Can We Predict *New* Facts with Open Knowledge Graph Embeddings? A Benchmark for Open Link Prediction

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, Rainer Gemulla
Data and Web Science Group
University of Mannheim, Germany
broscheit@informatik.uni-mannheim.de,
{k.gashteovski,ywang,rgemulla}@uni-mannheim.de,

Abstract

Open Information Extraction systems extract *("subject text", "relation text", "object text")* triples from raw text. Some triples are textual versions of facts, i.e., non-canonicalized mentions of entities and relations. In this paper, we investigate whether it is possible to infer *new* facts directly from the *open knowledge graph* without any canonicalization or any supervision from curated knowledge. For this purpose, we propose the open link prediction task, i.e., predicting test facts by completing *("subject text", "relation text", ?)* questions. An evaluation in such a setup raises the question if a correct prediction is actually a *new* fact that was induced by reasoning over the open knowledge graph or if it can be trivially explained. For example, facts can appear in different paraphrased textual variants, which can lead to test leakage. To this end, we propose an evaluation protocol and a methodology for creating the open link prediction benchmark OLPBENCH. We performed experiments with a prototypical knowledge graph embedding model for open link prediction. While the task is very challenging, our results suggests that it is possible to predict genuinely new facts, which can not be trivially explained.

1 Introduction

A knowledge graph (KG) (Hayes-Roth, 1983) is a set of (subject, relation, object)-triples, where the subject and object correspond to vertices, and relations to labeled edges. In curated KGs, each triple is fully disambiguated against a fixed vocabulary of entities[1] and relations.

An application for KGs, for example, is the problem of drug discovery based on bio-medical knowledge (Mohamed et al., 2019). The construction of a curated bio-medical KG, which is required for such an approach, is challenging and constrained by the available amount of human effort and domain expertise. Many tools that could assist humans in KG construction (e.g., an entity linker) need a KG to begin with. Moreover, current methods for KG construction often rely on the rich structure of Wikipedia, such as links and infoboxes, which are not available for every domain. Therefore, we ask if it is possible to make predictions about, for example, new drug applications from raw text without the intermediate step of KG construction.

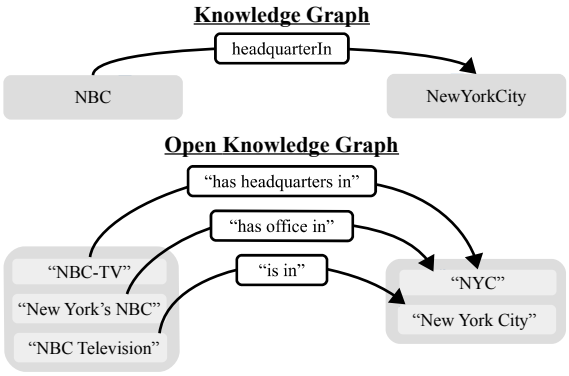

Figure 1: Entities and relations in curated knowledge graphs vs. open knowledge graphs.

Open information extraction systems (OIE) (Etzioni et al., 2011) automatically extract *("subject text", "relation text", "object text")*-triples from unstructured data such as text. We can view OIE data as an *open knowledge graph* (OKG) (Galárraga et al., 2014), in which vertices correspond to *mentions of entities* and edges to *open relations* (see Fig. 1). Our overarching interest is whether and how we can reason over an OKG without any canonicalization and without any supervision on its latent factual knowledge. The focus of this study are the challenges of benchmarking the inference abilities of models in such a setup.

A common task that requires reasoning over a

[1]For brevity, "entities" denotes both entities (e.g. Prince) and concepts (e.g. musician) throughout the paper.

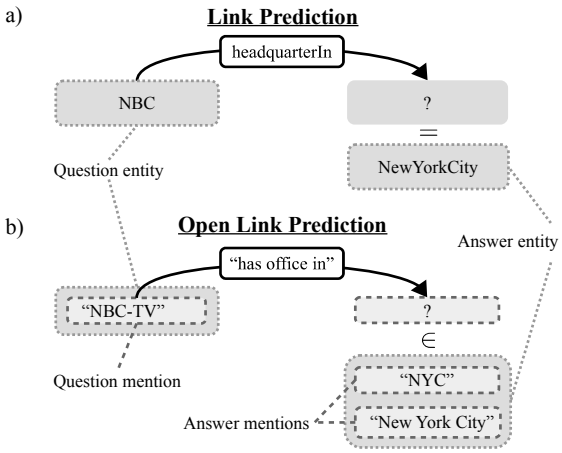

Figure 2: Comparing evaluation of link prediction and open link prediction.

KG is link prediction (LP). The goal of LP is to predict missing facts in a KG. In general, LP is defined as answering questions such as *(NBC, headquarterIn, ?)* or *(?, headquarterIn, NewYorkCity)*; see Fig. 2a. In OKGs, we define *open link prediction* (OLP) as follows: Given an OKG and a question consisting of an entity mention and an open relation, predict *mentions* as answers. A predicted mention is correct if it is a mention of the correct answer entity. For example, given the question *("NBC-TV", "has office in", ?)*, correct answers include *"NYC"* and *"New York"*; see Fig. 2b).

To evaluate LP performance, the LP model is trained on known facts and evaluated to predict unknown facts, i.e., facts not seen during training. A simple but problematic way to transfer this approach to OKGs is to sample a set of evaluation triples from the OKG and to use the remaining part of the OKG for training. To see why this approach is problematic, consider the test triple *("NBC-TV", "has office in", "New York")* and suppose that the triple *("NBC", "has headquarter in", "NYC")* is also part of the OKG. The latter triple essentially leaks the test fact. If we do not remove such facts from the training data, a successful models only *paraphrases* known facts but does not perform reasoning, i.e., does not predict genuinely new facts.

Furthermore, we also want to quantify if there are other trivial explanations for the prediction of an evaluation fact. For example, how much can be predicted with simple popularity statistics, i.e., *only* the mention, e.g. *("NBC-TV", ?)*, or *only* the relation,e.g. *("has office in", ?)*. Such *non-relational* information also does not require reasoning over the graph.

To experimentally explore whether it is possible to predict new facts, we focus on knowledge graph embedding (KGE) models (Nickel et al., 2016), which have been applied successfully to LP in KGs. Such models can be easily extended to handle the surface forms of mentions and open relations.

Our contributions are as follows: We propose the OLP task, an OLP evaluation protocol, and a method to create an OLP benchmark dataset. Using the latter method, we created a large OLP benchmark called OLPBENCH, which was derived from the state-of-the-art OIE corpus OPIEC (Gashteovski et al., 2019). OLPBENCH contains 30M open triples, 1M distinct open relations and 2.5M distinct mentions of approximately 800K entities. We investigate the effect of *paraphrasing* and *non-relational* information on the performance of a prototypical KGE model for OLP. We also investigate the influence of entity knowledge during model selection with different types of validation data. For training KGE models on such large datasets, we describe an efficient training method.

In our experiments, we found the OLP task and OLPBENCH to be very challenging. Still, the KGE model we considered was able to predict genuinely new facts. We also show that paraphrasing and non-relational information can indeed dilute performance evaluation, but can be remedied by appropriate dataset construction and experimental settings.

2 Open Knowledge Graphs

OKGs can be constructed in a fully automatic way. They are open in that they do not require a vocabulary of entities and relations. For this reason, they can capture more information than curated KGs. For example, different entity mentions can refer to different versions of an entity at different points of time, e.g., *"Senator Barack Obama"* and *"President Barack Obama"*. Similarly, relations may be of varying specificity: *headquarterIn* may be expressed directly by open relations such as *"be based in"* or *"operate from"* but may also be implied by *"relocated their offices to"*. In contrast to KGs, OKGs contain rich conceptual knowledge. For example, the triple *("a class action lawsuit", "is brought by", "shareholders")* does not directly encode entity knowledge, although it does provide information about entities that link to *"a class action lawsuit"* or *"shareholders"*.

OKGs tend to be noisier and the factual knowledge is less certain than in a KG, however. They

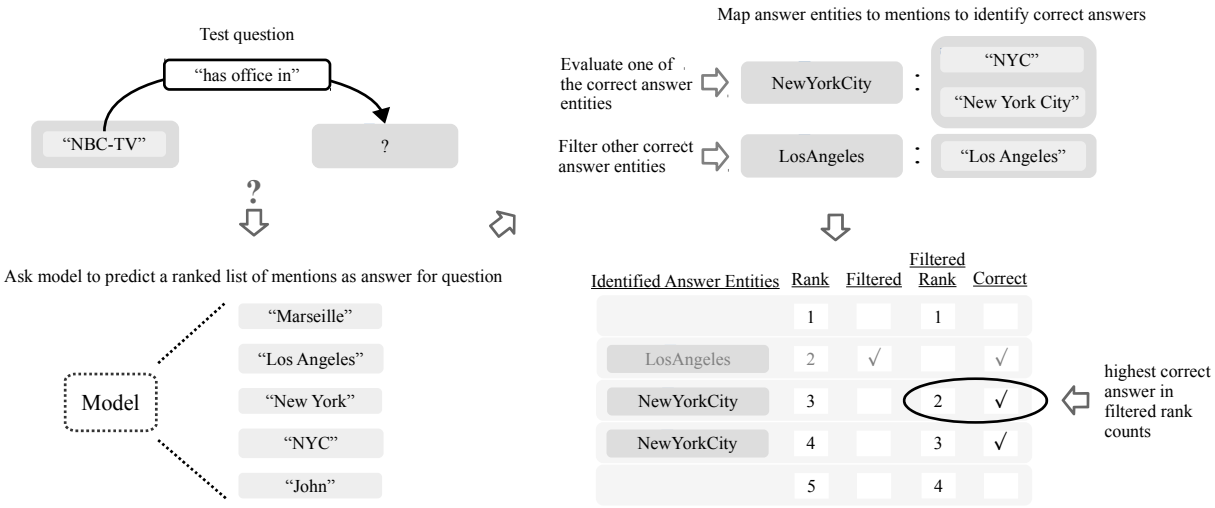

Figure 3: **Mention-ranking protocol**: Example for computing the **filtered rank** for a test question.

can not directly replace KGs. OKGs have mostly been used as a weak augmentation to KGs, e.g., to infer new unseen entities or to aid link prediction (see App. A for a comprehensive discussion of related work). Much of prior work that solely leverages OKGs without a reference KG—and therein is closest to our work—focused on canonicalization and left inference as a follow-up step (Cohen et al., 2000, inter alia). In contrast, we propose to evaluate inference in OKGs with OLP directly.

3 Open Link Prediction

The open link prediction task is based on the *link prediction* task for KGs (Nickel et al., 2016), which we describe first. Let $\mathcal{E}$ be a set of entities, $\mathcal{R}$ be a set of relations, and $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ be a knowledge graph. Consider questions of the form $q_h = (?, k, j)$ or $q_t = (i, k, ?)$, where $i, j \in \mathcal{E}$ is a *head* and *tail* entity, respectively, and $k \in \mathcal{R}$ is a relation. The link prediction problem is to provide answers that are correct but not yet present in $\mathcal{T}$.

In OKGs, only mentions of entities and open relations are observed. We model each entity mention and each open relation as a non-empty sequence of tokens from some vocabulary $\mathcal{V}$ (e.g., a set of words). Denote by $\mathcal{M} = \mathcal{V}^+$ the set of all such sequences and observe that $\mathcal{M}$ is unbounded. An open knowledge graph $\mathcal{T} \subset \mathcal{M} \times \mathcal{M} \times \mathcal{M}$ consists of triples of form (i, k, j), where $i, j \in \mathcal{M}$ are head and tail *entity mentions*, resp., and $k \in \mathcal{M}$ is an *open relation*. Note that we overload notation for readability: i, j, and k refer to entity mentions and open relations in OKGs, but to disambiguated entities and relations in KGs. The intended meaning will always be clear from the context. We denote by $\mathcal{M}(\mathcal{E})$ and $\mathcal{M}(\mathcal{R})$ the sets of entity and relations present in $\mathcal{T}$, respectively. The *open link prediction task* is to predict new and correct answers to questions $(i, k, ?)$ or $(?, k, j)$. Answers are taken from $\mathcal{M}(\mathcal{E})$, whereas questions may refer to arbitrary mentions of entities and open relations from $\mathcal{M}$. For example, for the question ("*NBC-TV*", "*has office in*", ?), we expect an answer from the set of mentions {*"New York"*, *"NYC"*, ...} of the entity *NewYorkCity*. Informally, an answer (i, k, j) is correct if there is a correct triple (e_1, r, e_2), where e_1 and e_2 are entities and r is a relation, such that i,j, and k are mentions of e_1, e_2, and r, respectively.

3.1 Evaluation protocol

To describe our proposed evaluation protocol, we first revisit the most commonly used methodology to evaluate link prediction methods for KGs, i.e., the entity-ranking protocol (Bordes et al., 2013). Then, we discuss its adaptation to OLP, which we call the mention-ranking protocol (see Fig. 3).

KGs and entity ranking. For each triple $z = (i, k, j)$ in the evaluation data, a link prediction model ranks the answers for two questions, $q_t(z) = (i, k, ?)$ and $q_h(z) = (?, k, j)$. The model is evaluated based on the ranks of the correct entities j and i; this setting is called *raw*. When true answers for $q_t(z)$ and $q_h(z)$ other than j and i are filtered from the rankings, then the setting is called *filtered*.

OKGs and mention ranking. In OLP, the model predicts a ranked list of mentions. But questions might have multiple *equivalent* true answers,

i.e., answers that refer to the same entity but use different mentions. Our evaluation metrics are based on the highest rank of a correct answer mention in the ranking. For the filtered setting, the mentions of known answer entities other than the evaluated entity are filtered from the ranking. This *mention-ranking protocol* thus uses knowledge of alternative mentions of the entity in the evaluation triple to obtain a suitable ranking. The mention-ranking protocol therefore requires (i) ground truth annotations for the entity mentions in the head and tail of the evaluation data, and (ii) a comprehensive set of mentions for these entities.

4 Creating the Open Link Prediction Benchmark OLPBENCH

An OLP benchmark should enable us to evaluate a model's capability to predict genuinely new facts, i.e., facts can not be trivially derived. Due to the nature of OKGs, paraphrasing of facts may leak facts from validation and test data into training, making the prediction of such evaluation facts trivial. Nevertheless, the creation of training and validation data should require as little human effort as possible so that the methodology can be readily applied to new domains. Our mention-ranking protocol uses knowledge about entities for disambiguation (of the evaluation data, not the training data), however, which requires human effort to create. We investigate experimentally to what extent this entity knowledge is necessary for model selection and, in turn, how much manual effort is required to create a suitable validation dataset.

In the following, we describe the source dataset of OLPBENCH and discuss how we addressed the points above to create evaluation and training data.

4.1 Source Dataset

OLPBENCH is based on OPIEC (Gashteovski et al., 2019), a recently published dataset of OIE triples that were extracted from the text of English Wikipedia with the state-of-the-art OIE system MinIE (Gashteovski et al., 2017). We used a subset of 30M distinct triples, comprised of 2.5M entity mentions and 1M open relations. In 1.25M of these triples, the subject *and* the object contained a Wikipedia link. Fig. 4 shows how a Wikipedia link is used to disambiguate a triple's subject and object mentions. Tab. 1 shows an excerpt from the unlinked and linked triples. For the evaluation protocol, we collected a dictionary, where each entity

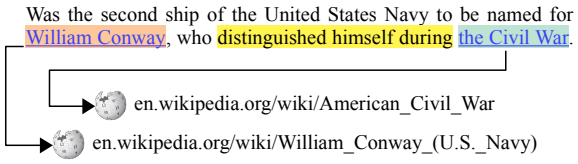

Figure 4: Example for a (subject , relation , object) triple extracted from Wikipedia. With a Wikipedia hyperlink, a mention is disambiguated to its entity. Inversely this yields a mapping from an entity to all its mentions.

is mapped to all possible mentions. See App. B for more details about the dataset creation.

4.2 Evaluation Data

From the source dataset, we created validation and test data with the following requirements:

Data quality. The evaluation data should be challenging, and noise should be limited as much as possible. We chose a pragmatic and easy heuristic: we did not consider short relations with less than three tokens as candidates for sampling evaluation data. This decision was based on the following observations: (i) Due to the OPIEC's extractions, short relations—e.g. *("kerry s. walters", "is", "professor emeritus")*—are often subsumed by longer relations—e.g. *("kerry s. walters", "is professor emeritus of", "philosophy")*—, which would always lead to leakage from the longer relation to the shorter relation. (ii) Longer relations are less likely to be easily captured by simple patterns that are already successfully used by KG construction methods, e.g. *("elizabeth of hungary", "is the last member of", "the house of árpád")*. We conjecture that long relations are more interesting for evaluation to measure progress in reasoning with OKG data. (iii) The automatically extracted entity annotations were slightly noisier for short relations; e.g., *("marc anthony", "is" "singer")* had the object entity annotation *SinFrenos*.

Human effort for data creation. The mention-ranking protocol uses knowledge about entities for disambiguation. We want to experimentally quantify the influence of this entity knowledge on model selection, i.e., whether entity knowledge is necessary to find a good model. If so, human expertise is necessary to create the validation data. While our goal is to require almost no human domain expertise to learn a good model, the size of validation data is much smaller than the size of the training data. Therefore, this effort—if helpful—may be

	subject	relation	object	subject mentions	object mentions
Unlinked	conway	has	plot		
	henry s. conway	is	field marshal		
	conway tearle	has	members		
	highway 319	begins outside	conway		
	bloomsbury	bought	conway publishing		
	mike conway	is teammate of	toyota		
	w. conway gordon	served as adc to	gen. p. maitland		
	w. conway gordon	entered	the service		
Linked	willam conway	distinguished himself during	civil war	willam conway conway	civil war american civil war
	terry venables	is manager of	fc barcelona	terry venables	fc barcelona f.c. barcelona futbol club barcelona cf barcelona barcelona
	background music	is composed by	hikaru nanase	the background music background music background score	masumi ito hikaru nanase

Table 1: Example from the unlinked and linked data in OLPBENCH. For the unlinked data, we show the first of 3443 triples from the unlinked data containing the token *"conway"*. For the linked data, we show the triples and also the alternative mentions for their entities. The first linked triple is about *William_Conway_(U.S._Navy)*.

feasible. To investigate this, we perform model selection performed with three different validation datasets that require increasingly more human effort to create: VALID-ALL (no effort), VALID-MENTION (some effort) and VALID-LINKED (most amount of human effort).

TEST and VALID-LINKED data. Sample 10K triples with relations that have at least three tokens from the 1.25M linked triples. In these triples, the subject and object mentions have an annotation for their entity, which allows the mention-ranking protocol to identify alternative mentions of the respective entities.

VALID-MENTION data. Proceed as in VALID-LINKED but discard the entity links. During validation, no access to alternative mentions is possible so that the mention-ranking protocol cannot be used. Nevertheless, the data has the same distribution as the test data. Such validation data may be generated automatically using a named entity recognizer, if one is available for the target domain.

VALID-ALL data. Sample 10K triples with relations that have at least three tokens from the entire 30M unlinked and linked triples. This yields mostly triples from the unlinked portion. These triples may also include common nouns such as "a nice house" or "the country". Entity links are discarded, i.e., the mention-ranking protocol cannot be used for validation.

4.3 Training Data

To evaluate LP models for KGs, evaluation facts are generated by sampling from the KG. Given an evaluation triple (i, k, j), the simplest action to avoid leakage from the training data is to remove only this evaluation triple from training. For KGs, it was observed this simple approach is not satisfactory in that evaluation answers may still leak and thus can be trivially inferred (Toutanova et al., 2015; Dettmers et al., 2018). For example, an evaluation triple $(a, siblingOf, b)$ can be trivially answered with the training triple $(b, siblingOf, a)$.

In OKGs, paraphrases of relations pose additional sources of leakage. For example, the relations *"is in"* and *"located in"* may contain many of the same entity pairs. For evaluation triple (i, k, j), such leakage can be prevented by removing *any* other relation between i and j from the training data. However, individual tokens in the arguments or relations may also cause leakage. For example, information about test triple (*"NBC-TV", "has office in", "NYC"*) is leaked by triples such as (*"NBC Television", "has NYC offices in", "Rockefeller Plaza"*) even though it has different arguments. Fig. 5 visualizes this example.

We use three levels of leakage removal from training: SIMPLE, BASIC, and THOROUGH. To match evaluation triple (i, k, j) with training triples, we ignored word order and stopwords.

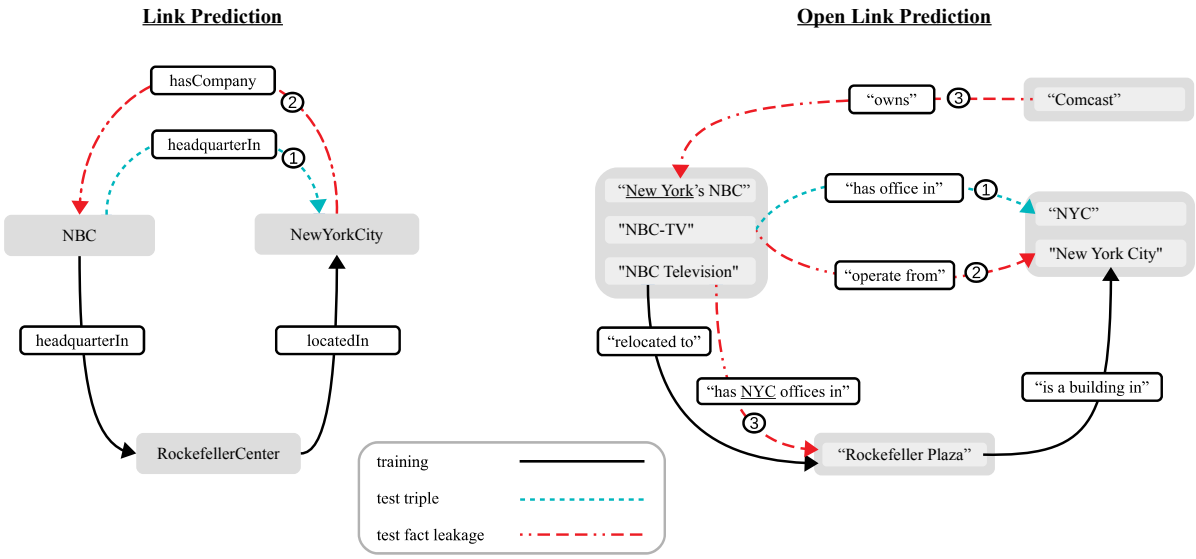

Figure 5: Comparison of removal of test data from training data for link prediction compared to open link prediction. The example test triples are *(NBC, headquarterIn, NewYorkCity)* and *("NBC-TV", "is in", "NYC")*, respectively. **Link Prediction**: (1) remove the test fact from training, (2) remove any link between the test fact's arguments; **Open Link Prediction**: (1) remove only the open test triple from training, (2) consider any link between any mention of the open triple's arguments, (3) consider test leakage from the *tokens* in the open triple's arguments or relation. Underlined tokens are the source of leakage.

SIMPLE removal. Only the triple (i, k, j) is removed. Triples with alternative mentions for i or j are kept.

BASIC removal. (i, k, j) as well as (j, k, i) are removed from the training data. Triples with with alternative mentions of i and j are also removed.

THOROUGH removal. Additionally to BASIC removal, we also remove triples from training matched by the following patterns. The patterns are explained with the example *("J. Smith", "is defender of", "Liverpool")*:

(a) $(i, *, j)$ and $(j, *, i)$. E.g., matches *("J. Smith", "is player of", "Liverpool")*.

(b) $(i, k + j, *)$ and $(*, k + i, j)$.[2] E.g., matches *("J. Smith", "is Liverpool's defender on", "Saturday")*.

(c) $(i + k + j, *, *)$ and $(*, *, i + k + j)$. E.g., matches *("Liverpool defender J. Smith", "kicked", "the ball")*.

For OLPBENCH, THOROUGH removed 196,717 more triples from the OKG than BASIC. Note that this yields three different training data sets.

[2]Other permutations of this pattern did not occur in our data.

5 Open Knowledge Graph Embeddings

KG embedding (KGE) models have been successfully applied for LP in KGs, and they can be easily extended to handle surface forms, i.e., mentions and open relations. We briefly describe KGE models and their extension.

Knowledge Graph Embedding (KGE) model. A KGE model (Nickel et al., 2016) associates an embedding with each entity and each relation. The embeddings are dense vector representations that are learned with an LP objective. They are used to compute a KGE model-specific *score* $s(i, k, j)$ for a triple (i, k, j); the goal is to predict high scores for true triples and low scores for wrong triples.

KGE model with composition. For our experiments, we considered composition functions to create entity and relation representations from the tokens of the surface form. Such an approach has been used, for example, by Toutanova et al. (2015) to produce open relation embedding via a CNN. A model that reads the tokens of mentions and open relations can, in principle, handle any mention and open relation as long as the tokens have been observed during training.

We use a general model architecture that combines a relational model and a composition func-

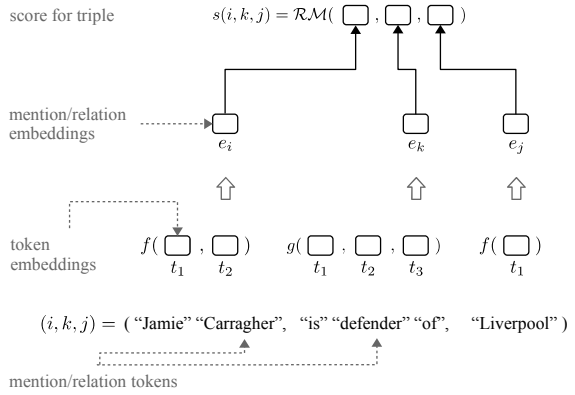

Figure 6: KGE model with composition. The tokens in triple (i, k, j) are first embedded individually and then composed into mention or relation embeddings. Finally, a KGE model $\mathcal{RM}$ is used to compute the triple's score.

tion, see Fig. 6. Formally, let $\mathcal{V}(\mathcal{E})^+$ be the set of non-empty token sequences over the token vocabulary $\mathcal{V}(\mathcal{E})$ of entity mentions. We denote by $d, o \in \mathbb{N}_+$ the size of the embeddings of entities and relations. We first embed each entity mention into a continuous vector space via an *entity mention embedding function* $f : \mathcal{V}(\mathcal{E})^+ \to \mathbb{R}^d$. Similarly, each open relation is embedded into a continuous vector space via a *relation embedding function* $g : \mathcal{V}(\mathcal{R})^+ \to \mathbb{R}^o$. The embeddings are then fed into a *relational scoring function* $\mathcal{RM} : \mathbb{R}^d \times \mathbb{R}^o \times \mathbb{R}^d \to \mathbb{R}$. Given a triple (i, k, j), where $i, j \in \mathcal{V}(\mathcal{E})^+$ and $k \in \mathcal{V}(\mathcal{R})^+$, our model computes the final score as $s(i, k, j) = \mathcal{RM}(\, f(i), g(k), f(j)\,)$.

6 Experiments

In our experimental study, we investigated whether a simple prototypical OLP model can predict genuinely new facts or if many successful predictions can be trivially explained by leakage or non-relational information. Our goal was to study the effectiveness and necessity of the mention-ranking protocol and leakage removal, and how much human effort is necessary to create suitable validation data. Finally, we inspected data and model quality.

We first describe the models and their training, then the performance metrics, and finally the evaluation. In our experimental results, model performance dropped by ≈25% with THOROUGH leakage removal so that leakage due to *paraphrasing* is indeed a concern. We also implemented two diagnostic models that use *non-relational information* (only parts of a triple) to predict answers. These models reached ≈20–25% of the prototypical model's performance, which indicates that relational modelling is important. In our quality and error analysis, we found that at least 74% of the prediction errors were *not* due to noisy data. A majority of incorrectly predicted entity mentions have a type similar to the one of the true entity.

6.1 Models and Training

Prototypical model. We use COMPLEX (Trouillon et al., 2016) as relational model, which is an efficient bilinear model and has shown state-of-the-art results. For the composition functions f and g, we used an LSTM (Hochreiter and Schmidhuber, 1997) with one layer and the hidden size equivalent to the token embedding size. We call this model COMPLEX-LSTM.[3]

Diagnostic models. To expose potential biases in the data, we employ two diagnostic models to discover how many questions can simply be answered without looking at the whole question, i.e., by exploiting *non-relational information*. Given question $(i, k, ?)$, the model PREDICT-WITH-REL considers $(r, ?)$ for scoring. E.g., for question *(“Jamie Carragher”, “is defender of”, ?)*, we actually ask *(“is defender of”, ?)*. This is likely to work reasonably for relations that are specific about the potential answer entities; e.g., predicting popular football clubs for *(“is defender of”, ?)*. The model uses scoring functions $s_t : \mathbb{R}^o \times \mathbb{R}^d \to \mathbb{R}$ and $s_h : \mathbb{R}^d \times \mathbb{R}^o \to \mathbb{R}$ for questions $(i, k, ?)$ and $(?, k, j)$ respectively:

$$s_t(k, e) = g(k)^T f(j), \quad s_h(i, k) = f(i)^T g(k)$$

Likewise, the PREDICT-WITH-ENT model ignores the relation by computing a score for pair (i, j). We use $s_e(i, j) = f(i)^T f(j)$

Training. See App. C for details about the hyperparameters, training and model selection.

Performance metrics. For evaluating a model's predictions, we use the ranking metrics mean reciprocal rank (MRR) and HITS@k. MRR is sensitive to the top-3 ranks with rapidly decaying reward,

[3] In a preliminary study, we investigated COMPLEX, ANALOGY, DISTMULT and RESCAL as relational models. COMPLEX was the most efficient and best performing model. For composition functions, we also investigated unigram pooling, bi-gram pooling with CNNs, self-attention and LSTMs. Here LSTMs worked well consistently. See App. E for additional results.

Leakage Removal	Model	Model Selection	MRR	HITS@1	HITS@10	HITS@50
SIMPLE	PRED-WITH-ENT	LINKED	0.0	0.0	0.0	0.0
	PRED-WITH-REL	LINKED	1.5	0.8	2.6	5.4
	COMPLEX-LSTM	LINKED	6.5	3.8	11.6	20.7
BASIC	PRED-WITH-ENT	LINKED	0.0	0.0	0.0	0.0
	PRED-WITH-REL	LINKED	1.0	0.5	1.6	3.6
	COMPLEX-LSTM	LINKED	4.8	2.6	8.9	17.6
THOROUGH	PRED-WITH-ENT	LINKED	0.0	0.0	0.0	0.0
	PRED-WITH-REL	LINKED	1.0	0.6	1.5	3.3
	COMPLEX-LSTM	LINKED	3.9	2.1	7.0	14.6
	COMPLEX-LSTM	ALL	2.7	1.5	4.7	9.1
	COMPLEX-LSTM	MENTION	3.8	2.1	7.1	14.1

Table 2: **Test** results. Comparing COMPLEX-LSTM, PREDICT-WITH-ENT and PREDICT-WITH-REL with all removal settings. Model selection on VALID-LINKED for all settings except in THOROUGH, where we also show VALID-MENTION and VALID-LINKED. Results in percent.

while HITS@k equally rewards correct answers in the top-k ranks. See App. D for a more formal definition of MRR and HITS@k. The ranks are based on mention ranking for VALID-LINKED and TEST and on entity-ranking (treating distinct mentions as distinct entities) for VALID-ALL and VALID-MENTION.

6.2 Results

Influence of leakage. In Tab. 2, we observed that BASIC leakage removal of evaluation data lowers the performance of all models considerably in contrast to the SIMPLE leakage removal. With the THOROUGH leakage removal, performace drops further; e.g., HITS@50 performance dropped by $\approx 25\%$ from SIMPLE. This confirms our conjecture that leakage can trivially explain some successful predictions. Most predictions, however, cannot be explained by paraphrasing leakage.

Influence of non-relational information. In Tab. 2, we see that PREDICT-WITH-ENT, which essentially learns popularity statistics between entity mentions, has no success on the evaluation data. However, PREDICT-WITH-REL reaches $\approx 20-25\%$ of HITS@50 performance of COMPLEX-LSTM by simply predicting popular mentions for a relation, even in the THOROUGH setting.

Effectiveness of mention-ranking. Tab. 3 shows validation results for the three types of validation data for COMPLEX-LSTM and THOROUGH removal. The evaluation protocol has access to alternative mentions only in VALID-LINKED, but not in VALID-ALL and VALID-MENTION. Clearly, using VALID-LINKED results in higher metrics when models associate different mentions to an answer entity.

Influence of model selection. The THOROUGH block of Tab. 2 shows the results for model selection based on VALID-ALL, VALID-MENTION or VALID-LINKED. In VALID-ALL, many triples contain common nouns instead of entity mentions, while in VALID-MENTION or VALID-LINKED triples have entity mentions in both arguments. Model selection based on VALID-ALL clearly picked a weaker model than model selection based on VALID-LINKED, i.e., it led to a drop of ≈35% of HITS@50 performance. However, there is no improvement when we pick a model based on VALID-LINKED versus VALID-MENTION. Thus, computing the MRR using alternative entity mentions did not improve model selection, even though—as Tab. 3 shows—the mention-ranking protocol gives more credit when alternative mentions are ranked higher. Our results suggest that it may suffice to use validation data that contains entity mentions but avoid costly entity disambiguation.

Overall performance. In Tab. 2 we observed that performance numbers seem generally low. For comparison, the HITS@10 of COMPLEX on FB15k-237—a standard evaluation dataset for LP in curated KGs—lies between 45% and 55%. We conjecture that this drop may be due to: (i) The

Leakage Removal	Model	Model Selection	MRR	HITS@1	HITS@10	HITS@50
	ComplEx-LSTM	All	2.9	1.8	5.0	8.9
Thorough	ComplEx-LSTM	Mention	3.6	2.0	6.5	13.0
	ComplEx-LSTM	Linked	4.2	2.3	7.5	14.9

Table 3: **Validation** results. Comparing the performances of ComplEx-LSTM for different validation datasets.

Types of prediction errors	
correct sense / wrong entity	68.0 %
wrong sense	13.5 %
noise	18.5 %
Types of data errors	
triple has error	12.0 %
mention is generic	14.0 %

Table 4: Error assessment of 100 sampled HITS@50 (filtered) prediction errors from Valid-Linked.

level of uncertainty and noise in the training data, i.e., uninformative or even misleading triples in OKGs (Gashteovski et al., 2019). (ii) Our evaluation data is mostly from the more challenging long tail. (iii) OKGs might be fragmented, thus inhibiting information flow. Also, note that the removal of evaluation data from training removes evidence for the evaluated long-tail entities. (iv) Naturally, in LP, we do not know all the true answers to questions. Thus, the filtered rank might still contain many true predictions. In OLP, we expect this effect to be even stronger, i.e., the filtered ranking metrics are lower than in the KG setting. Still, like in KG evaluation, with a large enough test set, the metrics allow for model comparisons.

Model and data errors. We inspected predictions for Valid-Linked from ComplEx-LSTM trained on Thorough. We sampled 100 prediction errors, i.e., triples for which no correct predicted mention appeared in the filtered top-50 rank. We classified prediction errors by inspecting the top-3 ranks and judged their consistency. We classified triple quality judging the whole triple. We counted an error as *correct sense / wrong entity*, when the top-ranked mentions are semantically sensible, i.e. for *("Irving Azoff", "was head of", ?)* the correct answer would be *"MCA Records"*, but the model predicted other record companies. We counted an error as *wrong sense* when—for the same example—the model mostly consistently predicted other companies or music bands, but not other record companies. If the predictions are inconsistent, we counted the error as *noise*.

An additional quality assessment is the number of wrong triples caused by extraction errors in OPIEC, e.g., *("Finland", "is the western part of", "the balkan peninsula")*, *("William Macaskill", "is vice-president of", "giving")*, or errors in alternative mentions. We also looked for generic mentions in the evaluation data. Such mentions contain mostly conceptual knowledge like in *("computer science", "had backgrounds in", "mathematics")*. Other generic triples, like *("Patrick E.", "joined the team in", "the season")*, have conceptual meaning, but miss context to disambiguate *"the season"*.

The results in Tab. 4 suggest that the low performance in the experiments is not due to noisy evaluation data. 74% of the examined prediction errors on Valid-Linked contained correct, non-generic facts. The shown model errors raise the question of whether there is enough evidence in the data to make better predictions.

7 Conclusion

We proposed the OLP task and a method to create an OLP benchmark. We created the large OLP benchmark OLPBench, which will be made publicly available[4]. We investigated the effect of leakage of evaluation facts, non-relational information, and entity-knowledge during model selection using a prototypical open link prediction model. Our results indicate that most predicted true facts are genuinely new.

Acknowledgments

The first author would like to gratefully thank the NVIDIA Corporation for the donation of a TITAN Xp GPU that was used in this research.

[4] https://www.uni-mannheim.de/dws/research/resources/olpbench/

References

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

William W. Cohen, Henry Kautz, and David McAllester. 2000. Hardening soft information sources. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 255–259, New York, NY, USA. ACM.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: The second generation. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 3–10.

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing open knowledge bases. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1679–1688.

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing open knowledge bases. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1679–1688, New York, NY, USA. ACM.

Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro. 2017. Minie: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2630–2640.

Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. 2019. OPIEC: an open information extraction corpus. *CoRR*, abs/1904.12324.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, pages 249–256.

Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. 2017. Knowledge transfer for out-of-knowledge-base entities : A graph neural network approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1802–1808.

Frederick Hayes-Roth. 1983. *Building expert systems*, volume 1 of *Advanced book program*. Addison-Wesley.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Sameh K. Mohamed, Aayah Nounu, and Vít Nováček. 2019. Drug target discovery using knowledge graph embeddings. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 11–18, New York, NY, USA. Association for Computing Machinery.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Fabio Petroni, Luciano Del Corro, and Rainer Gemulla. 2015. CORE: context-aware open relation extraction with factorization machines. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1763–1773.

Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *The Semantic Web – ISWC 2013*, pages 542–557, Berlin, Heidelberg. Springer Berlin Heidelberg.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia. Association for Computational Linguistics.

Baoxu Shi and Tim Weninger. 2018. Open-world knowledge graph completion. In *Proceedings of the*

Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 1957–1964.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1499–1509.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2071–2080.

Shikhar Vashishth, Prince Jain, and Partha Talukdar. 2018. Cesi: Canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 1317–1327, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Patrick Verga, Arvind Neelakantan, and Andrew McCallum. 2017. Generalizing to unseen entities and entity pairs with row-less universal schema. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 613–622.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL*, 6:287–302.

Tien-Hsuan Wu, Zhiyong Wu, Ben Kao, and Pengcheng Yin. 2018. Towards practical open knowledge base canonicalization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 883–892, New York, NY, USA. ACM.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380.

A Related Work

The following studies investigated KGs and OKGs in various ways, either by deriving KGs from OKGs or by using them jointly to improve inference in KGs.

Unseen entities. Shi and Weninger (2018) introduced *open-world knowledge base completion* (OWKBC), which assumes a curated KG as basis. The goal is to obtain new triples with unseen entities and known relations from the KG. Shi and Weninger (2018) proposes a link prediction model that allows questions involving unseen entities. Their model leverages the KG, relevant text fragments, word embeddings as well as an entity resolution module. Other approaches use structural information from the KG itself. Hamaguchi et al. (2017) assigns an embedding to an unseen entity based on embeddings of its neighboring entities and relations, whereas Verga et al. (2017) encodes an unseen entity pair by averaging the embeddings of the relations that link to it.

OpenIE-enhanced KGs. Universal schema models (Riedel et al., 2013) augment an existing KG with open relations between KG entities. Petroni et al. (2015) build upon Riedel et al.'s work by considering context information to improve the results further. Toutanova et al. (2015) embed open relations based on their tokens and dependency relations to augment the KG. In our work, we explore LP for OKGs, which differs in that only mentions are observed for both entities and relations. Neither a KG nor a vocabulary of entities is available during training and prediction.

Canonicalizing open knowledge. Cohen et al. (2000); Pujara et al. (2013); Vashishth et al. (2018); Wu et al. (2018); Galárraga et al. (2014) are the closest in spirit to this study, as they also want to make OKGs accessible without using a reference knowledge base. Cohen et al. (2000) calls open information a *soft database*, while Pujara et al. (2013) calls it an extraction graph from which a latent KG has to be identified. Common to all those approaches is that their ultimate target is to create a symbolic database with disambiguated entities and distinct relations. Thus they canonicalize the entities *and* the relations. In contrast, we are not canonicalizing the OKG but reason directly on the OKG. Galárraga et al. (2014) directly evaluates the induction of entity clusters, while we evaluate this

jointly in the context of LP.

Reading comprehension QA and language modelling. Two recently published reading comprehension question answering datasets—QAngaroo (Welbl et al., 2018) and HotPotQA (Yang et al., 2018)—evaluate multi-hop reasoning over facts in a collection of paragraphs. In contrast to these approaches, OLP models reason over the whole graph, and the main goal is to investigate the learning of relational knowledge despite ambiguity and noise. We consider those two directions as complementary to each other. Also, in their task setup, they do not stipulate a concept of relations between entities, i.e., the relations are assumed to be a latent/inherent property of the text in which the entities occur. This is true as well for language models trained on raw text. It has been shown that such language models can answer questions in a zero-shot setting (Radford et al., 2019). The authors of the latter study inspected the training data to estimate the number of near duplicates to their test data and could show that their model seemed to be able to generalize, i.e., to reason about knowledge in the training data.

TAC KBP Slot Filling. The TAC KBP Slot Filling challenge datasets provide a text corpus paired with canonicalized multi-hop questions. There are similarities to our work in terms of building knowledge from scratch and answering questions. The main difference is that our goal is to investigate the learning of knowledge without supervision on canonicalization and that we use link prediction questions to quantify model performance. If models in OLP show convincing progress, they could and should be applied to TAC KBP.

B Dataset creation

The process of deriving the dataset from OPIEC was as follows. Initially, the dataset contained over 340M non-distinct triples,[5] which are enriched with metadata such as source sentence, linguistic annotations, confidence scores about the correctness of the extractions and the Wikipedia links in the triple's subject or object. Triples of the following types are not useful for our purpose and are removed: (i) having a confidence score < 0.3,[6] (ii) having personal or possessive pronouns, wh-determiner, adverbs or determiners in one of their arguments, (iii) having a relation from an implicit appositive clause extraction, which we found to be very noisy, and (iv) having a mention or a relation that is longer than 10 tokens. This left 80M non-distinct triples. Next, we lowercased the remaining 60M distinct triples and collect an entity-mentions map from all triples that have an annotated entity. We collected token counts and created a mention token vocabulary with the top 200K most frequent tokens, and a relation token vocabulary with the top 50K most frequent tokens. This was done to ensure that each token is seen at least ≈ 50 times. Finally, we kept only the triples whose tokens were contained in these vocabularies, i.e., the final 30M distinct triples.

[5] The triples can be non-distinct, i.e., duplicates, when they have been extracted from different sentences.

[6] The confidence score is computed by a classifier that determines the probability of the triple having an extraction error. Refer to OPIEC's publication for further description.

C Training details

C.1 Multi-Label Binary Classification Batch-Negative Example Loss

Recent studies (Dettmers et al., 2018) obtained state-of-the-art results using multi-label binary classification over the full entity vocabulary. Let the cardinality of the OKG's mention set be $N = |\mathcal{T}_h \cup \mathcal{T}_t|$. A training instance is either a prefix (i, k) with label $y^{ik} \in \{0, 1\}^N$ given by

$$y_c^{ik} = \begin{cases} 1 & \text{if } (i, k, c) \in \mathcal{T} \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for } c \in \{1, .., N\}$$

or, likewise, a suffix (k, j) and $y^{kj} \in \{0, 1\}^N$.

Computing such a loss over the whole entity mention vocabulary is infeasible because (a) our entity mention vocabulary is very large and (b) we have to recompute the entity mention embeddings after each parameter update for each batch. To improve memory efficiency and speed, we devise a strategy to create negative examples dubbed *batch negative examples*. This method simplifies the batch construction by using only the entities in the batch as negative examples. Formally, after sampling the prefix and suffix instances for a batch b, we collect all true answers in a set $\hat{B}_b$, such that the label vectors y^{ik} and y^{kj} in batch b is defined over $\hat{B}_b$ and the loss in batch b is computed by

$$L^{ik} = \frac{1}{|B_b|} \sum_{c \in \hat{B}_b} -[y_c^{ik} \cdot \log \sigma(s(i, k, c)) + (1 - y_c^{ik}) \cdot \log(1 - \sigma(s(i, k, c)))]$$

Leakage Removal	Model	Model Selection	MRR	HITS@1	HITS@10	HITS@50
	COMPLEX-UNI	ALL	2.2	0.8	4.7	10.2
	COMPLEX-UNI	MENTION	2.2	0.9	4.7	10.3
	COMPLEX-UNI	LINKED	2.2	0.9	4.7	10.3
	DISTMULT-LSTM	ALL	3.2	1.7	5.9	11.6
	DISTMULT-LSTM	MENTION	3.3	1.8	5.9	12.2
	DISTMULT-LSTM	LINKED	3.3	1.8	5.9	12.2
THOROUGH	COMPLEX-LSTM-XL	ALL	3.3	1.8	5.8	12.0
	COMPLEX-LSTM-XL	MENTION	3.6	1.9	6.6	13.9
	COMPLEX-LSTM-XL	LINKED	3.6	1.9	6.6	13.9
	COMPLEX-LSTM	ALL	2.7	1.5	4.7	9.1
	COMPLEX-LSTM	MENTION	3.8	2.1	7.1	14.1
	COMPLEX-LSTM	LINKED	3.9	2.1	7.0	14.6

Table 5: Additional **Test** results. Comparing DISTMULT-LSTM, COMPLEX-LSTM-XL with embedding size 768, COMPLEX-UNI with uni-gram pooling as composition function. Model selection on VALID, VALID-LINKED and VALID-MENTION, models trained on THOROUGH; Results in percent.

and L^{kj} is computed likewise. With *batch negative examples* the mentions/entities appear in expectation proportional to their frequency in the training data as a "negative example".

C.2 Training settings

We used Adagrad with mini batches (Duchi et al., 2011) with batch size 4096. The token embeddings were initialized with the Glorot initialization (Glorot and Bengio, 2010). One epoch takes ≈ 50 min with a TitanXp/1080Ti. We performed a grid search over the following hyperparameters: entity and relation token embedding sizes $[256, 512]$, drop-out after the composition function f and g $[0.0, 0.1]$, learning rate $[0.05, 0.1, 0.2]$ and weight decay $[10^{-6}, 10^{-10}]$. We trained the models for 10 epochs and selected the hyperparameters, which achieved the best MRR with mention ranking on VALID-LINKED. We trained the final models for up to 100 epochs but did early stopping if no improvement occured within 10 epochs.

D Performance Metrics

Denote by $\mathcal{M}(\mathcal{E})$ all mentions from the dataset. Denote by $\mathcal{Q}$ the set of all questions generated from the evaluation data. Given a question $q_t \in \mathcal{Q}$, we rank all $m \in \mathcal{M}(\mathcal{E})$ by the scores $s(i, k, m)$ (or $s(m, k, j)$ for $q_h \in \mathcal{Q}$), then filter the raw rank according to either the entity-ranking protocol or the mention-ranking protocol. Finally, we record the positions of the correct answers in the filtered ranking.

MRR is defined as follows: For each question $q \in \mathcal{Q}$, let RR_q be the filtered reciprocal rank of the *top-ranked* correct answer. MRR is the micro-average over $\{\text{RR}_q \mid q \in \mathcal{Q}\}$. HITS@k is the proportion of the questions where at least one correct mention appears in the top k positions of the filtered ranking.

E Additional Results

Tab. 5 provides results for other models and hyperparameters. The COMPLEX-LSTM results from the Sec. 6 are given at the bottom for comparison. COMPLEX-LSTM-XL has a larger embedding size of 768, which did not help to improve the results. COMPLEX-UNI is the ComplEx model with the uni-gram pooling composition function, i.e., averaging the token embeddings. Compared to COMPLEX-LSTM it shows that LSTM as a composition function did yield better results. DISTMULT-LSTM is the DistMult relational model (Yang et al., 2015) with an LSTM as composition function, which did not improve over COMPLEX-LSTM. In Summary, the results support the hyperparameters, model and composition function chosen for the experiments in Sec. 6. Overall, we observed that model selection based on VALID-ALL seems to have a higher variance because the model selected for COMPLEX-LSTM with VALID-ALL is outperformed by other models, whereas COMPLEX-LSTM performed best for models selected with VALID-MENTION and VALID-LINKED.