in a football league, it is a football team.

0 b) p and t are almost equivalent

Correct triple t:
(***Naval Postgraduate School***, */location/location/containedby*, ***United States of America***)
Context path p:
(***Naval Postgraduate School***, */organization/organization/headquarters./location/mailing_address/state_province_region*, ***California***, *government/political_district/representatives./government/government_position_held/office_holder* ***Richard Nixon***, */people/person/nationality*,
United States of America)

When a school is located in a region that has a (political) representative who has nationality of a certain state, it is significantly more likely that the school is located in this state. There is no guarantee that this holds since not all political positions require the nationality of the respective state. In the other direction this holds as well: Knowing that a school is located in a certain state makes it very likely that the region of the school is represented by a person having the nationality of the state.

1 a) p is sufficient for t

Correct triple:
(***Alto saxophone***, */music/instrument/instrumentalists*, ***John Coltrane***)
Context path p:
(***Alto saxophone***, */music/group_member/instruments_played*$^{-1}$, ***John Coltrane***)

This class very rarely occurs in the annotation set and is often close to the equivalence class. In this example it is sufficient to play an instrument in a group to be called an instrumentalist of this instrument. It is not an equivalence since there are instruments that are not necessarily played in a group.

1 b) p is almost sufficient for t

Correct triple:
(***Feroz Khan***, */people/person/languages*, ***Hindi Language***)
Context path p:
(***Feroz Khan***, */people/person/nationality*, ***India***
/language/human_language/countries_spoken_in^{-1}, ***Hindi Language***)

It is very likely to speak the language that is spoken in the state of the own nationality. It is not a guarantee because states might have several official languages. For example, the English language shows that p is not necessary for t.

2 a) p is necessary for t

Triple t with e_1 corrupted:
(***New York Film Critics Circle Award for Best Actor***,
/award/award_category/nominees./award/award_nomination/award_nominee, ***Miranda Richardson***)
Context path p:
(***New York Film Critics Circle Award for Best Actor***,
/award/award_category/nominees./award/award_nomination/nominated_for,
{Cast Away, There Will Be Blood}, */film/film_job/films_with_this_crew_job./film/film_crew_gig/film*$^{-1}$,
Miranda Richardson)

It is necessary to have a job in the film business to be nominated for a film award.

2 b) p is almost necessary for t

Triple t with e_2 corrupted:
(***Tsui Hark***, */people/person/place_of_birth*, ***Exeter***)
Context path p:
(***Tsui Hark***, */people/person/places_lived./people/place_lived/location*, ***Hong Kong***
is not the same as ***Exeter***)

While having lived in a certain location does not imply to be born there, the other direction holds very likely. Examples where one might be born at a place without having lived there might be families living in smaller towns without a hospital. Here, the place of birth could be a larger city with a hospital in the region.

3) p has low or no reliability

Triple with e_2 corrupted:
(***Feroz Khan***, */people/person/languages*, ***Japanese Language***)
Context path p:
(***Feroz Khan***, */people/person/places_lived./people/place_lived/location*, ***Mumbai***
has no connection over */location/location/adjoin_s./location/adjoining_relationship/adjoins*$^{-1}$
to ***Asia*** */language/human_language/region*$^{-1}$, ***Japanese Language***)

In contrast to the last example, this context path is annotated as not reliable, because there are too many adjoint states with different languages.

This is another example for a not reliable context path:
Triple with e_2 corrupted:
(***Judy Davis***, */people/person/place_of_birth*, ***Nottingham***)
Context path p:
(***Judy Davis***, */education/educational_institution/students_graduates./education/education/student*$^{-1}$,
National Institute of Dramatic Art is not the same as ***University of Nottingham***
/organization/organization/headquarters./location/mailing_address/citytown, ***Nottingham***)

It is not uncommon to graduate at an educational institution in the same place where one is born. Despite that, the educational institution still provides not enough evidence for the place of birth, especially for larger institutions like universities where many students are even international.

Remarks

When the relevance scores of the CPM do not correspond to the annotation, it does not necessarily mean that the CPM is not capable of capturing the respective logical or probabilistic connection. It might be also due to an unequal distribution of facts in Freebase which does not accurately represent the real world. Having mostly American actors in the knowledge base could for example lead to false conclusions like being an actor implies the American nationality. In the example for 1 a), I do not consider whether solo instruments actually occur in Freebase. If Freebase only contains instruments that are typically played in groups, this example could be annotated as equivalence.

4.3.5 Path reliability evaluation metric

Unlike the annotation, the relevance scores of the CPM are not discrete but continuous. This should be taken into account when analyzing the annotation results. I now consider a context path that is annotated as not relevant. If it receives a high relevance score, it should be penalized more by the evaluation metric than if it receives a low relevance score by the CPM. This can be transferred to the other annotation classes as well. As described in Section 4.3.3, the relevance scores of paths should depend on whether they act as pro or as con paths. I only define the evaluation metric for pro paths since the definitions translate directly to con paths.

I denote the set of context paths classified as pro paths for a triple t as $P_{t,pro}$ and the subset of paths in $P_{t,pro}$ that are annotated with class a as $P_{t,pro,a}$.

Furthermore, I denote the fraction of relevance scores associated with a subset of context paths P_1 in the set of context paths P_2 as relevance fraction of P_1 in P_2, or formally as $RF_t(P_1, P_2)$:

$$RF_t(P_1, P_2) = \frac{\sum_{p_1 \in P_1} r(t, p_1)}{\sum_{p_2 \in P_2} r(t, p_2)} \tag{4.2}$$

Now, I can define the relevance fraction of paths annotated with a in pro paths as $RF_t(P_{t,pro,a}, P_{t,pro})$.

Simply averaging $RF_t(P_{t,pro,a}, P_{t,pro})$ over all triples in the test knowledge graph G_{Test} would produce a flawed metric. To show that, I first consider the relevance fraction of pro paths in P_t or $RF_t(P_{t,pro}, P_t)$.

Triples with low values for $RF_t(P_{t,pro}, P_t)$ would have the same impact on the averaged value of $RF_t(P_{t,pro,a}, P_{t,pro})$ as triples with high values for $RF_t(P_{t,pro}, P_t)$. This contradicts the requirement mentioned above that the metric should be sensible to the absolute relevance scores of paths. I fix this issue by weighting the values of $RF_t(P_{t,pro,a}, P_{t,pro})$ with $RF_t(P_{t,pro}, P_t)$ and normalizing the output with the total relevance associated with pro paths in the test set. This leads to the weighted relevance fraction of annotation a in pro paths $WRF_{pro}(a)$:

$$WRF_{pro}(a) = \frac{1}{\sum_{t \in G_{Test}} RF_t(P_{t,pro}, P_t)} \sum_{t \in G_{Test}} RF_t(P_{t,pro}, P_t) RF_t(P_{t,pro,a}, P_{t,pro}). \tag{4.3}$$

For the set of annotation classes A, it holds that

$$\sum_{a \in A} WRF_{pro}(a) = 1 \tag{4.4}$$

This means that $WRF_{pro}(a)$ is normalized and its output lies in $[0, 1]$.

I now define the metric $WRFM_{pro}$ which combines WRF values and the criteria for meaningful explanations given in Section 4.3.2:

$$WRFM_{pro} = \sum_{a \in \{0a, 1a\}} WRF_{pro}(a) \tag{4.5}$$

To also take into account the annotations of type b), I define a weak variant of $WRFM_{pro}$:

$$WRFM_{pro}^{(weak)} = WRFM_{pro} + \sum_{a \in \{0b, 1b\}} WRF_{pro}(a) \tag{4.6}$$

A perfect model should be able to achieve values close to 1 in the weak variant.[6]

4.3.6 Experimental setup

As manually annotating explanations for predictions is time consuming, only a small subset of triples from the test set can be evaluated. I generate the set used for the annotation by the following procedure: First, I manually select a set of 24 relations that cover topics that require not too much expert knowledge when deciding over the relevance of paths for triples. This test set does not contain the relations I use for validation. I list the selected relations in the Appendix A. For each selected relation, I randomly sample two facts from the test set. In order to obtain negative samples, I randomly corrupt $e1$ or $e2$ in each sampled fact. This results in 96 triples in total. The selected relations make up for 13.2 % of the facts in the test set. The selected relations include both infrequent relations that occur only two times in the test set and frequent relations that occur up to 1451 times in the test set.

When considering the relevance of context paths for a triple t, it is trivial to assign the path $p = t$ a high relevance score. The triple t itself if less interesting for the explainability of the model since it only reveals how the underlying KBC model, TransE in my case, scores the correctness of t. To focus the evaluation on more interesting paths, I use the setting of the exclusive context path model (ECPM). There might be triples for which no relevant context paths exist. I account for that by including only those triples into to the evaluation that have at least 10 % of the total relevance scores assigned to paths other than t (in the

[6] A perfect model does not need to achieve the value 1 exactly. For example, necessary paths can also provide some evidence for the correctness of triples, which is omitted in the annotation scheme.

Path length	1	2	3
Avg. number of paths	1.34	0.41	1.00

Table 4.3: Average number of paths per triple that are displayed and used for annotation.

non-exclusive setting). This aims to detect those cases in which the context path model is not able to capture any relevant context paths. This excludes 17 out of 96 triples from the evaluation. The relations of the excluded triples are listed in Appendix B.

Typically, there are only few paths per triple t that constitute the majority of the sum of relevance scores assigned to context paths of t. I make use of this by annotating only paths that are assigned at least 5 % of the total relevance. In practice, this means that on average 80 % of the total amount of relevance scores per triple is displayed and annotated. Table 4.3 lists the resulting numbers of paths per triple that are displayed and annotated.

4.3.7 Path reliability results

In Figure 4.2, I show the results of the path reliability evaluation. Considering the average WRFM values of pro paths and con paths, one can conclude that 77 % of the relevance scores associated with displayed paths is considered reliable by the logical formulation and 81 % is considered reliable when following the probabilistic formulation. Note that paths that are not considered reliable can as well be helpful for users when deciding whether to trust the prediction of the model.

The majority of relevance scores is assigned to paths that are equivalent with the predicted triple. This is expected since the CPM is only capable of modeling equivalence connections. Due to this symmetric property of the CPM, it is also not surprising that the results for pro paths and con paths are very similar and that the distribution of relevance scores does not differ significantly between pro paths and con paths.

I also state the absolute numbers of annotations in Figure 4.2 to indicate that these results are, especially for classes other than 0 a), limited in their resilience.

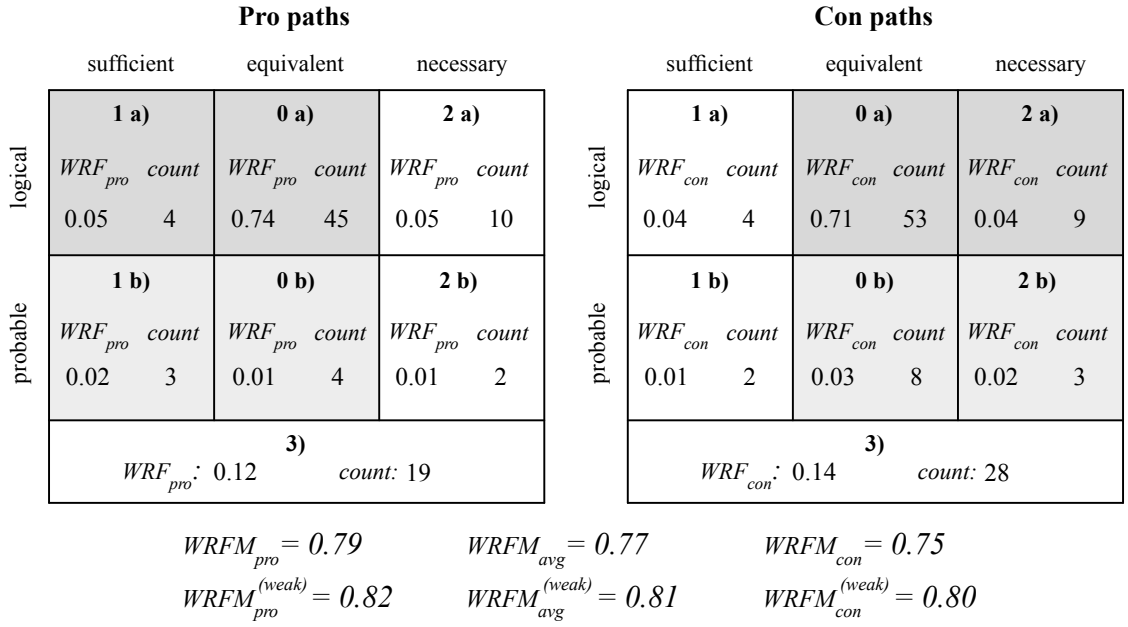

Figure 4.2: Results of the context path reliability evaluation. WRF values and the absolute numbers of annotations are listed. Cells that are considered by $WRFM$ are marked using a grey background. The cells that are only considered by $WRFM^{(weak)}$ are marked with a light grey background. The average of the WRFM of pro paths and con paths is also given.

4.3.8 Similar paths

On the one hand, the CPM is able to find paths that are semantically similar to a given triple and use them as evidence for or against the correctness of the triple. On the other hand, when using paths as explanation, the user might not want to read many similar paths. This can be a problem especially for knowledge bases that already have many similar or redundant relations.

For the relation */sports/sports_team/roster./american_football/football_roster_position/position*, the CPM assigns the following relation sequences relevance scores of at least 85% compared to the relevance score for the relation itself. They are all very similar or equivalent.

p_1 = */american_football/football_team/current_roster./american_football/football_roster_position/position*

p_2 = */american_football/football_team/current_roster./american_football/football_roster_position/player*, */sports/pro_athlete/teams./american_football/football_roster_position/position*

p_3 = */american_football/football_team/current_roster./sports/sports_team_roster/player*, */sports/pro_athlete/teams./american_football/football_roster_position/position*

p_4 = */american_football/football_player/current_team./american_football/*$^{-1}$ */football_roster_position/team*, */sports/pro_athlete/teams./american_football/football_roster_position/position*

p_5 = */sports/pro_athlete/teams./american_football/football_roster_position/team*$^{-1}$, */sports/pro_athlete/teams./american_football/football_roster_position/position*

4.4 Evaluation of Fact Prediction

The results of the ranking evaluation for predicting the correctness score of triples are presented in Table 4.4. The initial observation, especially in terms of the hits at 10 metric, is that the CPM does not reach the same ranking performance as TransE. When considering the mean quantile metric, the CPM only has a 0.8 % weaker performance than TransE.

An obvious reason is that the hyperparameters for the CPM are chosen to optimize both for high performance on the KBC task and for providing meaningful explanations. I give more details on that in Section 4.5. Despite that, one cannot directly follow that there has to be a tradeoff between a more interpretable model and a higher performance because the CPM is a more complex optimization problem than the TransE model. This difference in performance could therefore also be a question of hyperparameter tuning. Another reason for the difference in performance could also be that the CPM uses a TransE model that is trained on paths for scoring the correctness of paths. The path training already reduces the hits at 10 value by 2.5 % compared to the TransE model that is trained only on single edges. As I also mention in Section 4.3.1, this might be specific the FB15K data set. To answer those open questions, more extensive experiments with several data sets would be required.

Considering the ECPM[7], it is remarkable that the performance in ranking is only reduced by 7.1 % in hits at 10 and by 1.7 % in mean quantile when ignoring the representation of a triple t completely while predicting its correctness. This indicates that a large amount of

[7] All tested triples have at least one context path other than t itself. In 0.002 % of the neg samples, there is no context path. I then use the triple itself to estimate the correctness of the negative sample because excluding it would make the ranking problem easier.

	TransE	TransE (path)	TransE (path) %red	CPM	CPM %red	ECPM	ECPM %red
Hits at 10	**92.8**	90.5	2.5	89.0	4.1	86.2	7.1
Mean quantile	**97.7**	97.6	0.1	96.9	0.8	96.0	1.7

Table 4.4: Results of the ranking evaluation for predicting the correctness of triples. TransE (path) is the TransE model trained on paths. X %red is the reduction in performance of model X compared to the TransE model trained on single edges. For efficiency, the CPM and ECPM are evaluated on a random sample of the test set of relative size 25 %.

the expressiveness of TransE can actually be replicated by just considering the context of t and not t itself. This amount of expressiveness is also what can be made interpretable to a user by displaying the context paths.

4.5 Hyperparameters

For the validation of the explanations, I use a set of 10 relations that are different from the relations used in the final experiment. Besides the criteria for meaningful explanations given in Section 4.3.2, I also consider the number of paths with high relevance scores for selecting hyperparameters. This ensures that the CPM not only uses the triple itself as context path since this would be equivalent to the plain TransE model. In order to force the model to take into account not just the relation of the triple itself or very similar relations, I use random samples of context paths of relative size 50 % during training. This is also called dropout and encourages the model to not just memorize the training data but to generalize it.

I use the gradient based optimizer Adam [KB15] for minimizing both objectives. The following hyperparameters are selected based on the performance on the validation set. For learning $score(p)$, I choose the learning rate 0.0001 and for learning $r(t,p)$, I choose 0.001, both selected from $[0.001, 0.0005, 0.0001]$. I use mini-batches (subsets of the whole data set used for one optimization step) of size 300 when learning $score(p)$ and of size 30 when learning $r(t,p)$. All parameters are initialized using a normal distribution with variance 0.1. I do not constrain the parameters to be normalized during training.

5 Conclusion

This work considers the problem of knowledge base completion (KBC). A common approach for the KBC problem is to learn representations for entities and relations that allow for generalizing existing connections in the knowledge base to predict the correctness of a triple that is not in the knowledge base.

In this work, I propose the *context path model*, which is based on this approach. In contrast to existing KBC models, it also provides explanations for predictions. For this purpose, it uses paths that capture the context of a given triple. The proposed model can be used on top of several state of the art KBC models. I demonstrate this using the TransE model [BUG+13]. In a manual evaluation, I observe that most of the paths the model uses as explanation are meaningful and provide evidence for assessing the correctness of triples. I also show in an experiment that the performance of the context path model on a standard KBC task is close to TransE. The experiments also reveal that a large part of the expressiveness of TransE can be replicated without representing a given triple directly by only considering paths that capture the context the triple. This shows the potential of using paths as explanations for KBC predictions.

Future research could include extending the model to capture more complex connections between paths and relations like necessary and sufficient conditions. To provide a more resilient evaluation of the explanations, a larger data set could be annotated. Furthermore, methods for reducing redundant or very similar paths in explanations could be investigated.

A Manually selected relations used for the annotation

/location/location/containedby
/food/diet/followers
/organization/organization/geographic_scope
/business/business_operation/assets./measurement_unit/dated_money_value/currenc
/time/time_zone/locations_in_this_time_zon
/music/genre/artists
people/person/place_of_birth
/sports/sport/leagues
/award/award_category/nominees./award/award_nomination/award_nominee
/people/person/gender
/soccer/football_player/current_team./sports/sports_team_roster/position
/education/field_of_study/subdiscipline_of
/medicine/disease/causes
/travel/travel_destination/how_to_get_here./travel/transportation/mode_of_transportation
/people/person/parents
/people/person/children
/people/person/languages
/people/person/profession
/music/instrument/instrumentalists
/film/film/genre
/location/location/people_born_here
location/country/currency_used
/geography/island_group/islands_in_group
/award/award_winning_work/awards_won./award/award_honor/award_winner

B Excluded relations

These relations have corresponding facts that are excluded from the annotation because of their lack of context paths with high relevance scores. The numbers in parentheses state how many facts with the respective relation are excluded.

/music/genre/artists (4)
/time/tim_zone/locations_in_this_time_zone (3)
/people/person/profession (2)
/people/person/languages (2)
/people/person/gender (4)
/travel/travel_destination/how_to_get_here./travel/transportation/mode_of_transportation (2)

Bibliography

[BCFL13] J. Berant, A. Chou, R. Frostig, P. Liang. "Semantic Parsing on Freebase from Question-Answer Pairs". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, 2013, pp. 1533–1544. URL: http://aclweb.org/anthology/D13-1160 (cit. on p. 7).

[BCV13] Y. Bengio, A. Courville, P. Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (Aug. 2013), pp. 1798–1828. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2013.50 (cit. on p. 12).

[BEP+08] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor. "Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge". In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. SIGMOD '08. Vancouver, Canada: ACM, 2008, pp. 1247–1250. ISBN: 978-1-60558-102-6. DOI: 10.1145/1376616.1376746. URL: http://doi.acm.org/10.1145/1376616.1376746 (cit. on pp. 7, 30).

[BUG+13] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko. "Translating Embeddings for Modeling Multi-relational Data". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger. Curran Associates, Inc., 2013, pp. 2787–2795. URL: http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf (cit. on pp. 13, 20, 22, 29–31, 45).

[BWU14] A. Bordes, J. Weston, N. Usunier. "Open Question Answering with Weakly Supervised Embedding Models". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by T. Calders, F. Esposito, E. Hüllermeier, R. Meo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 165–180. ISBN: 978-3-662-44848-9. DOI: 10.1007/978-3-662-44848-9_11 (cit. on p. 7).

[Cha18] N. Chah. "OK Google, What Is Your Ontology? Or: Exploring Freebase Classification to Understand Google's Knowledge Graph". In: *arXiv preprint arXiv:1805.03885* (2018) (cit. on p. 30).

[GML15] K. Guu, J. Miller, P. Liang. "Traversing Knowledge Graphs in Vector Space". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015). DOI: 10.18653/v1/d15-1038. URL: http://dx.doi.org/10.18653/v1/D15-1038 (cit. on pp. 7, 14–16, 22, 31, 32).

[GTHS13] L. A. Galárraga, C. Teflioudi, K. Hose, F. Suchanek. "AMIE: Association Rule Mining Under Incomplete Evidence in Ontological Knowledge Bases". In: *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13. Rio de Janeiro, Brazil: ACM, 2013, pp. 413–422. ISBN: 978-1-4503-2035-1. DOI: 10.1145/2488388.2488425. URL: http://doi.acm.org/10.1145/2488388.2488425 (cit. on pp. 12, 15, 16).

[HMR86] G. E. Hinton, J. L. McClelland, D. E. Rumelhart. "Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1". In: ed. by D. E. Rumelhart, J. L. McClelland, C. PDP Research Group. Cambridge, MA, USA: MIT Press, 1986. Chap. Distributed Representations, pp. 77–109. ISBN: 0-262-68053-X. URL: http://dl.acm.org/citation.cfm?id=104279.104287 (cit. on p. 12).

[Hol18] A. Holzinger. "Explainable AI (ex-AI)". In: *Informatik-Spektrum* 41.2 (Apr. 2018), pp. 138–143. ISSN: 1432-122X. DOI: 10.1007/s00287-018-1102-5. URL: https://doi.org/10.1007/s00287-018-1102-5 (cit. on p. 8).

[KB15] D. P. Kingma, J. Ba. "Adam: A Method for Stochastic Optimization". In: *ICLR*. 2015. URL: http://arxiv.org/abs/1412.6980 (cit. on p. 44).

[KBK17] R. Kadlec, O. Bajgar, J. Kleindienst. "Knowledge Base Completion: Baselines Strike Back". In: *Proceedings of the 2nd Workshop on Representation Learning for NLP*. 2017, pp. 69–74. URL: https://www.aclweb.org/anthology/W17-2609 (cit. on p. 15).

[LLL+15] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, S. Liu. "Modeling Relation Paths for Representation Learning of Knowledge Bases". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 705–714. DOI: 10.18653/v1/D15-1082. URL: http://aclweb.org/anthology/D15-1082 (cit. on pp. 16, 31).

[MGW+13] B. Min, R. Grishman, L. Wan, C. Wang, D. Gondek. “Distant Supervision for Relation Extraction with an Incomplete Knowledge Base”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, 2013, pp. 777–782. URL: http://aclweb.org/anthology/N13-1095 (cit. on p. 7).

[NTK11] M. Nickel, V. Tresp, H.-P. Kriegel. “A Three-way Model for Collective Learning on Multi-relational Data”. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML’11. Bellevue, Washington, USA: Omnipress, 2011, pp. 809–816. ISBN: 978-1-4503-0619-5. URL: http://dl.acm.org/citation.cfm?id=3104482.3104584 (cit. on pp. 14, 20).

[SKW07] F. M. Suchanek, G. Kasneci, G. Weikum. “Yago: A Core of Semantic Knowledge”. In: *Proceedings of the 16th International Conference on World Wide Web*. WWW ’07. Banff, Alberta, Canada: ACM, 2007, pp. 697–706. ISBN: 978-1-59593-654-7. DOI: 10.1145/1242572.1242667. URL: http://doi.acm.org/10.1145/1242572.1242667 (cit. on p. 7).

[SVT+12] T. Steiner, R. Verborgh, R. Troncy, J. Gabarro, R. Van De Walle. “Adding Realtime Coverage to the Google Knowledge Graph”. In: *Proceedings of the 2012th International Conference on Posters & Demonstrations Track - Volume 914*. ISWC-PD’12. Boston, USA: CEUR-WS.org, 2012, pp. 65–68. URL: http://dl.acm.org/citation.cfm?id=2887379.2887396 (cit. on p. 7).

[VK14] D. Vrandečić, M. Krötzsch. “Wikidata: A Free Collaborative Knowledge Base”. In: *Communications of the ACM* 57 (2014), pp. 78–85. URL: http://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext (cit. on p. 7).

[WFH11] I. H. Witten, E. Frank, M. A. Hall. “Chapter 6 - Implementations: Real Machine Learning Schemes”. In: *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*. Ed. by I. H. Witten, E. Frank, M. A. Hall. Third Edition. The Morgan Kaufmann Series in Data Management Systems. Boston: Morgan Kaufmann, 2011, pp. 191–304. ISBN: 978-0-12-374856-0. DOI: https://doi.org/10.1016/B978-0-12-374856-0.00006-7. URL: http://www.sciencedirect.com/science/article/pii/B9780123748560000067 (cit. on p. 11).

[YYH+14] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng. “Embedding entities and relations for learning and inference in knowledge bases”. In: *arXiv preprint arXiv:1412.6575* (2014) (cit. on pp. 15, 16, 20, 22, 23, 31).

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift